

UNIVERSITY OF SPLIT  
FACULTY OF ELECTRICAL ENGINEERING, MECHANICAL ENGINEERING  
AND NAVAL ARCHITECTURE

**Lea Dujic Rodic**

**Applying Machine Learning Techniques to Enhance the  
Performance of Internet of Things Stack Services**

DOCTORAL THESIS

Split, 2023.



UNIVERSITY OF SPLIT  
FACULTY OF ELECTRICAL ENGINEERING, MECHANICAL ENGINEERING  
AND NAVAL ARCHITECTURE

**Lea Dujic Rodic**

***Applying Machine Learning Techniques to Enhance the  
Performance of Internet of Things Stack Services***

DOCTORAL THESIS

Split, 2023.

The research reported in this thesis was carried out at the Department of COMMUNICATION AND INFORMATION SYSTEMS, University of Split, Faculty of Electrical Engineering, Mechanical Engineering, and Naval Architecture, within the framework of a scientific research project "Internet of Things: Research and Applications", UIP-2017-05-4206, funded by the Croatian Science Foundation.

Supervisors:

Prof. dr. sc. Andrina Granić, Faculty of Science, University of Split

Izv. prof. dr. sc. Petar Šolić, FESB, University of Split, Croatia

Disertation number: xxx-xxx

---

## BIBLIOGRAPHIC INFORMATION

Keywords: xxxxx, yyyy

Scientific area: xxxxx

Scientific field: xxxxx

Scientific branch: xxxxx

Institution of PhD completion: University of Split, Faculty of Electrical Engineering, Mechanical Engineering and Naval Architecture

Supervisor of the thesis: Xxx dr.sc. Xxx Yyyy

Number of pages: yyy

Number of figures: yyy

Number of tables: yyy

Number of references: yyy

---

Committee for assessment of doctoral dissertation:

1. XXXX. dr. sc. XXXX Yyyyy, Institution name and City/Town
2. XXXX. dr. sc. XXXX Yyyyy, Institution name and City/Town
3. XXXX. dr. sc. XXXX Yyyyy, Institution name and City/Town
4. XXXX. dr. sc. XXXX Yyyyy, Institution name and City/Town
5. XXXX. dr. sc. XXXX Yyyyy, Institution name and City/Town

Committee for defence of doctoral dissertation:

1. XXXX. dr. sc. XXXX Yyyyy, Institution name and City/Town
2. XXXX. dr. sc. XXXX Yyyyy, Institution name and City/Town
3. XXXX. dr. sc. XXXX Yyyyy, Institution name and City/Town
4. XXXX. dr. sc. XXXX Yyyyy, Institution name and City/Town
5. XXXX. dr. sc. XXXX Yyyyy, Institution name and City/Town

Dissertation defended on: xx. month. 20xx.

# **Applying Machine Learning Techniques to Enhance the Performance of Internet of Things Stack Services**

## **Abstract:**

This research focuses on the exploration and enhancement within the three-layer IoT architecture - Perception, Network, and Application layers - through the implementation of Machine Learning algorithms. The research objective is to improve the delivery and performance of IoT services by harnessing data to gain hidden insights about the environment and devices in operation. For the Perception layer, Machine Learning models have been developed, using the Received Signal Strength Indicator from IoT devices to accurately interpret environmental conditions. The Network Layer sees the optimization of throughput of RFID Gen2 systems via leveraging Machine Learning models for frame size and tag count estimation. Finally, the Application Layer's advancements are demonstrated in an innovative plush Smart Toy that integrates IoT sensing technology and ML algorithms for educational purposes. The research concludes with results presenting the improved efficiencies and the vast potential for Machine Learning in shaping the future evolution of IoT services.

## **Keywords:**

Internet of Things, RFID, Machine Learning, Smart toy, human-computer interaction, LoRaWAN, parking occupancy, Neural Network, Soil humidity, RSSI

# **Primjena strojnog učenja za unapređenje performansi usluga stoga Interneta stvari**

## **Sažetak:**

Ovo istraživanje se usmjerno je na proučavanje i unapređenje performansi usluga stoga Interneta stvari - usluge sloja Percepcije, Mreže i Aplikacija - primjenom algoritama strojnog učenja. Cilj istraživanja je unaprijediti isporuku i performanse IoT usluga putem analize podataka kako bi se stekli skriveni uvidi o okolini i aktivnim uređajima. Za sloj Percepcije, razvijeni su modeli strojnog učenja koji iz jačine primljenog signala s IoT uređaja detektiraju stanja i promjene u okolišu. Na Mrežnom sloju postiže se optimizacija propusnosti RFID Gen2 sustava putem implementiranjem modela strojnog učenja za procjenu veličine okvira i broja odzivnika. Postignuća na sloju Aplikacije demonstrirana su kroz inovativnu plišanu pametnu igračku koja integrira IoT senzorsku tehnologiju i algoritme strojnog učenja u edukativne svrhe. Istraživanje završava rezultatima koji predstavljaju unaprijeđenje učinkovitosti i ogroman potencijal strojnog učenja u oblikovanju budućeg razvoja IoT usluga.

## **Ključne riječi:**

Internet stvari, RFID, strojno učenje, pametna igračka, interakcija čovjeka i računala, LoRaWAN, zazuzeće parkinga, neuronska mreža, vlažnost tla, RSSI





# Acknowledgments

*This PhD thesis ....*

# Contents

Abstract . . . . .	v
Sažetak . . . . .	vi
Acknowledgments . . . . .	viii
Curriculum Vitae . . . . .	x
List of Tables . . . . .	xii
List of Figures . . . . .	xv
<b>1 INTRODUCTION</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.1.1 Perception layer: Challenges and Opportunities . . . . .	6
1.1.2 Network Layer: Challenges and Opportunities . . . . .	7
1.1.3 Application Layer: Challenges and Opportunities . . . . .	8
1.1.4 Hypothesis . . . . .	9
1.2 Scientific methodology and contributions . . . . .	11
1.2.1 Theoretical part of the research . . . . .	12
1.2.2 The Empirical part of the research: Perception Layer . . . . .	12
1.2.3 The Empirical part of the research: Network Layer . . . . .	36
1.2.4 The Empirical part of the research: Application Layer . . . . .	49
1.2.5 The fundamental scientific contributions . . . . .	75
1.3 List of Published Papers Upon Which the Dissertation’s Scientific Contribution was Founded . . . . .	77
1.3.1 Other publications . . . . .	78
1.4 Overview of the Dissertation Structure . . . . .	80
<b>2 STATE OF THE ART</b>	<b>81</b>
2.1 Machine Learning: General Overview . . . . .	81
2.1.1 Algorithms . . . . .	82
2.2 Perception Layer: Overview of Scientific Literature . . . . .	94
2.2.1 Soil Humidity Sensing . . . . .	94
2.2.2 Smart Parking Solutions . . . . .	99
2.3 Network Layer: Overview of Scientific Literature . . . . .	109
2.4 Application Layer: Overview of Scientific Literature . . . . .	113

<b>3</b>	<b>OVERVIEW OF THE SCIENTIFIC CONTRIBUTION OF PUBLICATIONS</b>	<b>119</b>
3.1	Paper 1: Machine Learning and Soil Humidity Sensing: Signal Strength Approach . . . . .	119
3.2	Paper 2: Sensing Occupancy through Software: Smart Parking Proof of Concept . . . . .	121
3.3	Paper 3: Privacy leakage of LoRaWAN smart parking occupancy sensors . .	122
3.4	Paper 4: Tag Estimation Method for ALOHA RFID System Based on Machine Learning Classifiers . . . . .	124
3.5	Paper 5: Tangible Interfaces in Early Years' Education: A Systematic Review	126
3.6	Paper 6: Towards a Machine Learning Smart Toy Design for Early Childhood Geometry Education: Usability and Performance . . . . .	128
<b>4</b>	<b>CONCLUSION</b>	<b>131</b>
	<b>BIBLIOGRAPHY</b>	<b>135</b>
	<b>APPENDIX A</b>	<b>160</b>
	<b>APPENDIX B</b>	<b>195</b>
	<b>APPENDIX C</b>	<b>215</b>
	<b>APPENDIX D</b>	<b>237</b>
	<b>APPENDIX E</b>	<b>237</b>
	<b>APPENDIX F</b>	<b>277</b>

# List of Tables

1.1	<i>Pearson correlation matrix between soil humidity and RSSI and SNR.</i>	17
1.2	<i>Comparison parameters of ML algorithms for soil humidity estimation</i>	20
1.3	<i>Table of best results using the HMM model obtained for each gateway</i>	26
1.4	<i>Selection of the hyper parameters for Neural Network evaluation.</i>	27
1.5	<i>Result of grid search of Hyper-parameters for a particular sensor.</i>	28
1.6	<i>The results for Random Forest model.</i>	28
1.7	<i>Feature importance rate for each of the five sensors.</i>	30
1.8	<i>Hyper parameters selected for Neural Network model performance testing.</i>	30
1.9	<i>Best results obtained for NN model</i>	31
1.10	<i>Results of Dietterich's 5x2-Fold Cross-Validation statistical test for RF and NN model for all five sensors and significance level <math>\alpha = 0.05</math></i>	34
1.11	<i>Result of grid search of Hyper-parameters for a particular sensor without time variables.</i>	35
1.12	<i>The results for Random Forest model for a particular sensor without time variables.</i>	35
1.13	<i>Best results obtained for NN model without time variables.</i>	35
1.14	<i>Snapshot of the obtained data</i>	40
1.15	<i>Grid search results of RF Hyper-parameters for a particular frame size.</i>	41
1.16	<i>Tested Hyper-parameters for Random Forest.</i>	41
1.17	<i>Classification accuracy of NN, RF and the ILCM model for a particular frame size.</i>	42
1.18	<i>MAE of NN, RF and the ILCM model for a particular frame size.</i>	43
1.19	<i>Model accuracy before and after quantisation</i>	46
1.20	<i>Model performance on Teensy 4.0 MCU, Arduino DUE and Raspberry PI4</i>	47
1.21	<i>SVM Parameter Grid Search</i>	56

1.22	<i>Random Forest Parameter Grid Search . . . . .</i>	57
1.23	<i>Neural Network Parameter Grid Search . . . . .</i>	57
1.24	<i>ML results for raw data . . . . .</i>	57
1.25	<i>ML results for cubic interpolated data . . . . .</i>	59
1.26	<i>Structured Interview . . . . .</i>	66
1.27	<i>The Again-Again Table . . . . .</i>	66
1.28	<i>The number of correct and incorrect answers given by children in the pre-test stage when identifying geometric shapes. . . . .</i>	67
1.29	<i>Fun and subjective satisfactions measured with the Smileyometer rating scale. . . . .</i>	71
1.30	<i>The results of the answers to interview questions 4. and 5 . . . . .</i>	72
1.31	<i>The number of correct and incorrect answers given by children in the post-test stage for identifying geometric shapes. . . . .</i>	73
1.32	<i>Machine Learning gesture classification accuracy results . . . . .</i>	74
2.1	<i>Comparison table of Machine Learning models and applications for Soil moisture and drought estimation. . . . .</i>	97
2.2	<i>Comparison table of various radio technologies and applications based on signal strength variations. . . . .</i>	98
2.3	<i>Comparison table of various sensing technologies and it applications in Smart Parking. . . . .</i>	108

# List of Figures

1.1	<i>The IoT landscape.</i>	1
1.2	<i>The three-layer Architecture of IoT</i>	2
1.3	<i>Data rate and power consumption vs distance for the most diffused wireless communication technologies for IoT.</i>	4
1.4	<i>Conceptual framework for detection, processing, and state estimation in IoT environments based on Machine Learning models.</i>	12
1.5	<i>LoRa and LoRaWAN protocol stack</i>	13
1.6	<i>LoRaWAN (Long Range Wide Area Network) architecture.</i>	14
1.7	<i>Implementation of LoRaWAN-based soil moisture sensing device.</i>	15
1.8	<i>Network architecture of LoRaWAN-based soil moisture sensor system.</i>	15
1.9	<i>RSSI from Gateway 2, and soil moisture during the winter period.</i>	16
1.10	<i>Soil moisture estimation using the SVR model on the test set compared to expected soil moisture values</i>	18
1.11	<i>(a) Learning path of model with training and validation loss, (b) estimation of soil humidity with the model compared to expected values of soil humidity.</i>	19
1.12	<i>Position of parking lots with LoRaWAN Smart parking sensors and location of LoRaWAN gateways that captured information from sensor devices.</i>	20
1.13	<i>Probability density function of RSSI and SNR values alterations for diverse sensors and occupancy status from GW1, GW2 and GW3, respectively.</i>	22
1.14	<i>(left) Difference of RSSI values when the parking space remains free, (right) Difference of RSSI values for change of state from occupied to free for sensor 2 from GW 3.</i>	23
1.15	<i>Illustration of second-order Hidden Markov model for detecting occupancy status based on change of RSSI values</i>	25
1.16	<i>Architecture of Neural Network model for parking space occupancy classification</i>	26
1.17	<i>ROC curves for Adam optimizer with the learning rate of 0.001 and 100 epochs.</i>	27
1.18	<i>ROC curves for Sensor 1, 2, 3, 4 and 5 respectfully for Random Forest model.</i>	29

1.19	<i>ROC curves obtained on the test set for Sensor 1, 2, 3, 4 and 5 respectfully for Neural Network model . . . . .</i>	32
1.20	<i>Distribution of classification accuracy scores for both Random Forest and Neural Network model from repeated stratified 10-fold cross-validation. . .</i>	33
1.21	<i>An example of an interrogating frame of a frame size <math>L = 2^Q</math>. <math>i</math> represents the size of a particular part of the frame. . . . .</i>	37
1.22	<i>Architecture of the Neural Network model for tag estimation. . . . .</i>	40
1.23	<i>Comparison of absolute errors for Neural Network, Random Forest and ILCM model for frame sizes <math>L = 8</math>, <math>L = 16</math> and <math>L = 256</math>. . . . .</i>	44
1.24	<i>Comparison of throughput for the NN model, ILCM and Optimal model for scenario of frame size <math>L = 32</math> realization. . . . .</i>	45
1.25	<i>“Devices used in the test: Teensy 4.0 (left), Arduino DUE (center), Raspberry PI4 (right), source: Own photo . . . . .</i>	47
1.26	<i>Main Smart toy hardware components . . . . .</i>	50
1.27	<i>Electronic components and interfaces between devices . . . . .</i>	52
1.28	<i>Distance sensors evaluated during the development of the proposed device (a) Parallax Ping)), (b) In house developed capacitive sensor, (c) short-range Sharp IR sensor, (d) long-range Sharp IR sensor . . . . .</i>	52
1.29	<i>Researcher interacting with the first prototype of the device, featuring capacitive proximity sensors . . . . .</i>	53
1.30	<i>Raw sensor readings, 2 capacitive sensors . . . . .</i>	55
1.31	<i>Raw sensor readings, 4 short-range IR sensors . . . . .</i>	55
1.32	<i>Raw sensor readings, 4 long-range sensor . . . . .</i>	56
1.33	<i>IR long interpolation . . . . .</i>	59
1.34	<i>ANN Arhitecture . . . . .</i>	60
1.35	<i>Confusion matrix for models that include 3 shapes (left), 4 shapes (middle), and 5 shapes (right) . . . . .</i>	60
1.36	<i>Raw sensor data for five hand gestures -shapes recorded with a single sensor</i>	61
1.37	<i>Overall framework of the experiment . . . . .</i>	63
1.38	<i>Visualization of laboratory setup . . . . .</i>	64
1.39	<i>Smileyometer rating scale . . . . .</i>	66
1.40	<i>Interaction duration related to the established manner of the performed gesture</i>	68
1.41	<i>Gesture time per particular shape . . . . .</i>	69



1.42	<i>User mapping accuracy per particular shape . . . . .</i>	70
1.43	<i>The results of the answers to interview questions 2 and 3. . . . .</i>	70
1.44	<i>Results from the responses from Again-Again Table 1.27 . . . . .</i>	72
1.45	<i>Comparison of raw sensor data of child and adult subject while performing gestures for a same geometrical shape . . . . .</i>	74
1.46	<i>Realization and potential implementation of solutions . . . . .</i>	76
2.1	<i>Example of an Architecture of Random forest model. . . . .</i>	84
2.2	<i>Example of k- Nearest Neighbour classification. . . . .</i>	89
2.3	<i>Example of a Neural Network Architecture . . . . .</i>	90
2.4	<i>Long Short-Term Memory (LSTM) cell. . . . .</i>	91
2.5	<i>Example of a Convolutional Neural Network Architecture. . . . .</i>	94



# 1 INTRODUCTION

The introductory chapter of this dissertation explores the potential and perspective of the Internet of Things (IoT) for all three layers of the technology stack, investigating the specific challenges and opportunities posed by the IoT for various applications. Moreover, the section examines the utilization of modern Machine Learning (ML) techniques to address these issues. Within this context, the objectives and hypotheses are presented, focused on improving the performance of services in the Internet of Things stack through ML techniques. Furthermore, the remainder of the Introduction section expounds upon the scientific methodologies utilized to corroborate these hypotheses and showcases the primary scientific contributions made within the field.

## 1.1 Motivation

The technologies of the 21st century, particularly wireless technologies, have strongly influenced the development of various Internet-connected devices in recent years, shaping a new paradigm - the Internet of Things (IoT). IoT represents a concept of ubiquitous computing technology that enables the reception of information from the physical world that was previously unavailable and interconnects it for exchange and use in the digital world [1]. The IoT landscape is characterized by a complex interplay of four major components: "things", data, people, and processes as depicted in Figure 1.1 [2].

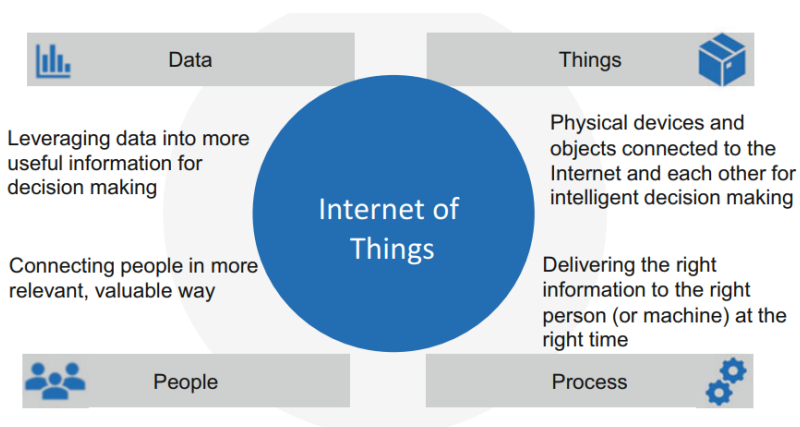


Figure 1.1. The IoT landscape [2].

At the center of the IoT are physical devices and objects that are embedded with sensors, processors, and network connectivity, so called "things". In the IoT context, the "things" include sensors, actuators, mobile devices, and other connected devices that enable data collection, analysis, and control [3]. The second component is data - the vast amounts of information generated by the things themselves, as well as by the people and processes that interact with them. The third component of the IoT is people - the individuals who design, operate, and interact with IoT systems. This includes end-users who interact with IoT-enabled devices, as well as the engineers, developers, and other professionals who design and deploy IoT systems. Finally, the fourth component are processes - the set of procedures, workflows, and protocols that enable IoT systems to function effectively. This includes everything from data collection and analysis to system monitoring and management, as well as the various interactions and transactions that occur between the other three components of the IoT.

Today, IoT technologies are considered one of the key pillars of the Fourth Industrial Revolution because of their significant potential for innovation and beneficial effects for society. These technologies have influenced many areas of daily life, particularly in the domains of automation, industrial production, logistics, healthcare, agriculture, business/process management, household appliances, and buildings [4, 5].

The 3-layer architecture of IoT systems (depicted in Figure 1.2), comprised of the Application, Network, and Perception Layers, has gained widespread recognition as a primary and essential framework.

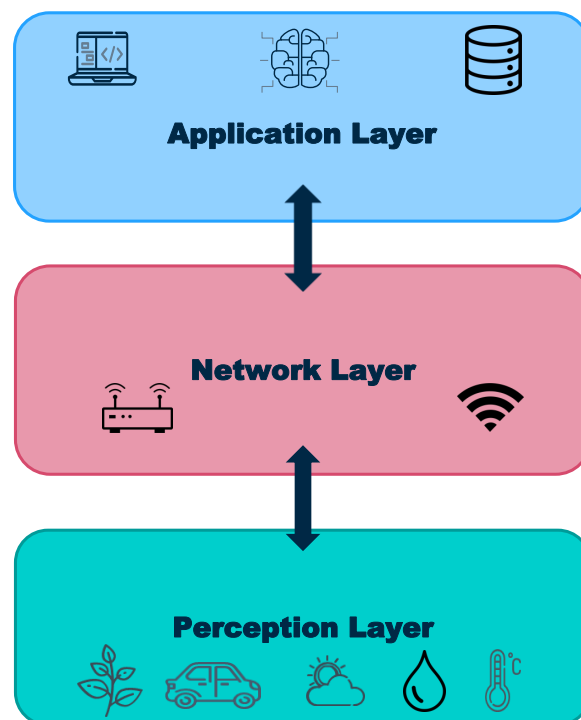


Figure 1.2. The Three-layer Architecture of IoT.

This architecture delineates the essential layers involved in various critical IoT operations, including data acquisition, processing, and deployment of applications. Its adoption and refinement have been widespread in the IoT research community, reflecting its significance and utility as a springboard for the development and realization of IoT solutions [6, 7, 8].

**The Perception Layer** in IoT architecture is the initial layer that captures data from the physical environment through sensors and other types of input devices [9]. This layer is analogous to the skin and sensory organs of the human body, as it serves as the interface between IoT devices and their surrounding environment [6]. The perception layer includes a range of sensors and actuators that can measure parameters such as temperature, pressure, motion, and humidity. Sensors serve as the fundamental components of the perception layer, responsible for identifying and collecting real-time data by recognizing changes in the environment. [10].

**The Network Layer** is responsible for transporting the data provided by the perception layer to the application layer. This layer includes all the technologies and protocols that make this connection possible and operates by using some of the latest technologies to provide heterogeneous network services [8]. The protocols used in IoT vary in terms of their advantages and disadvantages, and the selection of a particular protocol depends on the specific application requirements. One critical component of the network layer is the **network gateway**, which serves as the mediator between different IoT nodes by aggregating, filtering, and transmitting data to and from different sensors ensuring better interoperability between different systems [11]. Moreover, they perform initial, local preprocessing of the sensor data by filtering and organizing them into packages, thus reducing the amount of transmitted data, which results in a reduction of network communication costs. Wireless protocols are particularly important in this layer, as they allow sensors to be installed in hard-to-reach environments, require less material and human resources for installation, and can easily add or remove various nodes without reconsidering the entire network's structure. The choice of a protocol to use depends on the network's size, the power consumption of each node, and the transmission speed needed in a given application [8].

Generally, there are two types of network protocols in IoT solutions: Low Power Wide Area Networks (LPWANs) and short-range wireless networks. Short-range wireless networks include Radio Frequency Identification (RFID), Zigbee, IEEE 802.15.4, Bluetooth/BLE, and Wi-Fi, while LPWANs include SigFox, LoRa, and NB-IoT [12, 13, 14, 15]. The characteristics that differentiate these protocols are data transfer speed, cost, energy consumption, distance coverage, and security as depicted in Figure 1.3. LPWANs are generally considered a better-adapted protocol for IoT applications [16] since short-range technologies have high power consumption. It is important to note that network gateways also have the task of ensuring data security since they manage the flow of information in both directions. By applying appropriate encryption and security tools, they can prevent the leakage of IoT

data stored in the cloud and reduce the risk of malicious external attacks on IoT devices [17].

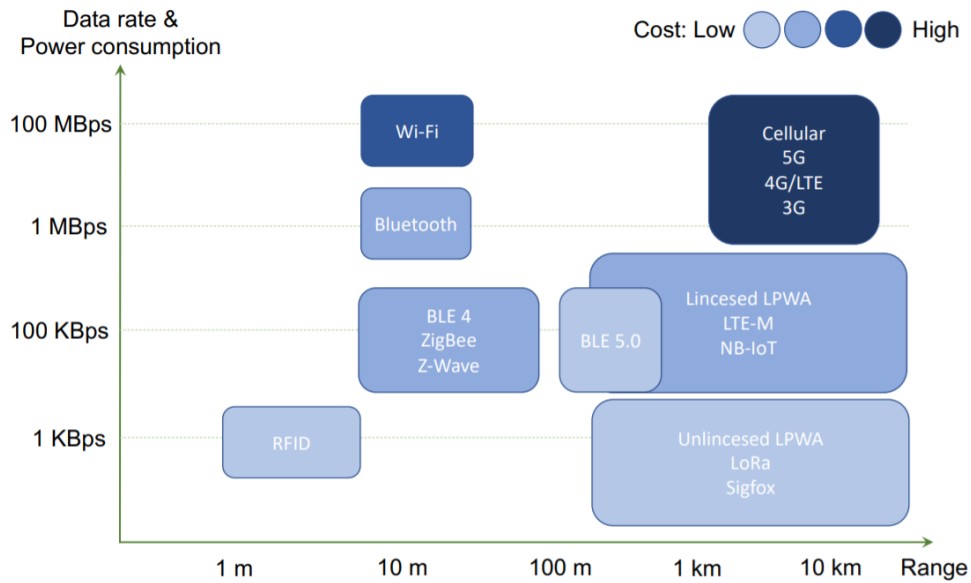


Figure 1.3. Data rate and power consumption vs distance for the most diffused wireless communication technologies for IoT [18].

**The Application Layer** within the context of IoT serves as a repository where data previously gathered from lower layers are stored, processed, aggregated, and filtered with the aim of providing bespoke services [8]. It includes all the necessary software required to deliver these services, with databases, analysis software, and middleware being commonly used to manage the vast amount of data provided by devices. Overall, the Application Layer is critical in offering the desired smart services and meeting the requirements of the end-users [6]. It covers numerous vertical markets such as smart homes, transportation, industrial automation, smart healthcare, and more [7]. Cloud computing and edge computing are two widely used software technologies to process data in the Application Layer, enabling services such as storage or data processing from a set of pre-existing resources in a distributed architecture.

The data collected by IoT devices is stored, analyzed, and processed in the cloud, using resources such as servers, advanced data analytic engines, and Machine Learning (ML) algorithms. In the IoT paradigm, there are many requirements for smart connected devices, including high data transmission speeds, very low latency, reliability, resilience, wide coverage, long battery life, security, and the ability to operate a network with a large number of users [19]. Therefore, efficient big data analysis that can extract significant information and correlations from the huge amounts of data generated by sensor devices is a key factor for success in many IoT domains [20]. Cloud computing provides an efficient way to perform advanced data processing on large amounts of data without significant costs [21]. Machine Learning algorithms are of immense potential in the aforementioned context for data analysis and precise predictions based on past observations for new sensor measurements. Moreover,

ML can enable users to gain deeper insights into the correlations among data, mine data, and discover hidden connections among data [22]. Deep learning (DL) is one of the leading Machine Learning technologies today, which is more effective in solving complex problems that were previously impossible to solve using traditional Machine Learning algorithms [23]. DL has been recognized as one of the top ten breakthrough technologies in 2013 and is the fastest-growing trend for big data analysis [24], as it enables the understanding of more complex patterns in data, classification, and prediction [25]. DL applications achieve exceptional accuracy and popularity in various fields, especially those related to image and sound processing [23].

The successful implementation of IoT as a ubiquitous technology is impeded by various challenges that need to be overcome. Key issues that must be addressed to enable efficient implementation of IoT include the absence of standardized architectures and protocols, as well as security and privacy concerns. Other obstacles include device heterogeneity, scalability, energy efficiency, interoperability, and effective data management [26, 2].

Although IoT technologies have given a new perspective for further advancement in various fields such as engineering or medicine, some potential areas of application of these technologies are still unknown or it is unclear how to approach them, which is an evident indicator that more intensive research activity is needed in this challenging domain [27]. Scientists point out that to overcome key technological challenges necessary for the adaptation of IoT systems, special attention needs to be directed towards overcoming energy and computational requirements [28]. Namely, the powering of sensor devices needs to be more efficient, data processing units and their algorithms more effective for extending battery life, and research needs to be focused on all layers of the IoT three-layer architecture to achieve adequate infrastructure [29]. To achieve this, diverse technologies are needed, from those for object identification to those for understanding the semantics of generated data. While in the dawn of IoT, most research and development efforts were dedicated to realizing efficient communication solutions for connecting with any object, today attention is focused on processing and analyzing data collected from the environment [30]. Research in this context particularly emphasizes the need to achieve the main goals of IoT, which are creating smart environments/spaces in the domains of energy, smart cities, transportation, smart homes, environment, supply chains, and healthcare [27, 28, 29].

In light of the aforementioned context, the following sections will consider several of the above-mentioned areas in line with each layer of the IoT three-layer architecture, addressing specific segments of application using Machine Learning models. Although ML has significant potential in various IoT applications, such as smart homes, industrial automation, transportation, and healthcare, there are still some challenges that need to be addressed in the application of Machine Learning to IoT.

### 1.1.1 Perception layer: Challenges and Opportunities

Sustainable global growth depends on several factors such as economic efficiency, quality of education, industry, and the environment, with agriculture being one of the most important factors in this growth [31]. In this context, the agricultural industry and environmental protection areas are ideal candidates for IoT solutions integration, as they require continuous monitoring and control, especially at the Perception Layer [32].

Implementing IoT devices within the agriculture sector could be an effective way to improve productivity by addressing some of the main issues faced by farmers [33]. It is predicted that in the next thirty years, the global population will reach 9.7 billion people, and approximately 70% of the world's population is expected to live in urban areas, resulting in increased demand for food [34]. Agricultural production must, therefore, significantly increase to keep up with market demands while also dealing with "traditional" problems such as unpredictable weather conditions. One of the crucial parameters for production efficiency is the proper use of water resources in irrigation, as the agricultural sector consumes 85% of the world's available freshwater resources [35]. Moreover, it is estimated that developing countries lose up to 40% of the water they use for irrigation [36]. Thus, there is a real need to improve the irrigation system, and irrigation implementation could be effectively addressed with new sensor technologies, namely the application of appropriate IoT infrastructure for soil moisture assessment, achieving financial and energy savings [37, 38]. A study by the California Department of Water Resources shows that with "smart" irrigation techniques, water consumption can be reduced by 6% to 41%, depending on the location of the research [39].

Most existing solutions are based on measuring the electrical properties of the soil, and the data is delivered through some wireless interface. Data is collected from energy-intensive and expensive sensors. Over time, such systems are difficult to maintain, especially if it is necessary to replace the batteries of a large number of devices in remote areas. Therefore, new solutions must produce an alternative, cost-effective and energy-efficient device that has a unique advantage over existing solutions [38]. A new, cost-effective concept of measuring soil moisture could be based on LPWA technology and appropriate machine and deep learning models to ensure optimal water use.

Another effect of accelerated population growth is the increase in the number of vehicles in urban areas. The European Commission and most developed countries emphasize that smart and sustainable mobility is one of the central concepts in the vision of a smart city [40], in which IoT plays a central role [41]. Parking, as a result of personal vehicle use, is becoming a major problem in terms of rational use of urban space. Available parking capacities on traffic surfaces in the centers of larger cities are almost completely used up and limited, and existing parking systems are inadequate [42]. Studies have shown that due to traffic congestion in urban areas, 30-50% of drivers search for free parking, while an IBM



study indicates that drivers in metropolitan cities such as Beijing or Madrid spend an average of 30 to 40 minutes searching for a free parking spot [43, 44]. One of the main problems that arises from this is the increase in fuel consumption and air pollution [45]. Moreover, due to traffic congestion, drivers are frustrated and the likelihood of accidents is higher[43].

Finally, dense traffic also incurs costs, for example, in a city of 50,000 inhabitants, which on average has 250 parking spaces, an annual cost of \$216,000 is generated [46]. To solve this problem, it is necessary to work on optimizing urban parking capacities, with one of the key aspects being the application of smart parking systems. In this context, the detection of vehicle presence with an adequate IoT system would represent an efficient solution to the aforementioned problem. Current technological solutions for smart parking vary, from image recognition to detection nodes that are most commonly based on one of the sensor technologies (usually infrared or magnetic sensors). The latter usually give the obtained data on the presence of a vehicle in a parking space for processing by microcontrollers (MCUs), which then transmit them via a radio interface. Therefore, such solutions require appropriate software support for sensor activation and reading, decision-making on parking status, as well as radio communication after the parking status changes. In addition, such devices are usually implemented with the possibility of receiving communication via radio from centralized systems/network access points for updating (e.g. for time synchronization), but also perform online firmware updates. All of the above has an extremely negative impact on the lifespan of battery-powered sensor devices. If we also take into account the additional requirement that the end-user must calibrate the sensor before installation, it is clear that there is a need for a more alternative solution.

A new approach to detecting parking space occupancy could be achieved by adequate implementation of Machine Learning algorithms that could achieve high detection accuracy while providing an economically and energy-efficient solution.

### **1.1.2 Network Layer: Challenges and Opportunities**

Among the technologies enabling IoT, RFID is considered one of the main drivers with applications including access control, parking management, logistics, object localization, people tracking, and retail [47]. In large infrastructures, such as commercial warehouses, reading RFID tags, particularly those of ultra-high frequency (UHF), is expensive and can involve a large amount of data [48]. A classic RFID system consists of an RFID reader, connected to a computer that controls its functions, and RFID antennas that are connected to the outputs of the RFID reader and used for communication with RFID tags. In RFID, tags are inexpensive devices consisting of a printed antenna and an integrated circuit (IC) that can be powered by batteries or by the principle of backscattering, using energy from the carrier signal sent by the reader [49].

Among existing technologies, passive Gen2 technology at UHF frequencies is consid-

ered the most attractive for use in IoT systems due to its simple design, flexibility, cost, and performance [50, 51]. Gen2 uses a standard physical layer, medium access control (MAC) layer, and network and application layer of the ISO/OSI architecture to establish reliable communication between the reader and the tag [52]. Since passive tags do not have a battery, it is necessary for the reader to provide enough energy to power the tags and enable them to respond with the requested information. The amounts of energy that tags can collect are small, so they cannot afford energy-inefficient MAC schemes. Passive RFID systems use Dynamic Framed Slotted ALOHA (DFSA) media access control protocol as an improved version of pure ALOHA protocol, and Binary Search Tree (BST) algorithm can also be used [53]. DFSA belongs to the group of time-division multiple access (TDMA) protocols, where communication between the reader and the tag is divided into time frames, which are, in turn, divided into time slots. MAC layer limitations in RFID reader communication require the correct selection of the frame size in order to achieve maximum throughput. To set the optimal frame length, it is necessary to estimate the number of tags in the reading area. Choosing the optimal frame length results in shortening the time for identifying a large number of RFID tags, and increasing system throughput. A new way to achieve better throughput compared to the latest scientific achievements could be achieved by the appropriate application of Machine Learning and deep learning models in RFID readers themselves.

### 1.1.3 Application Layer: Challenges and Opportunities

In recent years, the integration of IoT in education has been a growing trend, offering innovative solutions for teaching and learning [54]. IoT technology has the potential to create interactive and immersive learning experiences that can improve student engagement, motivation, and learning outcomes, due to the low-cost functionalities of smart devices [55]. These devices can collect and analyze data to improve educational quality and help educators make informed decisions [56]. As a consequence, this promotes creativity, critical thinking, communication, and collaboration, leading to the development of higher-order thinking skills among learners [57]. Furthermore, the IoT can help bridge the digital divide by providing students with equal access to education regardless of their location or socioeconomic status [58].

As such, the application of IoT technology in children's education is an area of particular interest, as "Things" play an important role in children's lives, given that their typical daily activities focus on manipulating physical materials, such as toys [59].

Modern-day children are commonly referred to as digital natives, as they have grown up with current technology being ubiquitous and seamlessly integrated into their daily lives [60]. They are known for their natural and intuitive ability to interact with technology and use digital devices effectively. This proficiency has revolutionized the way they learn, resulting in new methods and modalities of knowledge acquisition [61]. One major area that has been

impacted by the rise of digital natives is Science, Technology, Engineering, and Mathematics (STEM) education [62]. With the growing importance of technology in almost every aspect of our lives, including IoT applications, the demand for skilled professionals in the STEM field has increased significantly. In response, countries around the world, such as the European Union, are placing a renewed focus on STEM education and revising their school curricula to make it more engaging and relevant for young learners [63, 64]. These curricula emphasize the importance of lifelong learning, particularly emphasizing the acquisition of mathematical and digital skills that promote children’s cognitive development. Therefore, to facilitate meaningful and deeper learning in these areas, future IoT educational applications should be specifically designed to promote the development of abstract mathematical concepts [61, 65]. In this regard, both scientific research and commercial applications have focused on toys with IoT features such as software and sensors, commonly referred to as Smart toys [66]. These toys are characterized by their ability to facilitate two-way interactions between children and toys, using both tangible objects and electronic components. Smart toys offer a unique play experience that differs from traditional toys by providing an interactive environment that promotes general child development [67]. Moreover, as such, they have the potential to aid in the development of thinking and problem-solving skills, particularly in relation to abstract mathematical concepts such as geometry [68]. Although geometry is an essential subject in mathematics, many students struggle to visualize its concepts, which can impede their ability to learn and apply geometric principles effectively in the future [65].

Recent studies emphasize that there are currently limited empirical studies on STEM education in young children [69]. According to a rather novel study, there is little research on how children interact with IoT-based geometry learning systems and how these systems can be effectively integrated into educational settings [70]. In general, additional research is required to evaluate the effectiveness of smart toys in facilitating the learning process [67], while the authors in [71] suggest that the incorporation of such technology has the potential to revolutionize education. Further scientific research is needed to examine how Machine Learning can be effectively incorporated into smart educational toys, and to investigate the different interaction modalities that can be used to optimize their educational potential.

#### **1.1.4 Hypothesis**

The primary objective of this scientific research is to methodically develop a novel principle for detecting, processing, and assessing the state of IoT environments using Machine Learning techniques. The specific application of these methods is demonstrated across all three layers of the three-layer IoT architecture, evaluating various aspects of the performance and applicability of IoT systems. The fundamental aim of this research is to improve the performance of Internet of Things services by uncovering hidden information from the environ-

ment using available data and presenting new solution models based on Machine Learning algorithms.

Hypotheses:

**H1.** *Through the modeling, development, and testing of Machine Learning algorithms that leverage the Received Signal Strength Indicator (RSSI) data from LoRaWAN devices, it becomes feasible to estimate the conditions of the IoT environment from the Perception Layer of the three-layer IoT architecture with high precision.*

**H2.** *By leveraging Machine Learning models to estimate frame size and tag count, it is feasible to enhance the throughput of RIFD Gen2 systems that employ the ALOHA protocol on the Network Layer of the three-layer IoT architecture.*

**H3.** *It is possible to achieve highly accurate detection and interpretation of complex human gestures for interaction on the Application Layer of the three-layer IoT architecture by employing Machine Learning algorithms based on the sensor data output.*

The first hypothesis (H1) is based on the complex analysis of data from the Perception Layer of the three-layer IoT architecture, which establishes a correlation between the received signal strength indicator (RSSI) of LoRaWAN devices and changes in IoT environments, implying the possibility of estimating changes in IoT environments through software. By modeling, developing, and testing Machine Learning algorithms with high accuracy from the signal strength, it is possible to classify parking space occupancy and estimate soil moisture levels. Through complex statistical tests, the performance and efficiency of these Machine Learning models will be demonstrated. Based on the results obtained, it is possible to redesign the sensor device, extending its lifespan and simplifying its hardware, thereby minimizing the overall cost of the sensor device. Moreover, the results suggest that the privacy of parking space users may potentially be compromised, as signal strength data can be collected from a great distance and can be misused by using machine and deep learning models.

The second hypothesis (H2) assumes that it is possible to increase the throughput in RIFD Gen2 systems that use ALOHA protocol at the MAC layer by using machine and deep learning models to estimate frame size and tag number. Modeling and feasibility analysis of applying Machine Learning models to select the optimal frame length should result in a shorter time to identify a large number of RFID tags and increase the system's throughput. The performance of the Machine Learning model is comparable to state-of-the-art algorithms, demonstrating the possibility of achieving better throughput with Machine Learning models. Moreover, experimental results will show that these models can be implemented on modern microcontrollers with limited resources to maximize tag identification and throughput, while achieving adequate execution time to meet protocol requirements.

The third hypothesis (H3) is based on the assumption that complex human gestures can be recognized with high accuracy using machine and deep learning algorithms based on output sensor data. In order to design an educational smart interactive toy for learning geometric shapes and improving motor skills in preschool and elementary school children, the perfor-

mance of two sensor technologies will be tested and compared. This interactive toy, as an advanced interface, will include components for detecting and correctly interpreting complex gestures that form geometric shapes using sensor technology and Machine Learning models. Modeling, developing, and testing Machine Learning algorithms for recognizing complex human gestures will be based on output sensor data from capacitive and infrared sensors with different ranges. It will be shown that it is possible to recognize three complex human gestures that form geometric shapes with high detection accuracy.

## **1.2 Scientific methodology and contributions**

In accordance with the objectives of the study stated earlier, the research is divided into a Theoretical and Empirical part.

The Theoretical part of the research involved a review of relevant scientific literature in the field with the aim of identifying existing models and approaches to detection, processing, and assessment of conditions in IoT environments across all layers of the three-layered IoT architecture. The aim was also to identify any unexplored aspects of these approaches. Existing problems and guidelines from the scientific literature will be identified, and a more detailed analysis of research that has described technologies and the types of collected data, their specific processing, and the development, testing, and evaluation of individual Machine Learning models will be conducted.

The Empirical part of the research involved testing the hypotheses that were developed based on the findings of the Theoretical part. The research involved collecting data from various sources, such as IoT devices and sensors, and processing the data using Machine Learning algorithms to detect and evaluate the state of the IoT environment. The collected data was analyzed to identify patterns and trends, and to develop models for predicting and improving the performance of IoT services. The performance of the developed models was evaluated using various metrics, such as accuracy, precision, and recall. The results of the empirical analysis were used to validate the proposed approach and to provide recommendations for further research in this area. In accordance with the previously stated hypotheses, the empirical research is divided into three parts, depending on the layer of the three-layer IoT architecture in which it was conducted. All computations for this research study were conducted on a dedicated laptop computer. The laptop used for these computations featured an Intel(R) Core(TM) i7-7700HQ processor running at 2.80 GHz, 16 GB of RAM, and an NVIDIA GeForce GTX 1050 Ti graphics card with CUDA capabilities. The laptop operated on a 64-bit Windows 10 operating system. To optimize computational efficiency, the NVIDIA CUDA deep neural network library (cuDNN) was utilized. The entire codebase for this research study was implemented in Python 3.8 using Tensorflow 2.2.0.

### 1.2.1 Theoretical part of the research

The Theoretical aspect of this research involved conducting a comprehensive review of scientific literature in the field. The primary objective was to identify existing models and approaches related to the detection, processing, and assessment of conditions in IoT environments across all three layers of the IoT architecture. Furthermore, the aim was to uncover any unexplored aspects or gaps in these existing approaches. By analyzing relevant research, the study aimed to identify existing challenges, guidelines, and insights from the scientific community. Additionally, a detailed examination was conducted on research that described various technologies, data collection methods, data processing techniques, and the development, testing, and evaluation of Machine Learning models.

Hence, a new approach for uncovering hidden information from the environment using available information through the individual layers of the three-layered IoT architecture is presented, representing new solution models based on Machine Learning algorithms in order to enhance the performance of Internet of Things stack services. This concept (illustrated in Figure 1.4) sets a framework for the empirical part of the research, which will test the hypotheses.

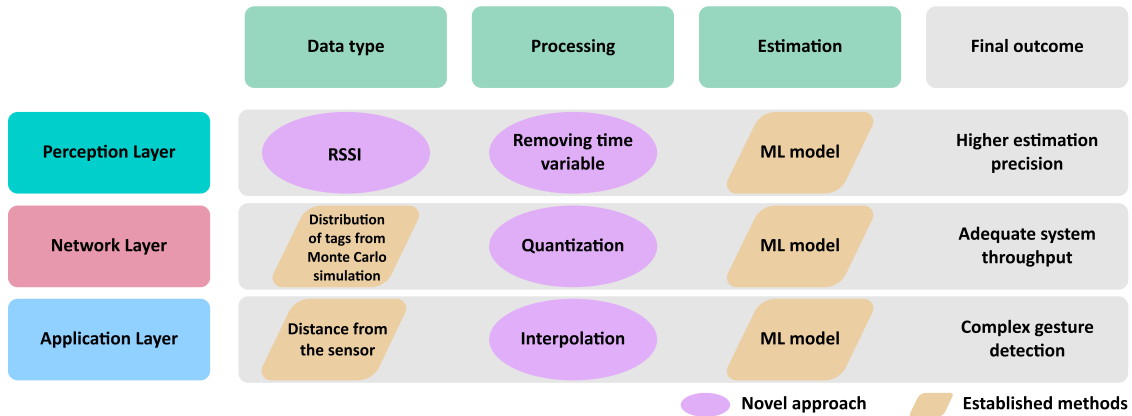


Figure 1.4. Conceptual framework for detection, processing, and state estimation in IoT environments based on Machine Learning models.

### 1.2.2 The Empirical part of the research: Perception Layer

In this part of the research the technology of Low Power Wide Area Networks was employed, more precisely LoRa, to convey data over the radio from sensor device to the base station. LPWANs such as LoRaWAN allow battery-operated sensors or "things" to communicate low throughput data over long distances with minimal infrastructure deployment. To meet the needs of overgrowing IoT demands and applications, especially in terms of lower consumption, cost-effectiveness and long communication range and distances, LPWANs have been considered as the ultimate solution [72]. To this day, LPWAN has been directed to

accomplish all of the emerging IoT demands and has broadly been classified as unlicensed and licensed technology [73]. The LPWANs obtain low power and wide coverage range by using the sub-1 GHz unlicensed, industrial, scientific and medical frequency band, high processing gains, narrow bandwidths, and by periodically transmitting packets at low data rates [74]. The LoRa alliance patented the LoRa (standing for long range), a spread spectrum modulation scheme that utilizes Chirp Spread Spectrum (CSS) modulation and which exchanges data rate for sensitivity within a fixed channel bandwidth [75]. It operates within the sub-Gigahertz unlicensed spectrum ISM bandwidths, namely for USA: 915MHz, for EU: 433MHz and 868MHz having a standardized MAC protocol, LoRaWAN, that determines the communication protocol and system architecture of the network for which the LoRa physical layer applies direct sequence spread spectrum with multiple spreading factors (SF) that enable the long-range communication link [76]. Figure 1.5 depicts the LoRa and LoRaWAN protocol stacks.

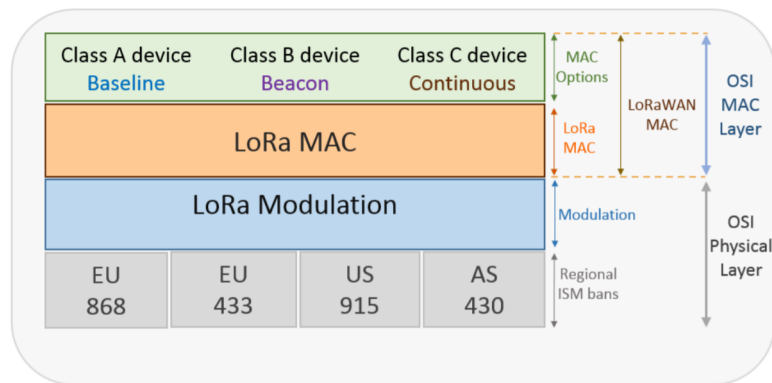


Figure 1.5. LoRa and LoRaWAN protocol stack [77].

LoRa applies six different spreading factors (SF7 to SF12) allowing the adaptation of the data rate and range and thus making it highly resilient to interference, where the generated signal has low noise levels and is difficult to detect or jam [78]. Precisely, a higher spreading factor enables a longer transmission range but at the expense of lower data rate, and vice versa, where the LoRa data rate ranges from 50bps and 300kbps depending on the spreading factor and channel bandwidth [79]. LoRaWAN does not allow device-to-device communications working mainly in the uplink and, if need be, network servers can send downlink data and control packets to end devices [80]. Different functionalities for bidirectional communications lead to the definition of three main classes of LoRaWAN devices, which have different capabilities to cover a wide range of applications and where each class constitutes a trade-off between battery life and network downlink communication latency [81].

LoRaWAN exhibits a star-of-stars topology, depicted in Figure 1.6, made out of LoRa modules (end-devices), one (or more) LoRa gateways; and a central network server where the gateway devices relay messages between end-devices and a central network server [79]. The central server further transmits the received packets to the application server, which then

processes them for further application usages.

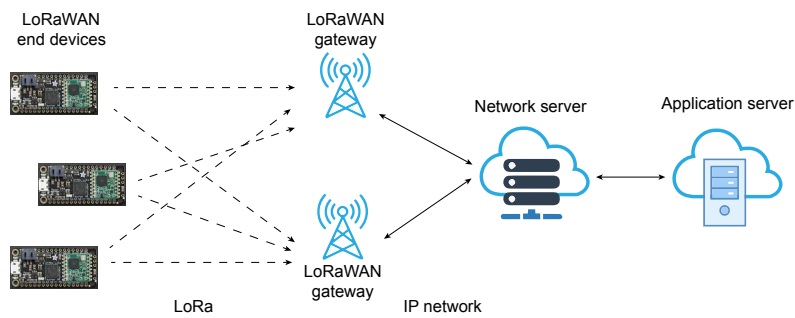


Figure 1.6. LoRaWAN (Long Range Wide Area Network) architecture.

Scalability is a prominent advantage of LoRaWAN, as it has the capability to accommodate a vast number of devices, ranging from thousands to millions, depending on various factors. These factors include the specific scenario, the average rate of message transactions, the average size of transmitted messages, and the number of LoRa channels employed [77]. Another crucial advantage of LoRa it lies within the fact that a single base station can cover hundreds of square kilometers [76]. However, the range may depend on the environment or obstructions.

LoRaWAN will reduce the nodes cost, providing extended battery lifetime and increase the capacity of the network, making it suitable for WSN that requires low communication range, low energy consumption and low data rate [82]. What is more, although LoRaWAN ensures data rates from 0.3 kbps up to 50 kbps, with the maximum payload length for each message of 243 bytes, which is considered sufficient for transmission of real-time sensor data in the IoT, Machine-to-Machine or industrial applications, the transmission of real-time image data, or anything that requires high bandwidth may not be suitable on LoRa networks [76]. LoRa technology has been compared in-depth with other LPWAN technologies in terms of architecture, battery lifetime, network capacity, device classes and was rated as advantageous, but, interestingly, the security issues in LoRa were mentioned repeatedly in several studies, pointing out to the potential security vulnerabilities of LoRa which may expose the LoRa network to jamming attacks [76].

### Soil Humidity sensing

The first part of the research on the Perception Layer of the three-layer IoT architecture establishes and analyzes the concept of a new soil moisture estimation system that is cost-effective and energy-efficient, based on LoRaWAN technology and Machine Learning algorithms. An I2C soil moisture sensor based on LoRaWAN technology was implemented for this purpose, and it monitored soil moisture and temperature over a period of several months. The sensor device was buried 14 cm under the ground with antenna vertically oriented, within the range of two LoRaWAN gateways as shown in Figure 1.7.



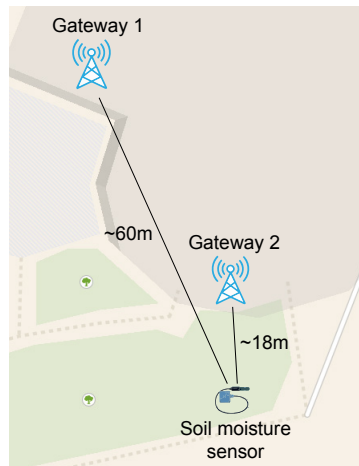


Figure 1.7. Implementation of LoRaWAN-based soil moisture sensing device.

Once the message has reached the base stations, it is forwarded to the TTN network and application server as depicted in Figure 1.8. Additionally, TTN allows the sender to forward messages from its infrastructure to servers using the MQTT protocol. The collected sensor data contained information about RSSI, SNR, soil temperature, soil moisture level, timestamp, and LoRaWAN gateway ID. The majority of data was collected within the months of November and December, for a sampling rate of 5 minutes. In the rest of the text, the RSSI and SNR of Gateway 1 and Gateway 2 will be referred to as  $RSSI_1$  and  $SNR_1$ , and  $RSSI_2$  and  $SNR_2$ , respectively.

An exploratory data analysis was performed to discover anomalies in the data, define the required data preparation approach, and identify ML algorithms that might help predict desired level of soil moisture.

Key characteristics of raw data variables were tracked by observing changes of signal strength in contrast to soil moisture over time. As can be observed from Figure 1.9, the change of RSSI is rapid, whereas the soil moisture alters gradually.

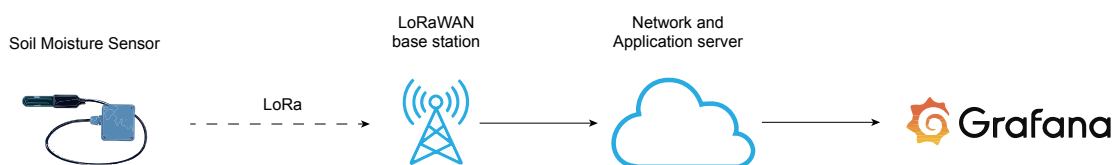


Figure 1.8. Network architecture of LoRaWAN-based soil moisture sensor system.

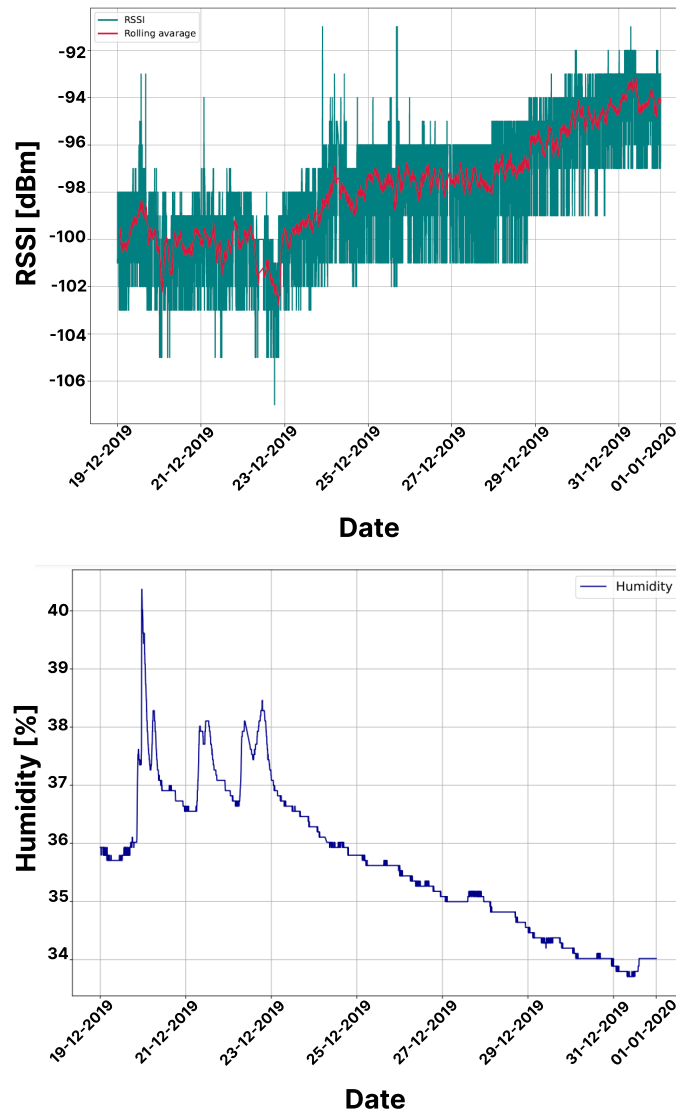


Figure 1.9. RSSI from Gateway 2, and soil moisture during the winter period.

A trend can be noticed, where the increasing moisture affects the signal strength causing its decrease, and vice versa. As a result of channel stochastic behaviour there are two major fading components of the received signal strength. One is the swift variation in signal strength due to the multipath propagation whilst the other is its slower variant and mostly a result of the signal reception in the radio shadow of large obstacles [83].

Hence, the raw data were smoothed by decomposing the received signal strength into long-term and short-term fading factors using a 2-hour time window. The long-term factor component was calculated by taking 24 samples of raw RSSI and SNR data, calculating their mean, and subtracting it from the raw values. Through the process of data aggregation, smoothed RSSI values were allocated to distinct soil moisture percentage classes, thereby eliminating the time variable and enabling the investigation of the potential correlation between RSSI values and specific soil moisture percentage classes in a more controlled and rigorous manner.

Pearson correlation coefficients were calculated to investigate the relationship between soil moisture, RSSI, and SNR values. The results are presented in in Table 1.1. As can be seen, there is an inverse correlation between signal strength and soil moisture, with SNR and RSSI from Gateway 2 having a stronger correlation with soil moisture. For data obtained from Gateway 1, SNR had a stronger correlation with soil moisture than RSSI, suggesting that distance affects RSSI more than soil moisture. Furthermore, a specific working-day time-frame was found to have modified the mean value of the signal strength parameter.

*Table 1.1. Pearson correlation matrix between soil humidity and RSSI and SNR.*

	<b>RSSI<sub>1</sub></b>	<b>SNR<sub>1</sub></b>	<b>RSSI<sub>2</sub></b>	<b>SNR<sub>2</sub></b>
Soil humidity	-0.29	-0.81	-0.65	-0.73

This implies that lower values of SNR and RSSI would indicate higher soil moisture as was observed in Figure 1.9. Based on the obtained results, it was concluded that RSSI and soil moisture values were significantly negatively correlated and that a suitable Machine Learning algorithm should be able to encompass the complexity of the data properties detailed in the above analysis.

Two Machine Learning algorithms, Support Vector Regression (SVR) and Long Short-Term Memory (LSTM), were employed as a means of estimating soil humidity. SVR is a supervised learning algorithm that is frequently used in regression analysis. It operates by identifying the hyperplane that maximizes the margin between predicted values and actual values. This technique was chosen for its ability to deal with high-dimensional data and non-linear correlations between variables. On the other hand, LSTM is a type of recurrent Neural Network (RNN) that is suitable for time series prediction. It is useful for modeling long-term dependencies in sequences since it can choose to remember or forget information from previous time steps. The LSTM was used because soil moisture data is inherently time-series data, with each measurement being influenced by preceding measurements. Both models were validated using the same data and in the same manner, as further specified. Rather than interpretation, the models' primary goal was the precise calculation of relative soil moisture based on signal strength. All raw RSSI and SNR data samples captured on two LoRaWAN Gateways, as well as soil moisture, were used in the models. The models were built in three steps: data pre-processing, model definition, and model validation, as described in the following.

Data normalization was a part of data pre-processing, because of the different value scales of variables in the collected data. In general, relative moisture was as a percentage, whereas RSSI and SNR values have been measured in decibels. The models were fed numerical RSSI and SNR values, and the output was a numeric value that estimated relative soil moisture. Furthermore, for model assessment, data were divided into training and test sets in an 80-

20% ratio respectively.

The test set was used to validate the models using two measures. Namely, Loss functions used for estimation of error were Mean Squared Error (MSE) and Mean Absolute Error (MAE) defined with Equations 1.1 and 1.2.

$$MSE = \frac{1}{2m} \sum_{i=1}^m (\hat{y}^{(i)} - y^{(i)})^2. \quad (1.1)$$

$$MAE = \frac{1}{m} \sum_{i=1}^m |\hat{y}^{(i)} - y^{(i)}|. \quad (1.2)$$

A lower MSE indicates greater estimation accuracy. MSE calculates the average squared difference between the estimation and the expected results, whereas MAE calculates the average magnitude of errors across a set of estimations. Furthermore, validation loss reflects how well or poorly the model performs during training.

The SVR model implemented the RBF kernel. The model's input was as follows: for each value of RSSI and SNR at time step  $t$  required for the estimation of moisture at time step  $t$ , values of RSSI and SNR at time step  $t - 1$  were also taken. This provided the model with a "hybrid short-term memory" of previously measured values in time step  $t - 1$ . The model was verified on the test set, yielding losses of  $MSE = 0.0243$  and  $MAE = 0.0487$ . These findings suggested that soil moisture could be estimated with reasonable accuracy based solely on Received Signal Strength and SNR values, even with a limited data set.

Figure 1.10 shows the model's estimation of soil moisture on the test set compared to expected soil moisture values.

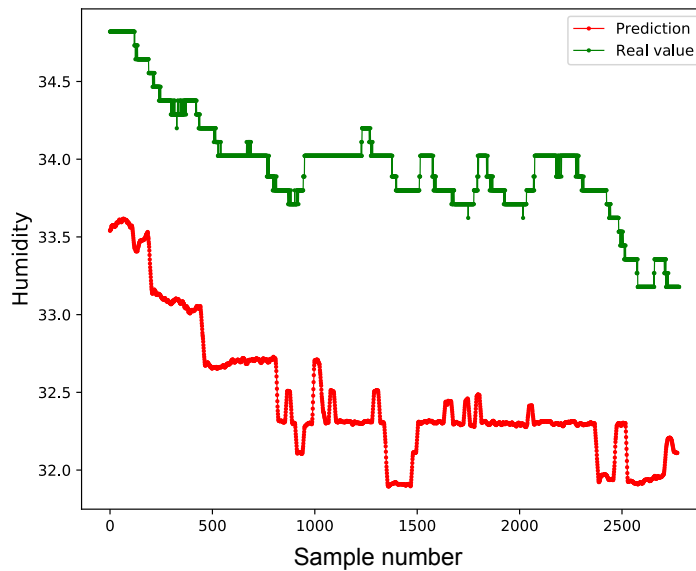


Figure 1.10. Soil moisture estimation using the SVR model on the test set compared to expected soil moisture values

The LSTM model was trained on previously described pre-processed data. In terms of

inputs, a time step of 18 was chosen to approximate 90 minutes of observations (18 samples 5 minutes period) for each estimation. Normalized data was fed into the LSTM model with the goal of estimating relative soil moisture based on signal strength. Several options for number of neurons per layer, learning rates, epochs and different optimizers were tested, and the best results was achieved for 32 neurons, on both LSTM layers with a learning rate of 0.001 and number of epochs 100. Three optimizers were evaluated: Adam, RMSprop, and SGD. RMSprop outperformed the other two optimizers and was chosen as optimizer for the final model design, achieving a  $MSE$  and  $MAE$  errors of 0.00018 and 0.01043, respectively. Figure 1.11 (a) represents the learning path of model with previously described parameters, training and validation loss in relation to each epoch. Estimation of soil humidity with the model compared to expected values of soil humidity that was done on the test set is presented in Figure 1.11 (b).

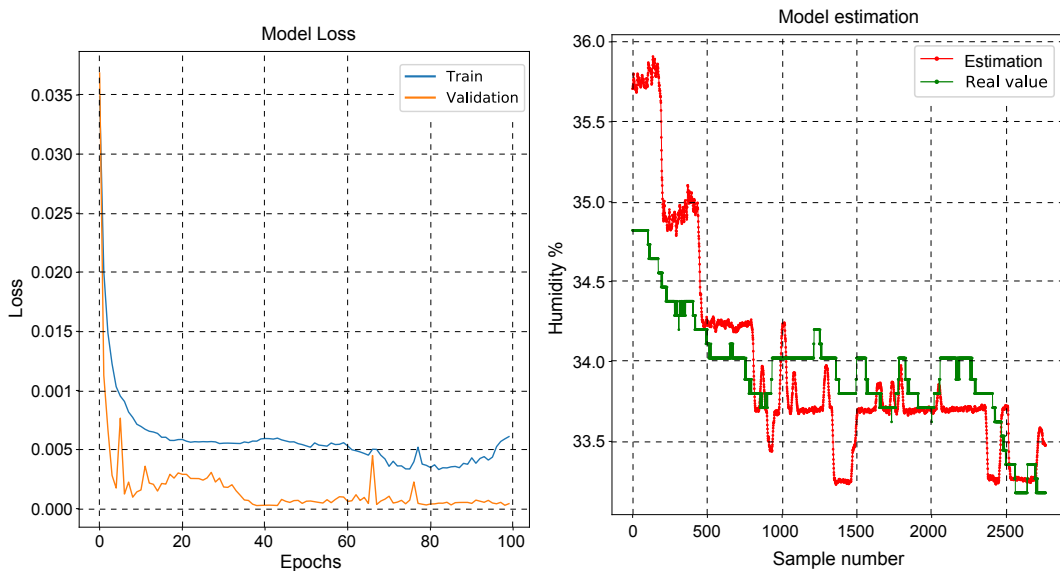


Figure 1.11. (a) Learning path of model with training and validation loss, (b) estimation of soil humidity with the model compared to expected values of soil humidity.

Even with a small and limited data set with only a few months of representative data (winter period), significant results were obtained for the Machine Learning approach for estimating soil moisture from signal strength. Table 1.2 provides a comparison between the estimation training and test time and the previously described errors. As can be observed from the table, although the LSTM model needs more time to train, it achieves a minimum delay between two consecutive estimations based on the testing time. Furthermore, it has lower  $MAE$  and  $MSE$  errors in contrast to the SVR model.

The study found that SVR provided good estimates of soil moisture from RSSI and SNR, validating the Signal Strength Approach and the premise of moisture sensing using just the RSSI and SNR data with Machine Learning. However, the stacked LSTM model using available data obtained significantly more accurate estimates and outperformed SVR, suggesting

Table 1.2. Comparison parameters of ML algorithms for soil humidity estimation

Algorithm	Training time (s)	Test time (s)	MAE	MSE
SVR	1.451	0.821	0.0487	0.0243
LSTM	1385.992	0.668	0.0104	0.00018

that LSTM was better able to encompass the complex correlation between RSSI, SNR, and soil moisture due to its ability to handle time series data.

### Detection of parking space availability

The second part of the research on the Perception Layer of the three-layer IoT architecture aimed to conceptualize and analyze a cost-effective alternative sensor system for parking space detection based on LoRaWAN technology and Machine Learning algorithms. To collect data on parking space occupancy, five Libelium Smart Parking sensors were deployed in a parking zone around the faculty. These sensors are equipped with a magnetometer and radar sensor device, situated at the surface and center of the parking space. They detect changes in occupancy status (car arrival or departure) and transmit this information via radio channel that employs LoRa radio capabilities as a communication peripheral.

Data collection was facilitated using three LoRaWAN gateways placed within the radio range of the sensors. These gateways were distinguished by their location relative to the sensors and environmental setting, as depicted in Figure 1.12. The gateways received occupancy data transmitted by the sensors and forwarded it to a cloud-based application for further analysis. It is worth noting that in addition to sensing changes in occupancy status and transmitting this information, the Libelium sensors also periodically sent keep-alive packets every two hours. The data collected from the sensors and gateways provided a comprehensive data-set on parking space occupancy.

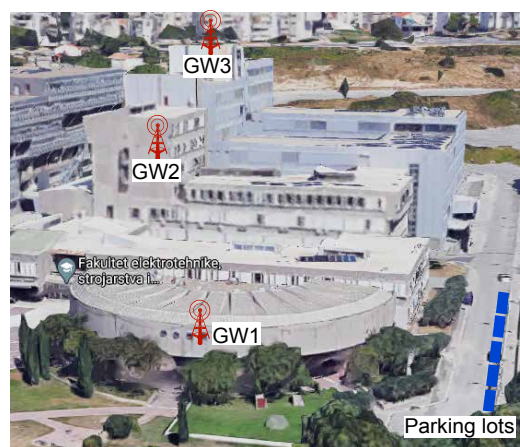


Figure 1.12. Position of parking lots with LoRaWAN Smart parking sensors and location of LoRaWAN gateways that captured information from sensor devices.

The first gateway (GW1) was positioned indoors approximately 30 meters away from the sensors on the first floor of the faculty building, at a height of 4 meters. The second gateway (GW2) was also positioned indoors, but on the fifth floor of the faculty building, at a height of 15 meters and approximately 75 meters away from the sensors on the surface. The third gateway (GW3) was positioned outdoors at the top floor of the building (ninth floor), approximately 145 meters away from the sensors and 30 meters from the ground.

GW1 was installed near the sensors to observe changes in RSSI levels from a short distance during changes in parking lot occupancy. GW2, on the other hand, was placed at a distance from the sensors and had to transmit the signal through numerous obstacles and walls, potentially causing deterioration of signal quality. This placement was chosen to observe the impact of vehicle presence and other obstacles on signal changes. GW3 was positioned outside with minimal obstacles to the radio signal traveling from sensors to the gateway, other than diffraction originating from vehicle presence and the edge of the building. The placement of the three gateways provided diverse observation scenarios for data collection, enabling a comprehensive analysis of the system's performance in different environmental settings.

Over a period of ten months, 130,984 raw data points were collected from the five sensors, and exported into CSV format for subsequent processing with InfluxDB. The data included occupancy information (0 for free parking space, 1 for occupied), timestamps of TTN gateway reception, Received Signal Strength Indicator (RSSI) in dBm, Signal to Noise Ratio (SNR) for each gateway, and sensor and gateway IDs.

The collected data from Libelium Smart Parking sensors was analyzed to identify correlations, anomalies, and suitable Machine Learning algorithms. The analysis was carried out separately for each sensor and gateway due to differences in data reception. The first results showed a skewed data set with a higher frequency of free parking spaces due to the university location. The class ratio for free space ranged from 81.6% to 89.2%, with an average ratio of 85% representing free space and 15% representing occupied parking space.

The second result gave the insight into how are the values of Received Signal Strength and Signal-to-Noise Ratio associated with occupancy status. It was necessary to identify if the specific value of RSSI (or SNR) correlates with the free and occupied parking status from the same sensor and gateway. Therefore, the probability density function for RSSI (as well as SNR) for diverse occupancy status were plotted to gain the needed information.

The distribution of a specific Received Signal Strength Indicator (RSSI) or Signal to Noise Ratio (SNR) value from Gateway 1 (GW1) exhibits a large amount of overlapping as shown in Figures 1.13 a) and b). It is also observed that higher RSSI and SNR values tend to imply a free parking state for all sensors connected to GW1 and GW2. However, the degree of overlapping starts to decrease for GW2. Notably, the data collected from GW3 differs significantly from those of GW1 and GW2, especially for sensors 2, 3, and 4. Figures 1.13 c) and d) show that GW3 has the least amount of overlapping in the distribution of RSSI

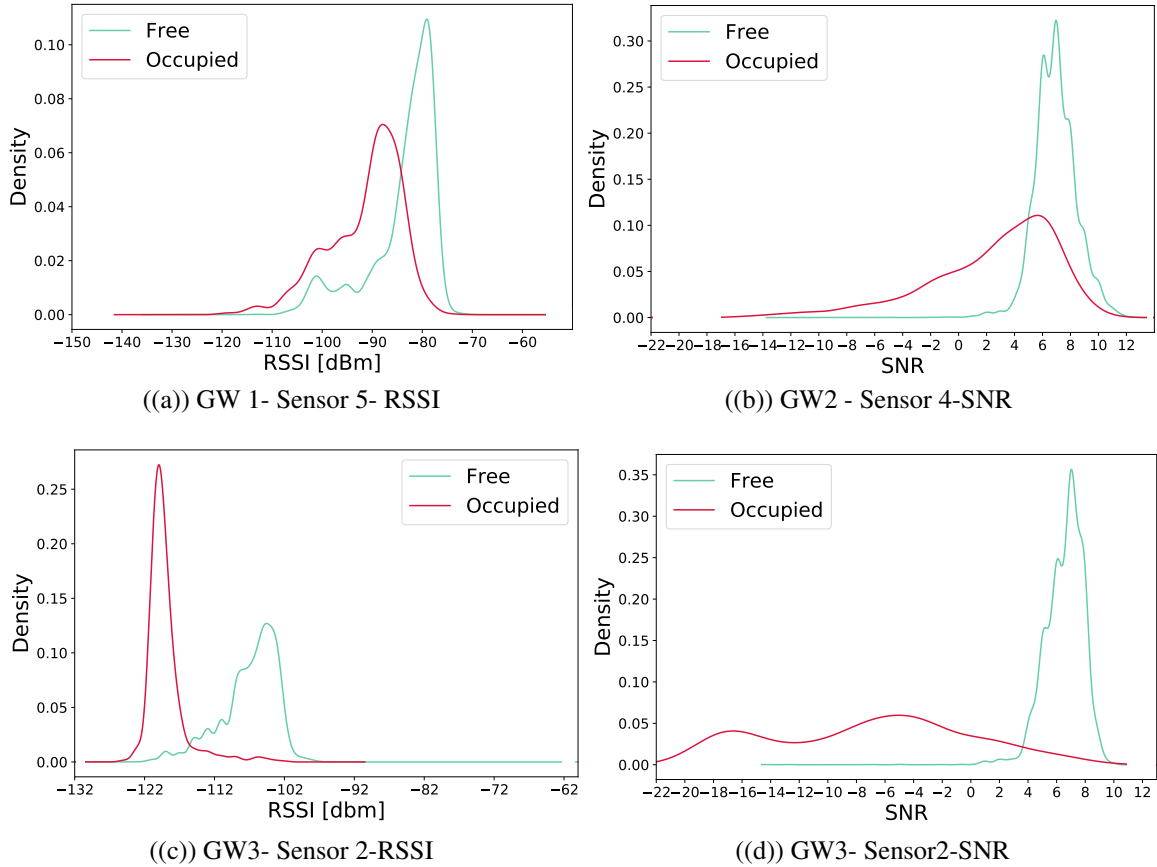


Figure 1.13. Probability density function of RSSI and SNR values alterations for diverse sensors and occupancy status from GW1, GW2 and GW3, respectively.

and SNR values for a particular occupancy state. It can also be observed that lower RSSI and SNR values correspond to occupied parking spaces, whereas higher values indicate a free parking status for GW3. These variations in results among different gateways can be attributed to their distance from the parking sensors. GW1 is the closest to the parking sensors, whereas GW3 is the furthest away. This observation suggests that the channel has a more significant impact on RSSI and SNR than the occupancy state if the gateway is located closer to the parking sensors.

Finally, to address the issue of overlapping RSSI and SNR values in different occupancy states, further analysis was conducted to examine the change of these values when the parking status changes and when it does not. It was found that when the parking status remains the same, there is minimal or no change in the RSSI and SNR values. However, when there is a change in parking status, there is a significant shift in these values. Figure 1.14 shows histograms of changes in RSSI values for sensor 2 from GW3, for instances when parking spaces remain free or become occupied prior to being free again.



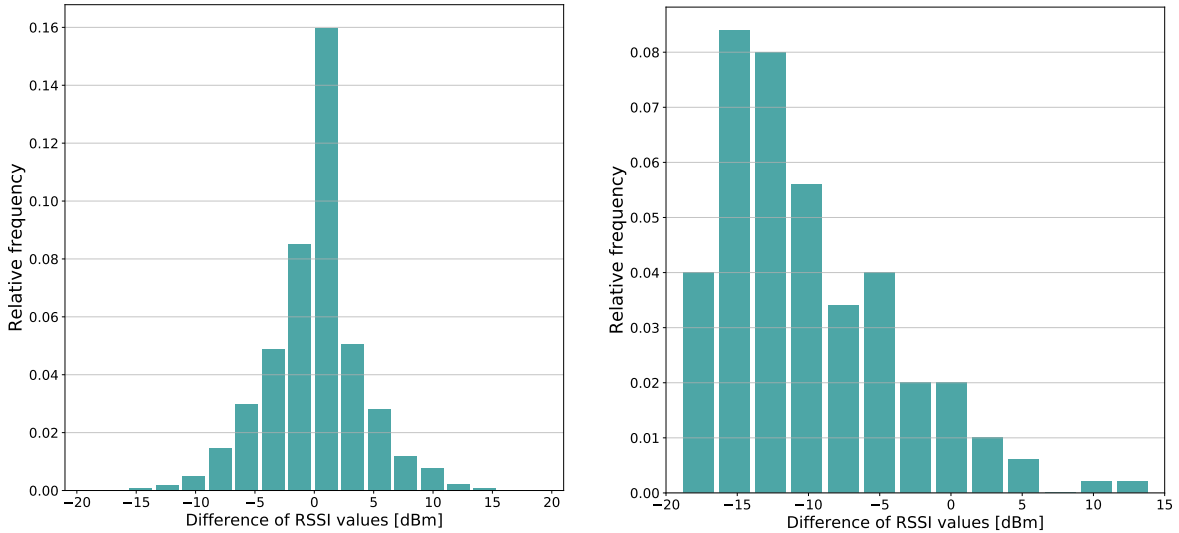


Figure 1.14. (left) Difference of RSSI values when the parking space remains free, (right) Difference of RSSI values for change of state from occupied to free for sensor 2 from GW 3.

Two different scenarios were considered for evaluating Machine Learning models. In the first case, a two-fold approach was employed to evaluate Machine Learning models, specifically the Neural Network (NN) and Markov Model. A critical aspect taken into account was data preprocessing, with a focus on stratification to address class imbalance. Stratification was applied to ensure an equitable distribution of samples from each class during the training and testing phases. This technique aimed to maintain the representative proportions of parking space occupancy states, thereby mitigating any potential bias and improving the overall model performance. Through iterative steps, the performance of the Neural Network and Markov Model was assessed. Notably, the Neural Network exhibited superior predictive accuracy compared to the Markov Model. This outcome underscored the Neural Network's ability to capture complex relationships within the data, leading to more accurate predictions of parking space occupancy. In the subsequent scenario, the evaluation expanded to include the Neural Network and Random Forest (RF) models. To address class imbalance, the Synthetic Minority Over-sampling Technique (SMOTE) was incorporated, which artificially generated synthetic samples to increase the representation of minority class instances. This step aimed to enhance the training process and mitigate potential biases introduced by imbalanced class distributions. Furthermore, for every sensor, all data about RSSI, SNR, and occupancy status from all three gateways were associated. The models were built and tested for each sensor separately. In the pre-processing of data for Neural Network models, the data was normalized since diverse variables have different value scale. Moreover, it was decided to incorporate the Time variables into the data, since Time variables can encompass effects like temporal dependence and seasonality, giving a deeper insight into occupancy history. Therefore, for each sensor, hour, day, month, and day of the week for a specific occupancy were included as a feature. Due to the nature of our data, the data split was done in 70% : 30% ratio, where 70% of data was taken for training and 30% for testing, with the

target values being occupancy status and all other values were given as input. Through this refined approach, the Random Forest model, coupled with the SMOTE algorithm, exhibited notable improvements in predictive accuracy. The integration of the Random Forest model and SMOTE technique provided a more robust and balanced framework for estimating parking space occupancy.

Performance of the classifiers has been evaluated in terms of evaluation of their different characteristics. Therefore, Accuracy, Area under the Receiver Operating Characteristic Curve Accuracy (ROC AUC) and F1 score, were used for evaluation. Mathematically, the evaluation metrics derive as follows:

- Let TP and TN be the number of positive and number of negative class that a correctly classified, respectively. Let FP and FN be the number of positive and negative class that are miss-classified, respectively. Accuracy is the proportion of correct predictions that the model makes is given by the formula:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}, \quad (1.3)$$

- F1-Score calculated as from the Precision (Pr) and Recall (Rcl) as their harmonic mean using formula:

$$F1 \text{ score} = 2 \cdot \frac{Pr \cdot Rcl}{Pr + Rcl}, \quad (1.4)$$

where  $Pr = TP/(TP + FP)$  and  $Rcl = TP/(TP + FN)$  [84].

- A ROC graph is a probability curve that illustrates a relative trade-off between TP and FP over a range of various thresholds of a classification model. A good classifier should have the ROC curve positioned as close as possible to the upper left corner of the diagram, in contrast to a poor classifier whose ROC curve is set along the main diagonal [85].
- AUC gives a measure of how much a ROC curve leans near the perfect classification point, that is, the point (0,1) on the ROC plot, i.e., the ability of the classifier to differentiate classes [86].

### First scenario

To capture the temporal dependencies and memory of occupancy changes, a second-order Hidden Markov Model (HMM) was employed. The HMM considered the Hidden States as the target variable, representing the occupancy status, and the Observable (Visible) States as the changes in RSSI values between consecutive occupancy states.

The Hidden States, denoted as  $FF$ ,  $FO$ ,  $OF$ , and  $OO$ , retained the memory of occupancy by representing different combinations of the previous and current occupancy states. The HMM, denoted as  $\lambda(A, B, \pi)$ , consisted of the transition matrix  $A$ , which captured the prob-

abilities of transitioning between states, the initial state distribution  $\pi$ , and the observation probability distribution matrix  $B$ .

The architecture of the second-order HMM for occupancy status detection based on RSSI value changes is illustrated in Figure 1.15. The model was implemented considering data from each sensor and gateway separately, allowing for comprehensive analysis.

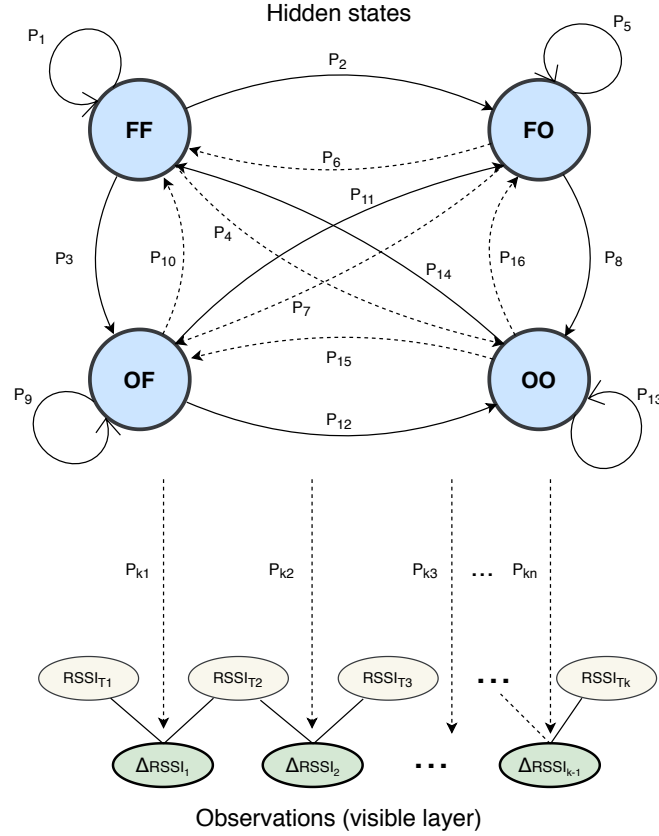


Figure 1.15. Illustration of second-order Hidden Markov model for detecting occupancy status based on change of RSSI values

The Viterbi algorithm was employed to determine the optimal sequence of states based on the observation sequence and the model. Subsets of consecutive values were used as observations and fed into the Viterbi algorithm with a specified step length. The specific form of the applied algorithm is as follows. Let  $\lambda(A, B, \pi)$  be a HMM and  $O = (o_1, o_2, \dots, o_T)$  given observations. The Viterbi algorithm finds single best state sequence  $q = (q_1, q_2, \dots, q_T)$  for the given model and observations. The probability of observing  $o_1, o_2, \dots, o_t$  using the best path that ends in state  $i$  at the time  $i$  given the model  $\lambda$  is:

$$\delta_t(i) = \max_{q_1, q_2, \dots, q_{t-1}} \mathbb{P}(q_1, q_2, \dots, q_{t-1}, q_t = i, o_1, o_2, \dots, o_t | \lambda) \quad (1.5)$$

$\delta_{t+1}(i)$  can be found using induction as:

$$\delta_{t+1}(i) = b_j(o_{t+1}) \max_{1 \leq j \leq N} [\delta_t(j) a_{ji}] \quad (1.6)$$

To return the state sequence, the argument that maximizes Equation 1.6 for every  $t$  and every  $j$  is stored in an array  $\psi_t(j)$  [87].

The accuracy of the model was assessed by comparing the classified values with the true values, utilizing the accuracy score function. Subset accuracy, where the set of classified labels must precisely match the set of true labels, was used to evaluate the model’s performance. Mean Absolute Error (MAE) was also calculated as an additional evaluation metric. The model was tested for all variables from various sensors and gateways, and the best results are provided in the corresponding Table 1.3.

Table 1.3. Table of best results using the HMM model obtained for each gateway

Gateway	Variable (sensor number)	Accuracy (best results)	MAE
GW1	RSSI (4)	87%	0.30
GW1	SNR (4)	87%	0.35
GW2	RSSI(3)	89%	0.27
GW2	SNR (3)	92%	0.20
GW3	RSSI (2)	93%	0.17
GW3	SNR (2)	96%	0.11

The Neural Network applied in this study has undergone a data preprocessing step, including normalization of variables, to account for their different scales. The implemented NN consists of two hidden layers as depicted in Figure 2.3. The input layer incorporates data such as the sensor ID, RSSI, SNR of the LoRa packet transmitted from the sensor to the Gateway, gateway ID, and the corresponding timestamp. The output layer is responsible for predicting the occupancy status of the parking space, indicating whether it is free or occupied.

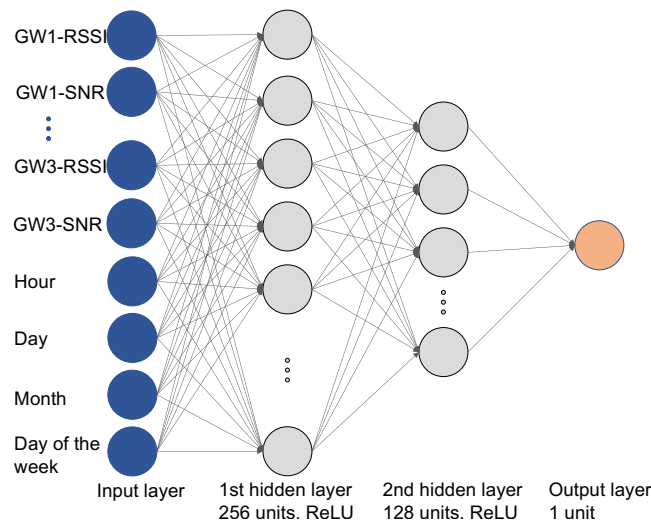


Figure 1.16. Architecture of Neural Network model for parking space occupancy classification

To address the class imbalance observed in the data set, stratification was employed during the data splitting process. This ensured that the distribution of classes was maintained in the training, validation, and test sets. The test set size was set to 10%, and further stratification was applied to split the training set into train and validation sets, with a 10% validation set size. The Neural Network utilized different optimizers, including Adam, RMSprop, and SGD, to optimize the model's performance. Furthermore, various hyperparameters were tested, such as the number of neurons in the layers, learning rate, number of epochs, as well batch size and are presented in Table 1.8. The evaluation of these hyperparameters helped fine-tune the model and improve its accuracy.

Table 1.4. Selection of the hyper parameters for Neural Network evaluation.

Hyper parameter	Values
Number of neurons	Layer1 - 256, Layer2 - 128
Learning rate	0.001 , 0.01
Number of epochs	50, 100, 150
Batch size	64

The NN model with Adam optimizer achieved the best performance with a learning rate of 0.001 and 100 epochs, which is consistent with our previous results. This configuration resulted in an accuracy of 96% on the validation set and 95% on the test set. Additionally, the model achieved an AUC of 96% on both the validation and test sets. The ROC curve is depicted in Figure 1.17 and it demonstrates that the Neural Network classifier, with the Adam optimizer, achieved a high true positive rate (TPR) and maintained a low false positive rate (FPR). The high AUC value of 98% indicates that the Neural Network model effectively distinguishes between occupied and free parking spaces.

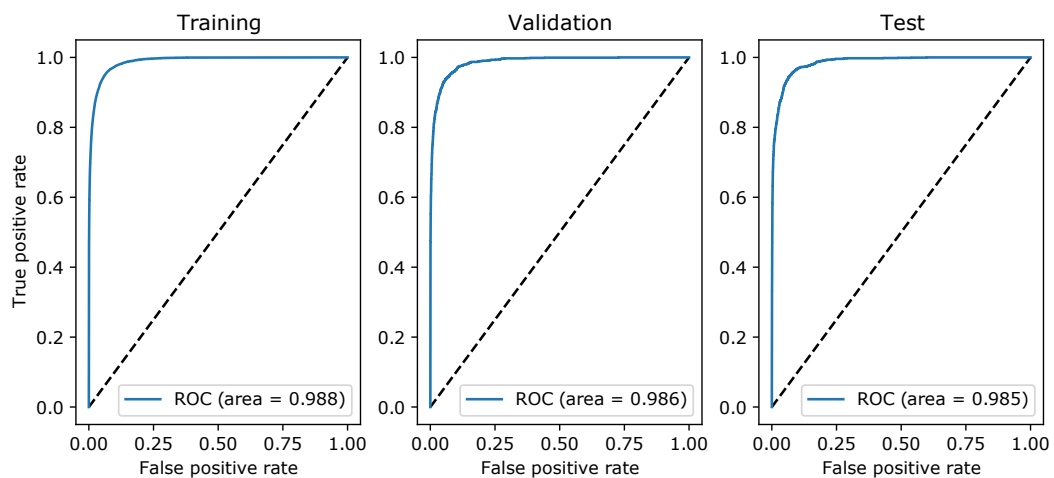


Figure 1.17. ROC curves for Adam optimizer with the learning rate of 0.001 and 100 epochs.

### Second scenario

In this scenario, both the Random Forest and Neural Network models were utilized to perform the classification of parking occupancy. The data used for training these models underwent a SMOTE preprocessing technique, and for every sensor the models were built and evaluated separately.

RF is a powerful ensemble learning method that combines multiple decision trees to make predictions. It is widely used for its ability to handle complex relationships between variables and produce robust and accurate results. One of the main advantages of RF is its good performance and relatively simple implementation [88], but one must regard hyper-parameters and tuning strategies to achieve the best possible classification accuracy.

Therefore, to optimize the RF model's performance, a thorough hyperparameter tuning was conducted. Key hyperparameters, such as the number of trees (`n_estimators`), splitting criterion (e.g., Gini impurity or entropy), maximum tree depth (`max_depth`), maximum number of features considered for splitting (`max_features`), and minimum samples required for node splitting (`min_samples_split`), were carefully selected and tuned. To find the best hyperparameter values, a grid search technique using the `GridSearchCV` class from the `scikit-learn` library was employed. This search was done for each sensor and the tested hyper-parameters are presented in Table 1.15.

Table 1.5. Result of grid search of Hyper-parameters for a particular sensor.

Sensor	<code>n_estimators</code>	critierion	<code>max_depth</code>	<code>max_features</code>	<code>min_samples_plit</code>
Sensor 1	100	gini	20	sqrt	2
Sensor 2	200	entropy	20	sqrt	2
Sensor 3	200	entropy	20	auto	4
Sensor 4	150	gini	20	sqrt	2
Sensor 5	150	gini	20	auto	2

Finally, for each of the sensors Random Forest model was trained based on the above presented hyper-parameters, with the previously described data and evaluated with previously described evaluation metrics. The Random Forest model demonstrated strong performance across all five sensors when evaluated on the test set, as indicated by the results presented in Table 1.6. The model achieved high accuracy scores and F-scores, indicating its ability to ac-

Table 1.6. The results for Random Forest model.

	TEST		
	Acc.	F-Score	AUC
Sensor 1	0.970	0.902	0.990
Sensor 2	0.983	0.923	0.995
Sensor 3	0.978	0.900	0.995
Sensor 4	0.985	0.939	0.995
Sensor 5	0.962	0.891	0.988

curately classify parking occupancy. Notably, these results were obtained on an imbalanced

test set, maintaining the original class distribution. The findings align with the data analysis, where Sensor 2 and Sensor 4 showed the least overlap in RSSI and SNR values for different occupancy states. These sensors yielded the best overall results, with Sensor 2 achieving an accuracy of 98.3% and Sensor 4 achieving 98.5%.

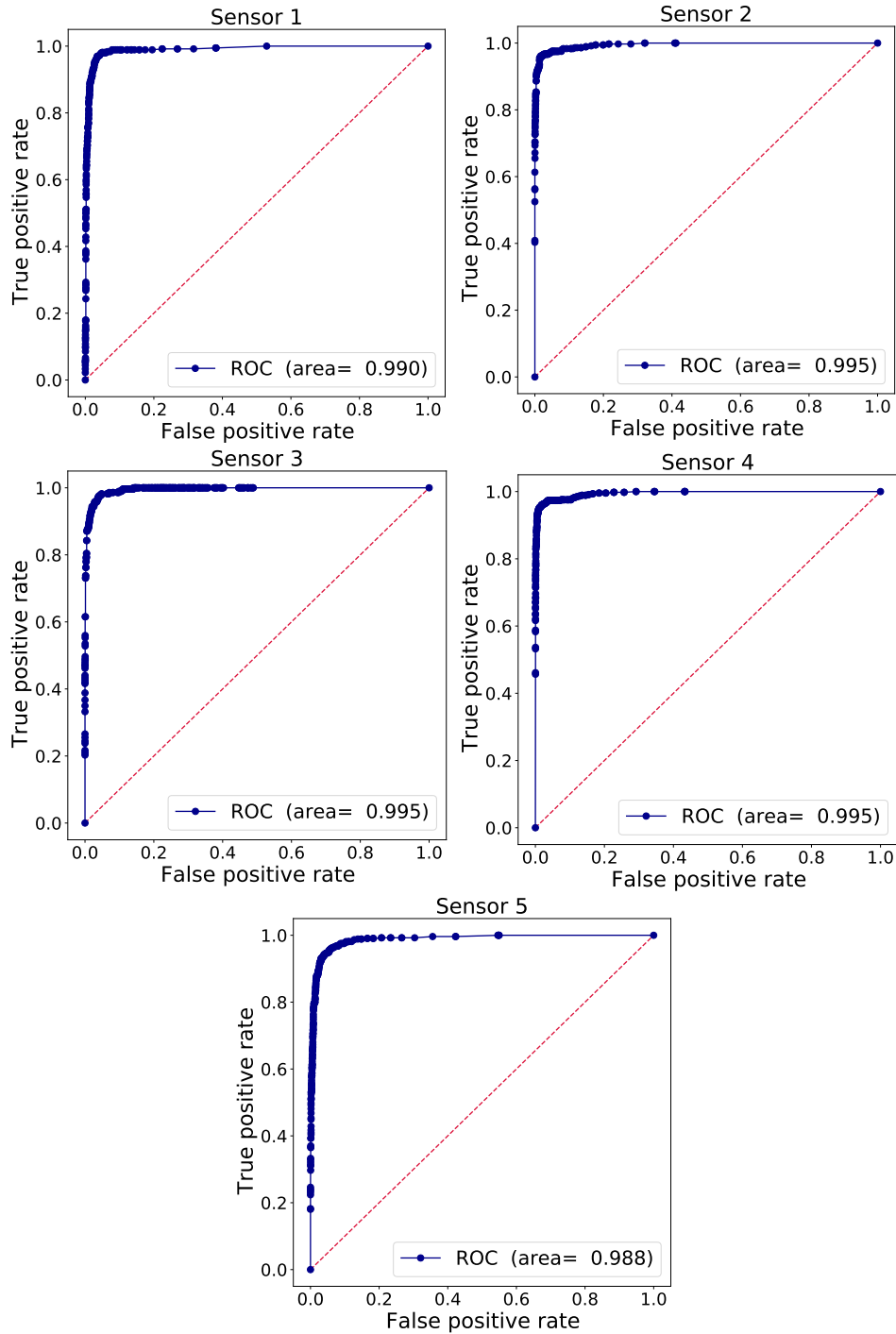


Figure 1.18. ROC curves for Sensor 1, 2, 3, 4 and 5 respectively for Random Forest model.

The ROC curves depicted in Figure 1.18 further support the model's performance, with high AUC scores across all sensors. Sensor 2 and Sensor 4 exhibited the best discrimination ability, with AUC scores of 99.5% and 99.4% respectively, confirming the model's profi-

ciency in distinguishing between occupancy classes.

To further confirm the obtained results and reasoning about the correlation between the change in parking occupancy and RSSI and SNR, one final computation was done, i.e. calculation of feature (variables) importance. Feature importance rates how important each feature is for the decision a tree makes. It results in a number ranging between 0 and 1 for each feature and all feature importance sum up to 1, where 0 means “not used at all” and 1 means “perfectly predicts the target” [89]. These results are presented in Table 1.7. As is shown in the table, the most important features are RSSI from GW1, followed by SNR values from GW3 and RSSI from GW3. These results confirm the importance and strength of RSSI and SNR values in obtaining the information about parking occupancy.

Table 1.7. Feature importance rate for each of the five sensors.

Sensors	GW1		GW2		GW3		month	day	hour	day of the week
	RSSI	SNR	RSSI	SNR	RSSI	SNR				
Sensor 1	0.31	0.02	0.17	0.06	0.04	0.13	0.27	0.03	0.16	0.04
Sensor 2	0.23	0.02	0.09	0.07	0.18	0.22	0.03	0.04	0.09	0.04
Sensor 3	0.21	0.02	0.07	0.04	0.20	0.26	0.07	0.04	0.06	0.03
Sensor 4	0.25	0.02	0.07	0.07	0.13	0.31	0.02	0.04	0.07	0.04
Sensor 5	0.26	0.03	0.03	0.03	0.08	0.17	0.05	0.08	0.22	0.08

The second Neural Network model architecture utilized in this study consisted of two hidden layers, employing the Rectified Linear Unit (ReLU) activation function in the hidden layers and the Sigmoid activation function in the output layer. The incorporated data were balanced using the SMOTE technique. The input layer incorporated information such as Sensor ID, RSSI, SNR values from three LoRaWAN gateways, and the event timestamp (month, day of the week, hour). The output layer predicted the parking occupancy status, with 0 representing a free space and 1 representing an occupied space. To address the binary classification problem, the Binary Cross-Entropy Loss function was utilized with different optimizer combinations. The aim of the optimizer was to help the model converge and minimize the loss or error function. The Neural Network model in this study employed Stochastic Gradient Descent (SGD), Root Mean Square Propagation (RMSProp), and Adaptive Moment Optimization (Adam) as the optimizers. Table 1.8 summarizes the hyper-parameters utilized for building the model.

Table 1.8. Hyper parameters selected for Neural Network model performance testing.

Hyper Parameter	Values
Number of neurons	First layer—256, Second Layer—128
Learning rate	0.001 , 0.01
Number of epochs	50, 100
Batch size	64
Optimizer	Adam, SGD, RMSProp



The best results for hyper parameters were obtained for learning rate 0.001 and 0.01 and 100 epochs for Adam optimizer, summarized in Table 1.9.

*Table 1.9. Best results obtained for NN model*

	l. rate	TRAINING			TEST			
		Acc.	F-Score	AUC	Acc.	F-Score	AUC	
Adam	Sensor 1	0.001	0.987	0.987	1.000	0.961	0.880	0.988
	Sensor 2	0.001	0.996	0.996	1.000	0.973	0.886	0.985
	Sensor 3	0.01	0.993	0.993	1.000	0.975	0.882	0.989
	Sensor 4	0.001	0.995	0.995	1.000	0.977	0.906	0.989
	Sensor 5	0.01	0.978	0.978	0.997	0.949	0.859	0.978

The highest Accuracy and AUC scores were consistently achieved for parking sensors 2, 3, and 4, which aligns with the findings from the data analysis. For these sensors, the Adam optimizer yielded Accuracy values of 97.3%, 97.5%, and 97.7% on the test set, respectively. Sensor 3 and Sensor 4 also demonstrated high AUC scores of 98.9%, indicating strong differentiation between classes. These results are consistent with the Random Forest model's performance and the conclusions drawn from the data analysis. On the other hand, Sensor 5 exhibited lower Accuracy and AUC scores of 94.9% and 97.8%, respectively, further supporting the alignment between the Neural Network model and the Random Forest model, as well as the data analysis conclusions. The ROC curves in Figure 1.19 demonstrate the model's ability to achieve a high True Positive Rate while maintaining a low False Positive Rate for all parking sensors. This indicates excellent classification performance for both occupied and free parking spaces.

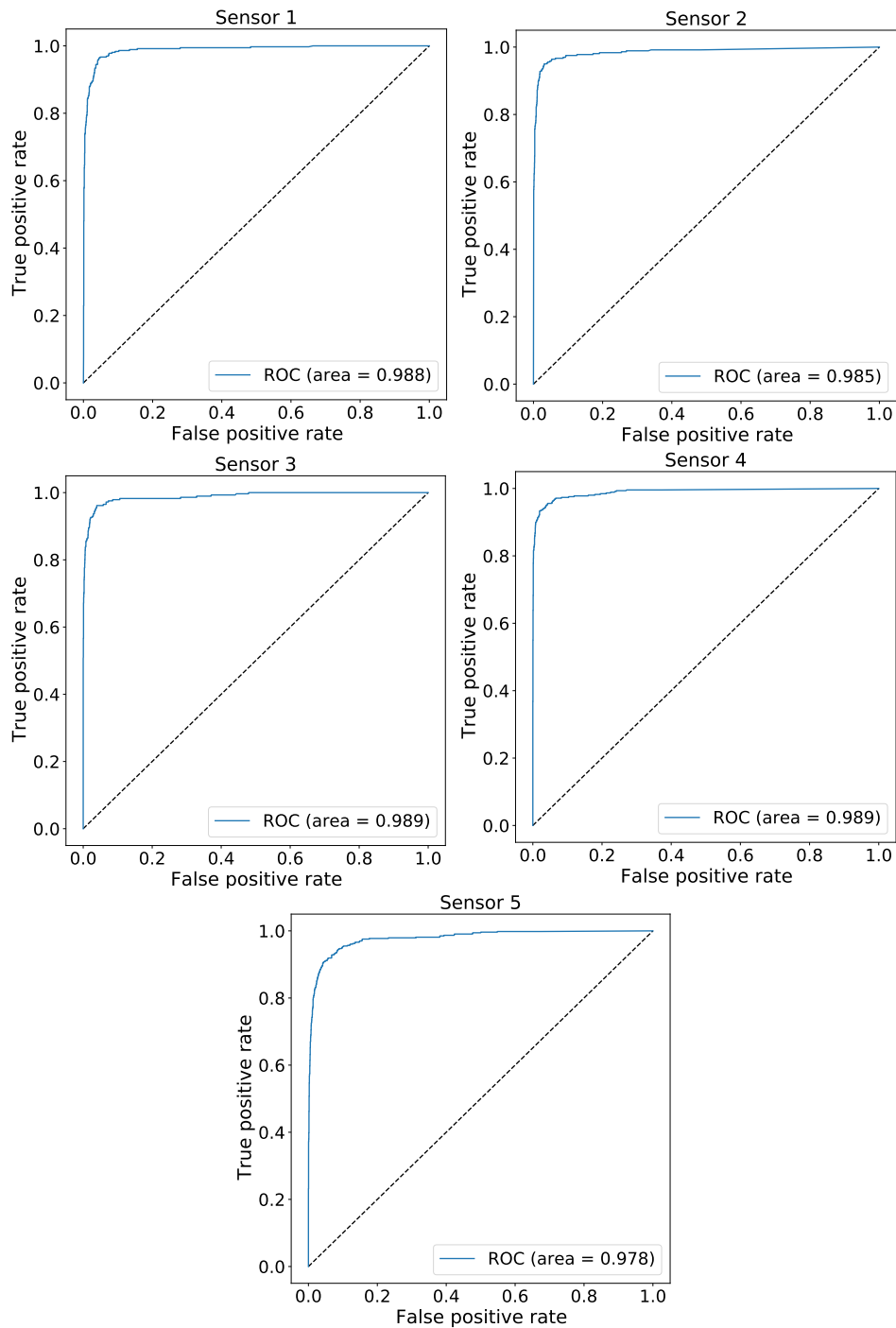


Figure 1.19. ROC curves obtained on the test set for Sensor 1, 2, 3, 4 and 5 respectively for Neural Network model

### Model performance comparison

The performance of the Random Forest and Neural Network models was compared using repeated stratified  $k$ -fold cross-validation with 10 folds and five repeats. This approach ensures that the class distribution in the validation sets is similar to the original dataset. The cross-validation procedure is repeated  $k$  times, with each fold serving as the validation set once. The results of the cross-validation, depicted in Figure 1.20, show the distribution of classification accuracy scores for both models.

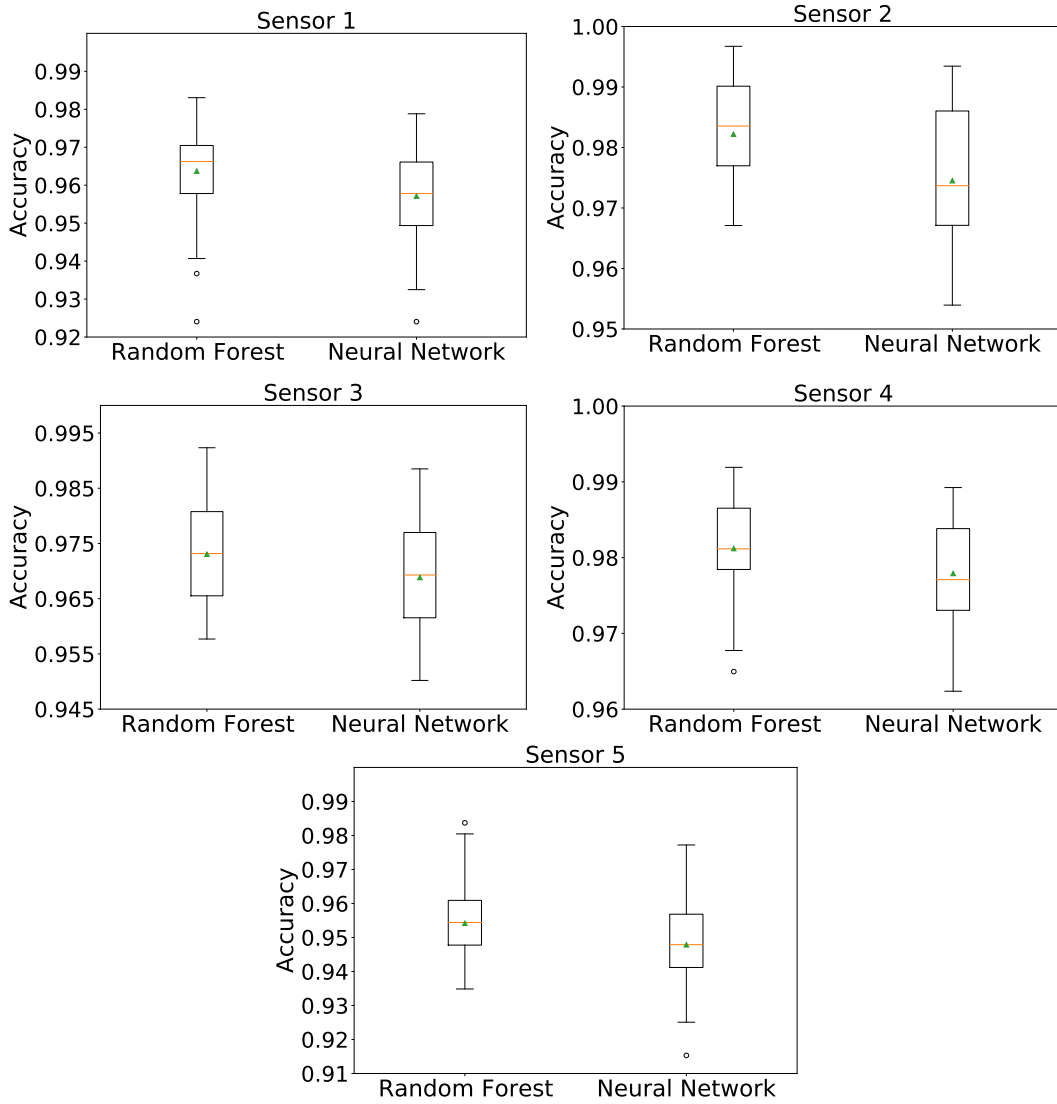


Figure 1.20. Distribution of classification accuracy scores for both Random Forest and Neural Network model from repeated stratified 10-fold cross-validation.

The results show that there is no significant difference in the accuracy of the presented models. To further confirm this observation, the Dietterich's 5x2-Fold Cross-Validation method (also known as 5x2-cv paired t-test) was conducted for statistical performance comparison. This test involves running five replications of 2-fold cross-validations. The original set is divided into two sets, namely  $X_1$  and  $X_2$ , where  $|X_1| : |X_2| = 50\% : 50\%$ . The RF model and the NN model are trained on  $X_1$  and onward tested on  $X_2$ . Let  $RF_{AX_1}$  and  $NN_{AX_1}$  be the accuracy in classification obtained for RF and NN model on the  $X_1$  set respectively. The performance difference measure is given by:

$$PM_{X_1} = RF_{AX_1} - NN_{AX_1}. \quad (1.7)$$

This process of computation if done once more but this time the  $X_2$  is used for training and

$X_1$  set for testing, obtaining the second difference performance measure

$$PM_{X_2} = RF_{Ax_2} - NN_{Ax_2}. \quad (1.8)$$

These calculations enable us to derive the mean and variance of differences as:

$$PM_{AVG} = \frac{PM_{X_1} + PM_{X_2}}{2}, \quad (1.9)$$

$$s^2 = (PM_{X_1} - PM_{AVG})^2 + (PM_{X_2} - PM_{AVG})^2. \quad (1.10)$$

Variance of differences is calculated for each of the 5 replications and utilized to derive the  $t$  statistic as follows:

$$t = \frac{PM_{x_1}}{\sqrt{\frac{1}{5} \sum_{i=1}^5 s_i^2}}. \quad (1.11)$$

Under the  $H_0$  hypotheses that there is no statistically significant difference between the RF model and NN model, the  $t$  statistics will approximately follow a  $t$  distribution with 5 degrees of freedom. Accepting  $H_0$  hypotheses, for a given level of significance, would show that the differences in the estimated performance metrics is a coincidence. Contrary, if  $H_0$  is discarded, it can be concluded that the differences in the performance metrics occurs because of the models do not have equal performance. A significance level  $\alpha = 0.05$  has been chosen enabling the computation of  $p$ -value using the  $t$ -statistic. If the  $p$ -value is smaller than  $\alpha$  the null hypothesis would be rejected. Thus, the calculated critical  $t$ -value is  $t_{5,0975} = 2.571$ , which is obtained using the `scipy.stats.t.ppf()` function in Python. If the absolute values of the  $t$ - statistics are greater than the critical  $t$  value, then the results of the test are statistically significant. Table 1.10 summarizes obtained results for the 5x2-cv paired  $t$ -test for all of the five sensors.

Table 1.10. Results of Dietterich's 5x2-Fold Cross-Validation statistical test for RF and NN model for all five sensors and significance level  $\alpha = 0.05$

Sensor	$p$ -value	$t$ -statistics value
Sensor 1	0.190	1.515
Sensor 2	0.464	0.793
Sensor 3	0.159	1.655
Sensor 4	0.264	1.259
Sensor 5	0.167	1.618

As can be observed from the results presented in the Table 1.10, it can be concluded that the RF and NN model have the same performance.

The importance of time-related variables in achieving accurate prediction or detection of parking status has been emphasized in prior research. This motivated the inclusion of these variables in the models presented in this study. However, it is important to consider whether

the observed parking patterns are driven by occupancy patterns or leakage of knowledge due to the use of LoRaWAN. To investigate this further, the feature importance rates for each sensor in the Random Forest model were examined. Therefore, a final comparison was conducted by removing the time variables from the data set, focusing solely on signal strength and signal-to-noise ratio. To ensure comparability, the same hyperparameters as previously used for the Random Forest and Neural Network models were considered in this analysis. Results of hyper-parameter tuning and detection accuracy of Random Forest model are presented in the Table 1.11 and Table 1.12 and for Neural Network model in Table 1.13. As can be observed from the tables, results remain consistent. The Random Forest model is not influenced by time variables with the exception for sensor 5. These results remain in accordance with the Data analyses and calculations of the feature importance. Furthermore, detection accuracy remain very high for the RF model. However, NN models performance has decreased especially for sensor 1 and sensor 5. This indicates that the information about time had an influence in accuracy detection for sensors 1 and 5.

Table 1.11. Result of grid search of Hyper-parameters for a particular sensor without time variables.

Sensor	n_estimators	criterion	max_depth	max_features	min_samples_split
Sensor 1	50	entropy	20	auto	2
Sensor 2	100	entropy	20	sqrt	2
Sensor 3	100	gini	20	auto	4
Sensor 4	50	gini	20	auto	2
Sensor 5	200	entropy	20	auto	2

Table 1.12. The results for Random Forest model for a particular sensor without time variables.

	TEST		
	Acc.	F-Score	AUC
Sensor 1	0.94	0.81	0.974
Sensor 2	0.97	0.87	0.994
Sensor 3	0.962	0.82	0.967
Sensor 4	0.974	0.89	0.974
Sensor 5	0.917	0.75	0.94

Table 1.13. Best results obtained for NN model without time variables.

	l. rate	TEST			
		Acc.	F-Score	AUC	
Adam	Sensor 1	0.0001	0.903	0.742	0.964
	Sensor 2	0.01	0.961	0.84	0.972
	Sensor 3	0.01	0.937	0.75	0.949
	Sensor 4	0.001	0.957	0.839	0.975
	Sensor 5	0.001	0.903	0.739	0.931

Overall, it can be noticed that Sensor 2 and Sensor 4 have gained slightly better overall accuracy for both Machine Learning models with and without the time variables. The reason

for this is due to the multipath propagation scenario which is more favourable for Sensor 2 and 4, than for others. When we compare the RSSI results of Sensor 5 in Figure 1.13 a) with Sensor 2 in Figure 1.13 c) be noticed that the difference between the most probable RSSI values for occupied and free parking states is higher in the case of Sensor 2 than in the case of Sensor 5. Additionally, the probability density curve is more distinct in Sensor 2 case, which may also lead to better accuracy.

### 1.2.3 The Empirical part of the research: Network Layer

In this part of the research, the feasibility of using Machine Learning algorithms for estimating frame size in RFID Gen2 systems using the ALOHA protocol at the MAC layer was considered and analyzed. To achieve this, Monte Carlo simulations were conducted to generate sets of tag distributions across slots for different frame sizes. The selected Machine Learning algorithms were compared with a state-of-the-art algorithm, specifically the Improved Linearized Combinatorial model (ILCM), for tag estimation. After comparing the performance of the algorithms and calculating the throughput for the given dataset, the analysis shifted to exploring the implementation possibilities of Machine Learning and deep learning algorithms on modern microcontrollers with limited resources. The aim was to maximize tag identification and throughput in such constrained environments.

In general, RFID presents radio technology that acts as a communication medium between a reader and the tag, with a unique identification used for communication [90]. In general, the RFID tag is distinguished by the presence or absence of the battery [91]. Passive tags are self-powered and communicate using the same RF waves emitted by the reader antennas, known as backscattering technology [92]. Among the existing technologies, passive Gen2 technology is considered the most attractive in IoT applications due to its simple design, flexibility, cost and performance [93, 94]. Gen2 as a standard is used on the physical and MAC levels to establish reliable communication between the reader and a tag. Readers must provide sufficient power to energize tags and respond to the necessary information since they are not equipped with batteries. The energy levels that tags can extract are quite low and therefore cannot afford energy-efficient MAC schemes [95]. In general, the MAC of RFID is random, and there are two widely used methods to achieve it: the first is a binary tree, and the second is the ALOHA-based protocol [91]. In binary tree protocol continuous YES/NO communication is achieved between reader and tags, while with ALOHA protocol tag initiates communication with a request from the reader [96, 97, 98].

One of the commonly utilized ALOHA-based protocols is the Dynamic Framed Slotted ALOHA (DFSA) since it has the most prominent performance which is the highest throughput. DFSA belongs to a group of time division multiple access (TDMA) protocols, where communication between a reader and tags is divided into time frames, which are, in turn, divided into time slots [99]. The beginning of the interrogation process in DFSA is induced

by the reader's announcement of the frame size [100]. This is done by the reader sending a QUERY command and the value of the main protocol parameter  $Q$  for the tags [101]. The value of  $Q$  is an integer ranging from 0 to 15 that sets the frame size at  $L = 2^Q$ . From there on, all tags that are being interrogated will occupy a randomly selected position in the frame (commonly known as a slot) and will onward reply back to the reader when their slot is being interrogated. Based on such an access control scheme three diverse scenarios may happen: a) only one tag is in the slot (the successful slot), no tags in the slot (empty slot), and c) numerous tags have taken the same spot (collision) [100]. The overall number of successful, empty, and collision slots is denoted with  $S$ ,  $E$  and  $C$  respectively. An example of an interrogating frame is exhibited in Figure 1.21. Therefore, the frame size is equal to the sum of

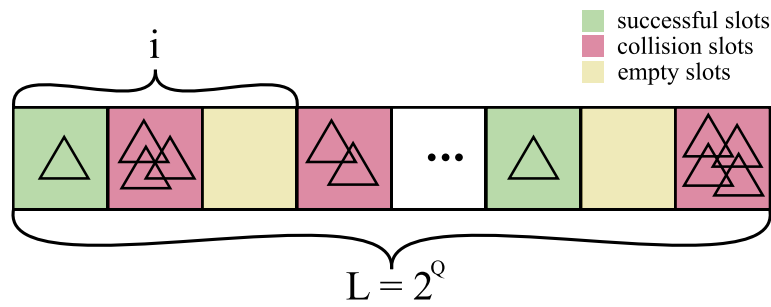


Figure 1.21. An example of an interrogating frame of a frame size  $L = 2^Q$ .  $i$  represents the size of a particular part of the frame.

successful, empty and collision slots, i. e.  $L = E + S + C$ . According to the previous notation, the throughput is defined using Equation (1.12) as :

$$\eta = \frac{S}{L}. \quad (1.12)$$

Therefore, the main goal in DFSA systems is to increase the number of successful slots  $S$  within the frame size  $L$ . As tags are fitting their slots randomly, previous studies [102] show that the maximum throughput will reach its maximum value of approximately 37% when the size of the frame equals number of tags. In usual situations, the number of tags is unknown, and has to be estimated in order to set adequate frame size, and consequently achieve maximum throughput.

In the context of tag estimation, commonly known scientific algorithms can be computationally demanding, primarily due to the calculations involving probabilities. This can be particularly challenging for standard microprocessors that are not designed to efficiently handle the computation of factorials. To solve the issue the Improved Linearized Combinatorial model (ILCM) for tag estimation has been introduced in [95]. The approach bypasses the conditional probability calculations by doing them in advance. Onward, the estimation model is uncomplicated and provides an effective tag estimation  $\hat{n}$  based on linear interpola-

tion given by Equation (1.13):

$$\hat{n} = kS + L, \quad (1.13)$$

where coefficients  $k$  and  $l$  are derived from Equation (1.14) and Equation (1.15) respectively:

$$k = \frac{C}{(4.344L - 16.28) + \left(\frac{L}{-2.282 - 0.273L}\right) \cdot C} + 0.2407 \cdot \ln(L + 42.56), \quad (1.14)$$

$$L = (1.2592 + 1.513L) \tan(1.234L^{-0.9907} \cdot C). \quad (1.15)$$

In the event of no collision the formula gives  $\hat{n} = S$ , whereas for cases when  $k < 0$ ,  $k$  must be set to 0. Following the tag estimation,  $Q$  value is calculated using Equation (1.16) as

$$Q = \text{round}(\log_2(\hat{n} - S)). \quad (1.16)$$

The results obtained by the researches have indicated that the ILCM shows a comparable behavior to state of art algorithms regarding the identification delay (slots), but is not computationally complex.

Aiming to improve the throughput of RFID systems, research presented in this study utilized Machine Learning classifiers as an approach for efficient tag number estimation. Performance of the ML algorithms is compared with state-of-the-art solution i.e. the ILCM model.

Tag number estimation can be regarded as a multi-class classification problem. Amongst many classifiers, Random Forest has been considered a simple yet powerful algorithm for classification, successfully applied in numerous problems such as image annotation, text classification, or medical data [103]. RF has been proven to be very accurate when dealing with large data sets, it is robust to noise and has a parallel architecture that makes it faster than other state-of-the-art classifiers [104]. Furthermore, it is also very efficient in stabilizing classification errors when dealing with unbalanced data sets [105]. On the other hand, Neural Networks offer great potential for multi-class classification due to their non-linear architecture and prominent approximation proficiency to comprehend tangled input-output relationships between data [106].

Discriminative models, such as Neural Networks and Random Forest can model the decision boundary between the classes [107], thus providing vigorous solutions for non-linear discrimination in high-dimensional spaces [108]. Therefore, their utilization for classification proposes has proven to be successful and efficient [109]. Both algorithms are able to model linear as well as complex nonlinear relationships, however, Neural Networks have a greater potential here due to their construction [110].

To obtain valuable data for model comparison, Monte Carlo simulations were done to produce an adequate number of possible scenarios that may happen during the interrogation



procedure in DFSA.

Generally, Monte Carlo methods are applied as algorithms for solving different computational problems by using random numbers (or rather, pseudo-random numbers), with applications ranging from materials science to biology to quantum physics [111]. Monte Carlo algorithms tend to be simple, flexible, and scalable and can be efficiently implemented on a computer [112].

Formally, the method can be mathematically described as follows.

Let  $f : [0, 1] \rightarrow [0, 1]$  be a continuous function. The integral  $\int_0^1 f(x)dx$  can numerically be calculated in the following manner. For a sequence of random independent variables  $(X_1, Y_1, X_2, Y_2, \dots)$  which are uniformly distributed on  $[0, 1]$ , let us define a new sequence  $(Z_n)$  as:

$$Z_n = 1, \text{ for } f(X_n) > Y_n, Z_n = 0, \text{ for } f(X_n) \leq Y_n, n \in \mathbb{N}. \quad (1.17)$$

$(Z_n)$  is therefore a sequence of independent equally distributed Bernoulli random variables. What is more

$$EZ_1 = P(f(X_1) > Y_1) = \int_0^1 \left( \int_0^{f(x)} dy \right) dx = \int_0^1 f(x)dx. \quad (1.18)$$

It can be concluded that  $\frac{1}{n}(Z_1 + Z_2 + \dots + Z_n) \rightarrow \int_0^1 f(x)dx$  almost everywhere. Therefore, for a numerical calculation of the integral  $\int_0^1 f(x)dx$ , one must generate random numbers  $(X_n, Y_n), n \in \mathbb{N}$ , from  $[0, 1]$  and the integral will approximately be equal to  $\frac{1}{n}(Z_1 + Z_2 + \dots + Z_n)$  for large  $n \in \mathbb{N}$ . This method of calculating the above integral by generating random numbers from  $[0, 1]$  is the Monte Carlo method. Basic idea of Monte Carlo simulations is to repeat the experiment many times (or use a sufficiently long simulation run) to obtain many quantities of interest using the Law of Large Numbers and other methods of statistical inference [112] and these simulations efficiently sample an equilibrium distribution.

The selected approach for Monte Carlo simulations of the distribution of tags in the slots follows the research done in studies [95] and [113]. Simulations were executed for frame sizes  $L = 4, 8, 16, 32, 128$  and  $256$  i. e. for  $Q = 2, 3, 4, 5, 7$  and  $8$ , where number of tags was in the range of  $[1, 2^{Q+2}]$  (this range was chosen based upon experimental findings elaborated in [113]). For each of the frame sizes and the number of tags, random 100 000 distributions of  $E$  empty slots, successful slots  $S$  and collision slots  $C$  were realized and are presented in Table 1.14.

Data obtained from the simulations were given to Neural Network, Random Forest and the ILCM model for adequate performance comparison and analyses of the accuracy of tag estimation.

Table 1.14. Snapshot of the obtained data

Q	L	S	C	E	N (number of tags)
2	4	2	1	1	6
2	4	0	3	1	15
...	...	...	...	...	...
8	256	79	122	55	401
8	256	18	229	9	943

The architecture of the NN model displayed in this research is constructed out of five layers as depicted in Figure 1.22.

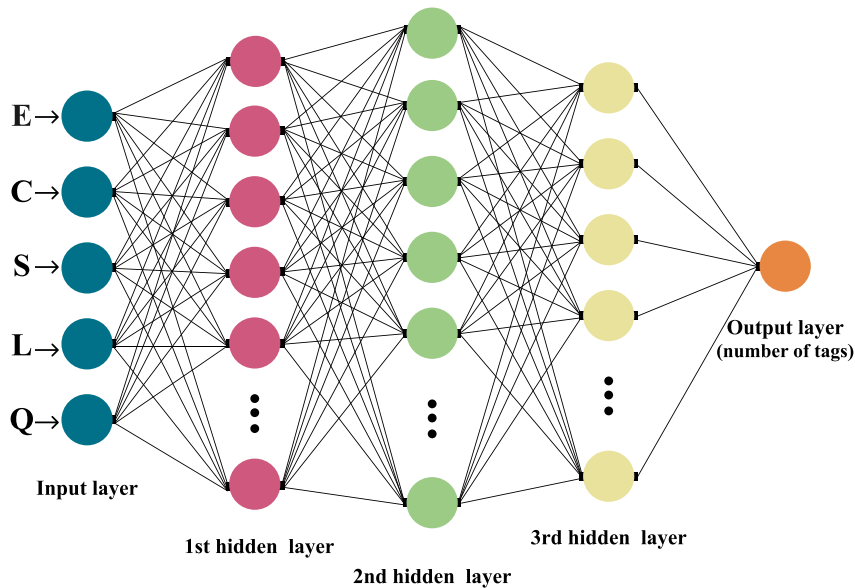


Figure 1.22. Architecture of the Neural Network model for tag estimation.

The first one is the input layer, followed by three hidden layers (one Dropout layer), and the final is the output layer. Applied activation functions were ReLU (in hidden layers) and Softmax (within the output layer). Data used for the input layer were number  $Q$ , frame size  $L$  and the number of  $S$  successful,  $E$  empty, and  $C$  collision slots. The number of tags that are associated with a particular distribution of slots within a frame is classified in the final, exit layer.

Data was further partitioned in 70% : 30% ratio, with 70% of data used for training and the other 30% for testing, with the target values being the number of tags, and all other values were provided as input. The training data was pre-processed and normalized, whereas target values were coded with One Hot Encoded with Keras library for better efficiency. By doing so, the integer values of number of tags are encoded as binary vectors. The dropout rate (probability of setting outputs from the hidden layer to zero) was specified to be 20%. The number of neurons varies based on the frame size, ranging from 64 to 1024 for the first four layers.

Since the classification of the number of tags is a multi-class classification problem, for this research, the Categorical Cross-Entropy Loss function was applied as the loss (cost) function with several optimizer combinations. Another important aspect of NN model architecture was thoroughly examined and that is the selection of optimizers and learning rates. Tested optimizers were RMSProp, SGD, and Adam. Adam provided the most accurate estimation results and was onward utilized in the learning process with 100 epochs and a 0.001 learning rate.

Aiming to produce the best classification accuracy for the Random Forest classifier, in this research hyper-parameters tuning has been done by utilizing *GridSearchCV* class from *scikit-learn* library with five-fold cross validation. Therefore, for every frame size hyper-parameters presented in Table 1.16 were tested resulting in a separate RF model for each of the frame sizes as presented in Table 1.15.

Table 1.15. Grid search results of RF Hyper-parameters for a particular frame size.

Frame size	n_estimators	criterion	max_depth	max_features	min_samples_split
L=4	50	gini	5	auto	2
L=8	50	gini	5	auto	2
L=16	100	gini	10	auto	4
L=32	100	entropy	20	sqrt	2
L=128	500	gini	20	sqrt	2
L=256	200	gini	20	sqrt	4

Table 1.16. Tested Hyper-parameters for Random Forest.

Hyper Parameter	Values
n_estimators	50, 100, 200, 500
criterion	gini, entropy
max_depth	3, 5, 10, 20
max_features	auto, sqrt
min_samples_split	2, 4, 6, 10

## Results and comparison

For ILCM, Neural Network, and Random Forest, the same data was used to make a comprehensive performance comparison. To provide a comprehensive classifier performance comparison several measures were taken into account. Firstly, to compare the performance of each classifier as a Machine Learning model, the accuracy measure was taken (since it is a standard metric for evaluation of a classifier), this being the categorical accuracy. Categorical accuracy is Keras built-in metric that calculates the result by finding the largest percentage from the prediction and then compares it to the actual result. If the largest percentage matches the index of 1, then the measured accuracy increases. If it does not match,

the accuracy goes down. Our experimental results point out that RF has out-performed the NN model in the classification task as shown in Table 1.17, but this measure alone is not enough to determine which of the two ML models would be preferable for utilization in the scenario of tags estimation. Therefore, given the nature of the tag estimation problem, have considered Mean Absolute Errors (MAE) and Absolute Errors (AE) as measures of classifiers performance (see eq.( 1.19) and eq.( 1.20) respectively). An accumulated estimation error will degrade the whole performance [114], meaning that the overall smaller MAE and AE for a classifier would determine the overall estimator efficiency, i. e. better system throughput. For approximated number of tags  $\hat{n}$  and exact number of tags  $n$ , MAE is defined as:

$$MAE = \frac{1}{m} \sum_{i=1}^m |\hat{n}^{(i)} - n^{(i)}|. \quad (1.19)$$

For every frame size AE was calculated as:

$$AE = |n - \hat{n}|. \quad (1.20)$$

Table 1.17. Classification accuracy of NN, RF and the ILCM model for a particular frame size.

Frame size	ACCURACY		
	NN	ILCM	RF
L=4	33.54%	23.55%	33.59%
L=8	28.56%	27.28%	28.22%
L=16	24.05%	23.27%	24.37%
L=32	19.78%	17.06%	19.54%
L=128	11.25%	4.42%	12.12%
L=256	5.74%	2.8%	9.46%

As can be noticed from Table 1.17, as frame size rises, the accuracy decreases for all of the three compared models. Furthermore, Random Forest seems to outperform other classifiers for the most challenging task for  $L = 256$ . Furthermore, ILCM performed similarly to NN and RF for smaller frame sizes.

On the other hand, the results presented in Table 1.18 indicate that the Neural Network model produces error rates comparable to RF although RF has better accuracy. What is more for the largest frame size NN will have an overall smaller MAE as can be seen from the Table 1.18 for frame sizes  $L = 128$  and  $L = 256$ . Overall, both Machine Learning classifier perform substantially better than the ILCM model.

This observation is further emphasised in the calculations of Absolute Errors of classification for RF, NN and the ILCM model. AE was derived for every frame size and histograms presented in the Figure 1.23 provide a pictorial comparison of the errors. As can be seen from Figure 1.23 a), for smaller frame sizes the NN model performs quite complementary to the RF model, but for the largest frame size, the NN (see Fig.1.23 c)) will have an overall smaller

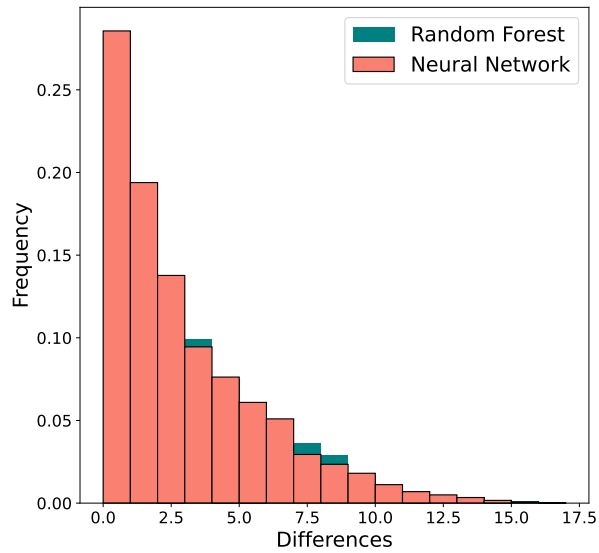
Table 1.18. MAE of NN, RF and the ILCM model for a particular frame size.

Frame size	MAE		
	NN	ILCM	RF
L=4	2.23	2.182	2.23
L=8	2.56	2.61	2.5
L=16	3.57	4.31	3.69
L=32	5.23	6.98	5.324
L=128	11.27	17.38	11.93
L=256	16.06	27.29	18.19

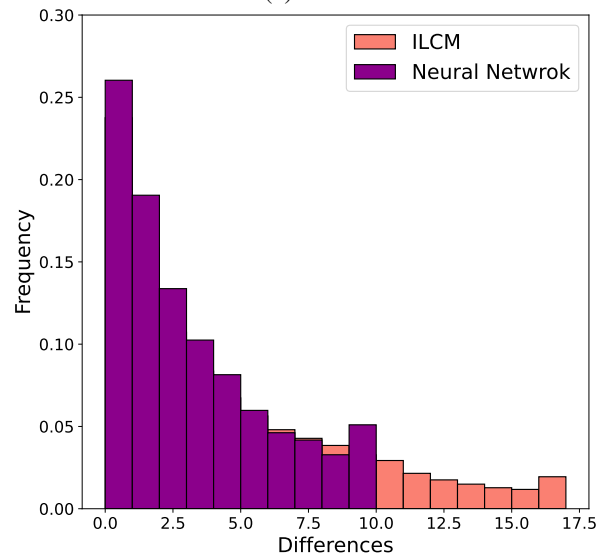
AE. These histograms are consistent with the MAE results from Table 1.18 confirming that the NN classifies values  $\hat{n}$  nearer to the true values of the number of tags  $n$ . This observation is important for estimating the length of the next frame, because the closer the estimated number of interrogating tags is to the actual number of tags, the better the frame size setting. Incorrect estimates of the total number of tags result in lower throughput. Results from this analysis show that, in comparison to the RF model, the NN model is generally "closer" to the real tag number. The overall goal is to reach maximum throughput and this cannot be done if the frame size adaptation is inadequate. The development of an effective and simple tag estimator is burdened by the variables that must be taken into account i.e. the frame size, the number of successful slots, the number of collisions or empty slots. Since major drawbacks of current estimators lie in their estimation capabilities, computational complexity, and memory demands. Therefore, to achieve a better setting of the next frame size, the focus of the estimation should be on the variable which contributes the most to the overall proficiency of the system [113].

Based on the obtained results, one final measure was done, i.e. comparison of throughput for the NN model, ILCM, and Optimal model. The Optimal being used as the benchmark is the one where the frame adaptation was set by the known number of tags. Results of the comparison are presented for the scenario of frame size  $L = 32$  realization and are exhibited in Figure 1.24. As can be observed from the Figure 1.24, the Neural Network model is close to the optimal one and outperforms the state-of-the-art ILCM model. This is particularly shown as the tag number increases, as can be seen in Figure 1.24 b).

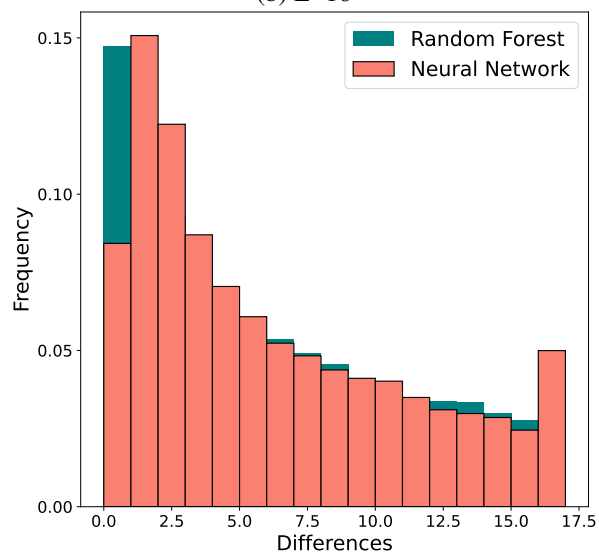
Based on the result of this examination of the performance of classifiers and comparison to the ILCM model, architectures of the Neural Network models were selected for further utilization.



(a)  $L=8$

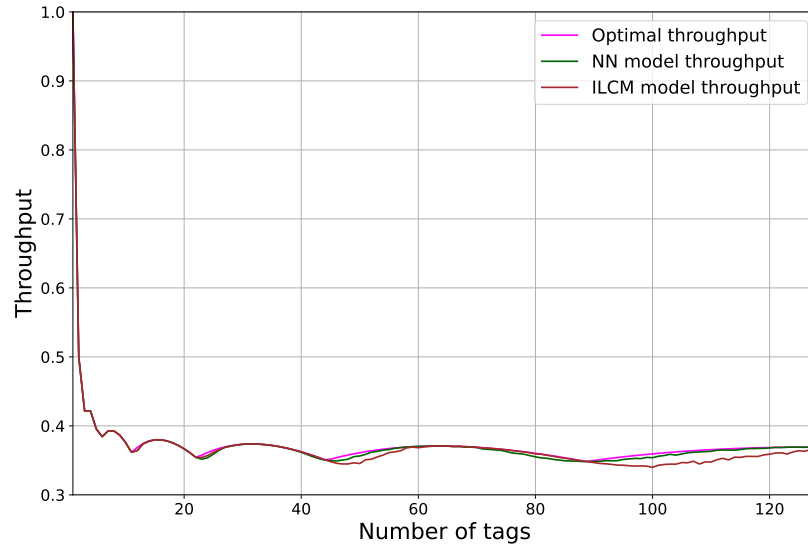


(b)  $L=16$

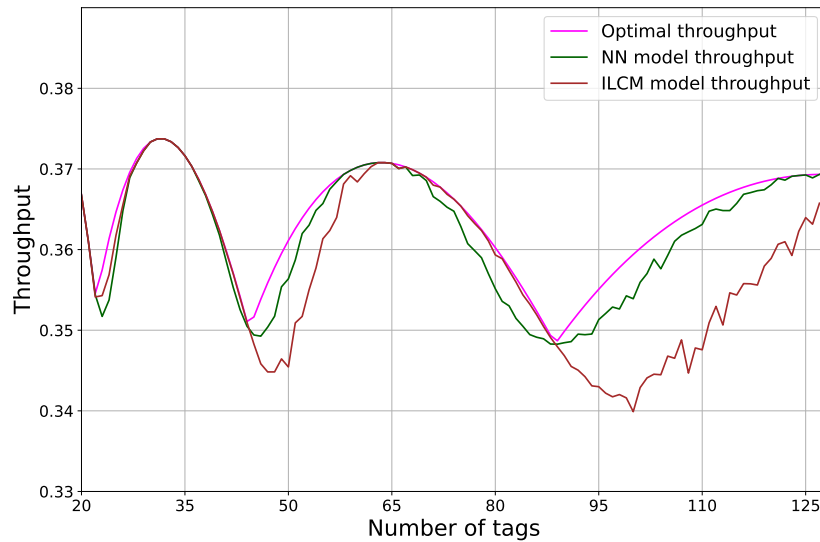


(c)  $L=256$

Figure 1.23. Comparison of absolute errors for Neural Network, Random Forest and ILCM model for frame sizes  $L = 8$ ,  $L = 16$  and  $L = 256$ .



a) Throughput for the NN model, ILCM and Optimal model



(b) Throughput for the NN model, ILCM and Optimal model for larger number of tags

Figure 1.24. Comparison of throughput for the NN model, ILCM and Optimal model for scenario of frame size  $L = 32$  realization.

### Mobile RFID Reader - Implementation Feasibility

Microcontroller boards are not optimized for performing complex floating-point calculations, unlike dedicated Personal Computers, as their main focus is on working seamlessly with peripheral components. However, by configuring the TensorFlow library, it is possible to use 32-bit floating-point data types for both data and weights, resulting in larger models. To address the limitations of microcontrollers, MCU-compatible models employ an approach that uses integer numbers (8-bit or 16-bit) instead of floating-point numbers, reducing the model size and significantly improving execution speed.

While the original model with 32-bit or 64-bit weights can be executed on a microcontroller, it may not fully leverage the capabilities of the TFLiteConverter. The TensorFlow Lite library for Microcontrollers enables the optimization of pre-trained Neural Network models for specific microcontrollers and their implementation on the devices. This optimization is achieved through smart quantization, which approximates 32-bit floating-point values to either 16-bit floating-point values or 8-bit integer values. Although there may be a slight loss in accuracy, this is compensated by reduced memory requirements and improved execution times, particularly in complex models.

Quantization plays a crucial role in determining whether a model can be run on a memory-restricted microcontroller. TFLiteConverter, part of the TFLite library, offers various optimization options. Float16 quantization reduces the size of the original model by half with minimal impact on accuracy. Dynamic range quantization uses 8-bit weights and floating-point activations, striking an optimal balance for certain low-performance yet capable computer boards. The third optimization option, ideal for low-power microcontroller devices, employs full integer quantization, where both weights and activations are 8-bit, and all operations are performed using integers. This quantization approach is slightly more complex, as the converter requires a representative dataset for the quantization process of the entire model.

Data shown in Table 1.19 provides simple insight of accuracy decrease due to performed quantization for models used in our paper.

*Table 1.19. Model accuracy before and after quantisation*

	<b>Original model</b>	<b>Quantised model</b>
Model L=4	33.33 %	32.72 %
Model L=8	28.53 %	27.58 %
Model L=16	23.11 %	22.04 %
Model L=32	19.00 %	12.08 %
Model L=128	8.03 %	4.08 %
Model L=256	6.71 %	3.03 %

It can be observed that a decrease in accuracy is observable for the two most complex NN architectures (L=128 and L=256), while for the least complex NN architectures (L=4 and L=8) loss due to quantization is merely measurable. Loss inaccuracy for the two most complex architectures is possibly the result of output quantization where more than 256 classes are possible (notably 512 for L=128 and 1024 for L=256).

After the quantized model is created, a file containing the model which the microcontroller will understand is created. Linux command tool `xxd` takes a data file and outputs a text-based hex dump, which we copy-paste as a c array, and add to a microcontroller project source code (as an additional header file).

Among the microcontroller boards tested, the Teensy 4.0 stands out as the fastest avail-



Table 1.20. Model performance on Teensy 4.0 MCU, Arduino DUE and Raspberry PI4

Frame size	Model size [Bytes]	Execution time [microseconds]		
		Teensy 4.0	Arduino DUE	Raspberry PI4
L=4	4320	22	897	143
L=8	5152	32	1284	159
L=16	6592	48	1983	173
L=32	13824	120	4928	187
L=128	75776	692	29615	270
L=256	283264	1669	111374	648

able on the market. It features an ARM Cortex-M7 processor with an NXP iMXRT1062 chip clocked up to 600 MHz, offering excellent performance for complex calculations. The Teensy 4.0 consumes approximately 100 mA of current at a 3.3V supplied voltage, making it more power-efficient than desktop computers or other microcontroller boards. With 1024K RAM available, it provides sufficient storage for ML models, which can be stored in FLASH memory and read into RAM as needed.

Another considered microcontroller board was the Arduino Due, based on the AMR M3 architecture. It features an Atmel SAM3X8E microcontroller clocked at 84 MHz, but has a smaller RAM capacity of 96 KB. This limited RAM may restrict its usability for executing complex ML models, as reading data from slower FLASH memory can result in longer execution times.

The third board considered was the STM32F103C8T6, also known as the "blue pill," which is based on the Arm Cortex-M3 microcontroller. This board operates at 72 MHz and has only 20 KB of RAM and 64 KB of FLASH memory, making it unsuitable for holding and executing most of the NN models tested.

Figure 1.25 presents an overview of the three tested microcontroller devices.

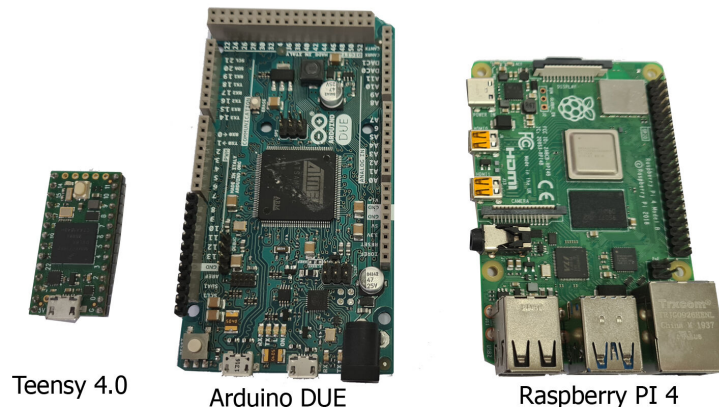


Figure 1.25. "Devices used in the test: Teensy 4.0 (left), Arduino DUE (center), Raspberry PI4 (right), source: Own photo

For a comprehensive assessment of the microcontroller's performance in executing the proposed NN models, the same quantized TensorFlow models were also tested on a Rasp-

berry Pi 4B computer board. The Raspberry Pi 4B features a quad-core Cortex-A72 1.5GHz SOC with 4GB of RAM and runs the Raspbian desktop OS with kernel version 5.10.

To ensure simplicity and ease of development, the coding for the microcontroller side was carried out using the Arduino IDE. This IDE provides a user-friendly interface and access to various additional libraries for extending the project's functionality. The ML model was incorporated into the project by including the hexdump file as an additional header file, which was then converted into a binary format and transferred to the microcontroller's FLASH memory during programming.

On the Raspberry Pi computer, a straightforward Python script was utilized along with the TFLite interpreter library to execute the ML model. This approach allowed for seamless integration and execution of the model on the Raspberry Pi.

Several proposed model architectures on the Teensy 4.0 microcontroller board, were trained, that have been considered as the optimal solution for executing proposed NN models. The presented analysis aims to indicate the real limits of the NN architecture that can be fluently run on selected hardware. ANN layers configuration was kept intact, while the complexity of the model was achieved by increasing the number of neurons in the third and fifth layers. By utilizing a microcontroller integrated timer, the average ANN execution time has been measured on the microcontroller. Another interesting piece of information obtained was the quantized model size and amount of RAM commonly assigned for storing global variables after initial programming. Please note that microcontrollers usually do not possess the possibility of measuring free RAM space during execution, as compared to computers. The used library offers some tweaking of tensor size, which may reduce or increase available RAM size and consequently affect execution time, but we kept this option on default for all tested models and all devices. It is recommended to keep at least 10% of available RAM for local variables for stable performance. Results for all 6 models' execution times and model size on Teensy 4.0 ARM Cortex M7 microcontroller, Arduino DUE ARM Cortex M3 microcontroller, and Raspberry PI4 computer board are listed in Table 1.20.

As can be observed from Table 1.20 increasing the number of neurons in hidden layers (notably 3. and 5. hidden layer) and in output layers increases the model size and prolongs execution. As an example, comparing models for  $L=4$  and with the model for  $L=16$  which have exactly twice more neurons in layers 3 and 5, the total model size increases by a factor of 1.5, while execution time on the Teensy 4.0 microcontroller observes an increase by factor 2.2. The last presented model ( $L=256$ ) features an increase in model size by a factor of 65 with execution time with a factor of 75 as compared to the simplest model ( $L=4$ ). It is worth mentioning, that the last model represents an example of the most complex ANN model that our microcontroller can hold, where after importing it to the microcontroller only 13% of RAM was free for local variables. We also observed that increasing the depth and/or increasing the number of neurons per layer of an NN pose a significant memory demand, which can be afforded only by high-end edge devices (eg. RaspberryPI). The average exe-

cutation time for the exemplary most complex model was 1.7 ms, which is surprisingly fast for this type of device and can offer real-time performance. ARM Cortex M3-based Arduino DUE behaved similarly to the Teensy 4.0 microcontroller with significantly larger execution times (41 times slower on the simplest model and 67 times slower on the most complex model). The execution of the most complex model took 111 ms, which makes it impractical for some real-time scenarios. Execution times on Raspberry PI4 computer varied greatly (due to non-real-time OS architecture) and surprisingly showed to be much slower for less complex models (up to  $L=32$ ). For more complex models Raspberry PI was able to benefit from its enormous computing power, and the most complex model executed in 0.6 ms, which is when compared to Teensy 4.0 not significantly better to persuade us to use computer boards instead of the microcontroller. This once more proves that if the loss in accuracy due to quantization is acceptable, the only real limitation is available RAM and FLASH memory on the used microcontroller.

In some scenarios RF can offer better or comparable results than deep NN with only fraction of execution time required on MCU [115]. As aim of our study was to increase throughput, which is achieved by better estimating number of interrogating tags, that which is best performed by NN model, thus only NN model was considered for implementation on microcontroller. Additionally, NN models can offer numerous optimization and quantization possibilities, which is worth further investigation.

Based on the overall result, one final observation is made. As can be noticed in Figure 1.24,  $\eta$  for ILCM and NN is quite different. Such diversity is a result of the ILCM's interpolation, even though it contributes to lower computation complexity. When examining the worst case for both models, i. e. frame size  $L = 256$ , the Neural Network model reaches  $\eta_{NN} = 0.2498$  in contrast to  $\eta_{ILCM} = 0.2265$ . This results in a difference of 0.0233, which is approximately 6 successful slots per given frame. Reader setting determines the execution time per frame and such a time cost needs to be compared with the time for a successful tag read, i.e. successful slot time. Based on empirical evidence from research studies as ones in [116], the time for standard reader setting in a general scenario is 3ms. Therefore, the read tags that are scarified as Neural Network computational burden are equal to  $1.7ms/3ms = 0.57$ .

#### 1.2.4 The Empirical part of the research: Application Layer

This particular aspect of the research was conducted at the Application Level of the IoT stack. The main research focus was on design and evaluation the first prototype of a Smart learning toy for preschool geometry education, utilizing IoT technology, particular sensing technology, Machine Learning algorithms, and user-centered design principles. Furthermore, a preliminary pilot testing study was done, aiming to assess the usability and performance of the toy prototype while also exploring how the toy can enhance young children's learning

experiences in a fun and engaging manner.

In the pursuit of designing a Smart toy for geometry learning, the integration of IoT sensing technology and Machine Learning algorithms was chosen for incorporation into an existing plush giraffe toy. This strategic decision aimed to leverage the toy's pliability, familiarity, and adaptability to create an interactive and captivating learning experience for young children. The inherent flexibility of plush toys facilitated the seamless integration of sensors and electronic components, preserving the overall aesthetic and tactile appeal of the toy. Plush toys have established themselves as child-friendly and comforting, making them an ideal platform for designing engaging educational tools. Additionally, these toys have demonstrated their ability to foster emotional connections with children, enhancing personalization and enjoyment [117, 118]. Furthermore, the gender-neutral nature of animal-themed and robotic toys presented an opportunity to explore potential gender-based preferences among children [119].

## Hardware

The main hardware components of the smart toy are presented in Figure 1.26. The specific functions of the components are elaborated in the rest of this section.

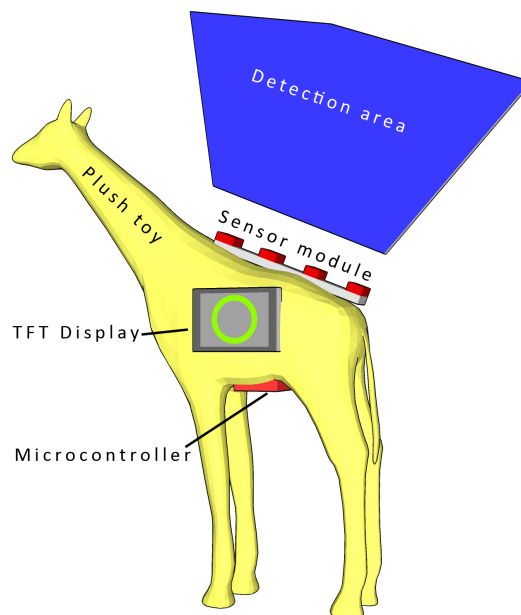


Figure 1.26. Main Smart toy hardware components

The interaction with the Smart toy follows these steps: a geometric shape is displayed on an LCD screen with an accompanying sound signal. The child is then prompted to draw the shape above the Sensor module using hand movements. The Machine Learning algorithm integrated into the toy analyzes the drawn gesture and determines if it matches the displayed shape, providing immediate feedback. This approach minimizes cognitive effort and enables

seamless interaction with the system [120]. Hand gestures enhance usability, especially for young children, and contribute to the development of fine motor skills. Fine motor skills are crucial for early childhood development and are linked to improved learning abilities and overall cognitive development [121]. Gesture studies highlight the role of kinetic movement in the origin and development of abstract geometric cognition in children [122, 123, 124]. The toy's audio and visual feedback enriches the learning experience, making it engaging and enjoyable. Additionally, this activity fosters the development of spatial skills, which are essential for success in STEM fields like mathematics and science [125].

The combination of the Teensy 3.6 microcontroller board, SdFat library, SanDisk MicroSD card, piezoelectric speaker, Newhaven TFT display, and GPIO-connected pushbutton constituted the hardware foundation of the system, enabling data acquisition, user feedback, and interaction. The proposed system incorporates a microcontroller with additional modules to fulfill its functionality requirements. For data acquisition and storage, the Teensy 3.6 microcontroller board was chosen due to its ARM Cortex-M4 MK66FX1M0VMD18 core, 1024 KB Flash, and 256K RAM, operating at 180 MHz. The Teensy 3.6 board met the necessary criteria for real-time acquisition and logging data to a MicroSD card. The SdFat library, compatible with Teensy microcontrollers, facilitated fast write, read, and file handling operations for data logging. A SanDisk Class 10 MicroSD card was utilized for this purpose, although any Class 10 microSD card would suffice. To provide user feedback, the system incorporated both audio and visual components. A piezoelectric speaker (buzzer) was employed to produce limited and short monophonic melodies, signaling events such as measurement start, end, or error states. Visual feedback was achieved through a Newhaven 4.3-inch TFT display with an integrated FTDI FT800 TFT Controller. This display offered a resolution of 480 x 272 pixels, capable of displaying up to 262K colors. Communication between the display and the microcontroller utilized SPI, with a clock rate of up to 30 MHz. The Newhaven library facilitated easy integration and the creation of simple geometrical objects and progress bar elements for display. A pushbutton, connected to an interrupt-enabled GPIO pin via a long cable, served as a trigger for initiating measurements. During the development and testing stage of the system, the microcontroller board was directly connected to the PC using a 480 Mbit/sec USB 2.0 interface. This configuration allowed insight into all raw sensor data, more flexibility when testing different ML models, and deeper information on the performance of each ML classification algorithm tested.

A schematic of all electronic components and interfaces between devices is presented in Figure 1.27.

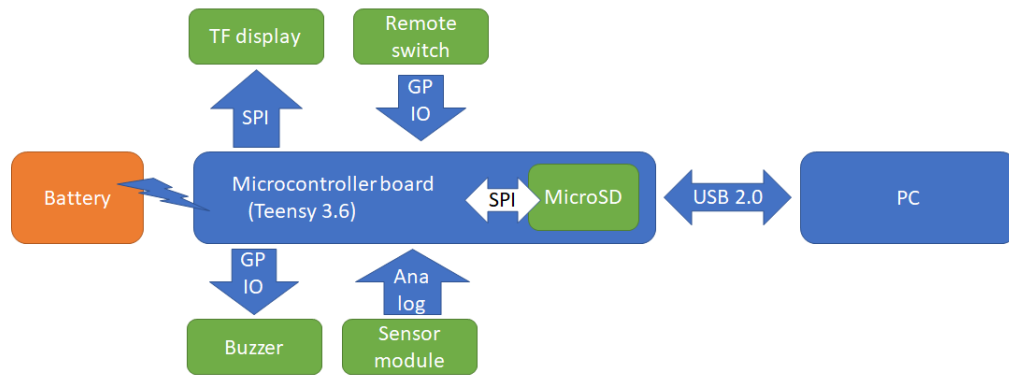


Figure 1.27. Electronic components and interfaces between devices

## Sensing Technology

Regarding sensing technologies employed, to locate a hand in space, several possibilities were considered, including visual recognition, capacitive sensor, ultrasonic sensors, TOF sensors, and finally selected infrared sensors. The proposed system is based on a microcontroller rather than a single-board computer like Raspberry Pi, primarily due to power requirements and faster boot-up times. Using an RGB or RGBD camera for real-time sensing would necessitate a powerful embedded computer for data processing [126, 127]. As an alternative, proximity sensors were chosen for their affordability, low power consumption, and simple 1D output and visualized in Figure 1.28.

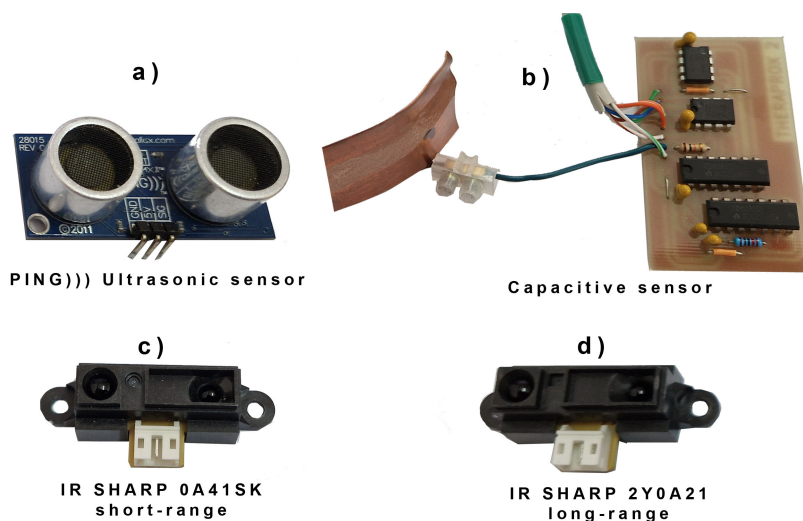


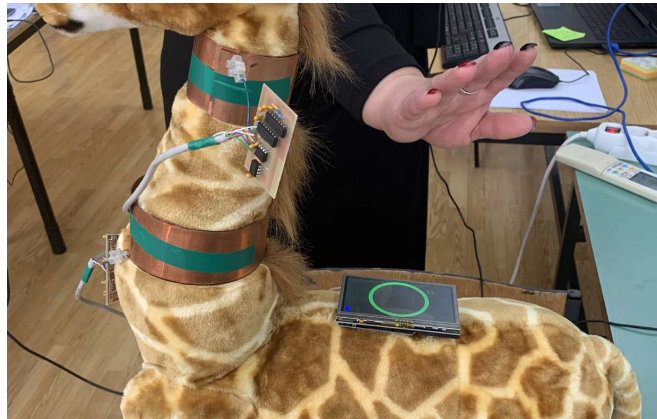
Figure 1.28. Distance sensors evaluated during the development of the proposed device (a) Parallax Ping))), (b) In house developed capacitive sensor, (c) short-range Sharp IR sensor, (d) long-range Sharp IR sensor

Initially, the HC-SR04 ultrasonic distance sensors (Parallax Ping))) were considered for their conical sensing area and ability to measure distances in large volumes (<https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf>). However, when

multiple sensors were used in overlapping areas, interference caused unsatisfactory performance and a useful acquisition rate below 10Hz.

Another sensor type, the TOF VL53L0X distance sensors, excelled in precise distance measurement and refresh rate due to their principle of operation and small sensing area (<https://www.st.com/en/imaging-and-photonics-solutions/vl53l0x.html>). However, the system would require a dense array of sensors to reliably detect and track hand movement, increasing the cost and complexity of the system.

For the initial prototype, an in-house developed capacitive proximity sensor with a 10 cm sensing range was selected [128]. To enable gesture recognition in a two-dimensional plane, two capacitive sensors were mounted on the neck of a plush toy, creating a virtual canvas for users to perform gestures on as presented in Figure 1.29 sensor acquisition rate to a. The sensing element of the capacitive sensor, made of copper sheet, was determined experimentally to provide the optimal sensing range without introducing excessive ambient capacitance. The calibration procedure equalizes the frequencies of the sensing and referent oscillators when no objects are present within the sensing range. This ensures a maximum output voltage from the sensor, which reduces proportionally as objects are brought closer [128].



*Figure 1.29. Researcher interacting with the first prototype of the device, featuring capacitive proximity sensors*

In addition to capacitive sensors, Another type of sensor that was considered was a family of Sharp infrared distance sensors ([https://global.sharp/products/device/lineup/data/pdf/datasheet/gp2y0a21yk\\_e.pdf](https://global.sharp/products/device/lineup/data/pdf/datasheet/gp2y0a21yk_e.pdf)). There are a few similar models that are completely compatible and the only difference is the minimum and maximum measurement distance. We have tested two models, GP2Y0A41SK0F (Figure 1.28 c) which operates in the range of 3 to 40 cm, and GP2Y0A21YK0F (Figure 1.28 d) which operates in the range of 10 to 80 cm. Both sensors are analog, which means that they output a signal with roughly a 2.15V differential between the minimum and maximum distance, which is read using the microcontroller's integrated AD converter. Both short-range (30 cm) and long-range (80 cm) IR sensors were implemented and a dataset for ML training was built. Based on real-world test-

ing and comparisons, the 80 cm version of the IR sensor was selected as the optimal choice, despite its longer minimal distance, to track gestures performed in larger volumes. These sensor choices, including the capacitive proximity sensor and the Sharp GP2Y0A21YK0F Analog Distance Sensor, provide analog signals that can be read using the microcontroller's integrated AD converter. However, the relationship between the measured distance and the analog signal is non-linear, requiring recalculation for accurate distance determination.

### **Data Collection, Processing, and Machine Learning**

Building accurate and robust models for complex hand gesture recognition is challenging due to the diversity and complexity of hand gestures. Therefore, preliminary testing of Machine Learning models with collected data is critical to ensuring their reliability and effectiveness. Data have been collected from 8 adult individuals to serve as data for building a Machine Learning model. The research employed a non-probability sampling method known as convenience sampling, which entails selecting study participants who are easily accessible and willing to participate. In this case, those were academic staff involved in the research project on a wider scope. All subjects signed an informed consent form in accordance with the Declaration of Helsinki and approved by the Ethics Committee of the Faculty of Electrical Engineering, Mechanical Engineering, and Naval Architecture. Each individual has performed gesture movements for around an hour. In general, around 200 gestures (depending on the sensing technology) were gathered per individual, and onward were processed, depending on the sensing technology. While preparing the training dataset, subjects were instructed to follow all five hand gestures (shapes) when one is displayed on the screen. Those were, namely, circle, square, triangle, rhombus and pentagon. ML models were originally trained in all shapes, but the performance of the test data set showed to be unsatisfactory. However, by keeping only three gestures (namely circle, square, and pentagon), we have achieved better categorical accuracy.

An example of the raw sensor reading is presented in the Figure 1.30 for capacitive sensor, Figure 1.31 for the short-range IR sensor, and Figure 1.32 for the long-range IR sensor. All three figures show raw data for the same performed gesture. As the system is set to capture raw data in 10 s intervals, useful data (actual hand gesture) is found only in a few seconds intervals and can be anywhere inside the original signal. While comparing raw data reading for the same gesture, some similarities can be found across all sensor readings (please note that capacitive sensor is using only 2 sensors and the signal is inverted).



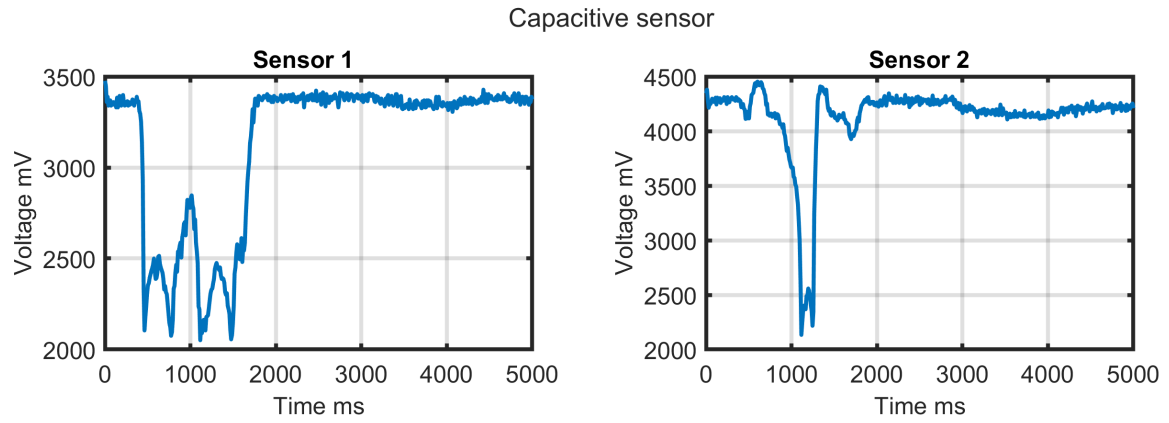


Figure 1.30. Raw sensor readings, 2 capacitive sensors

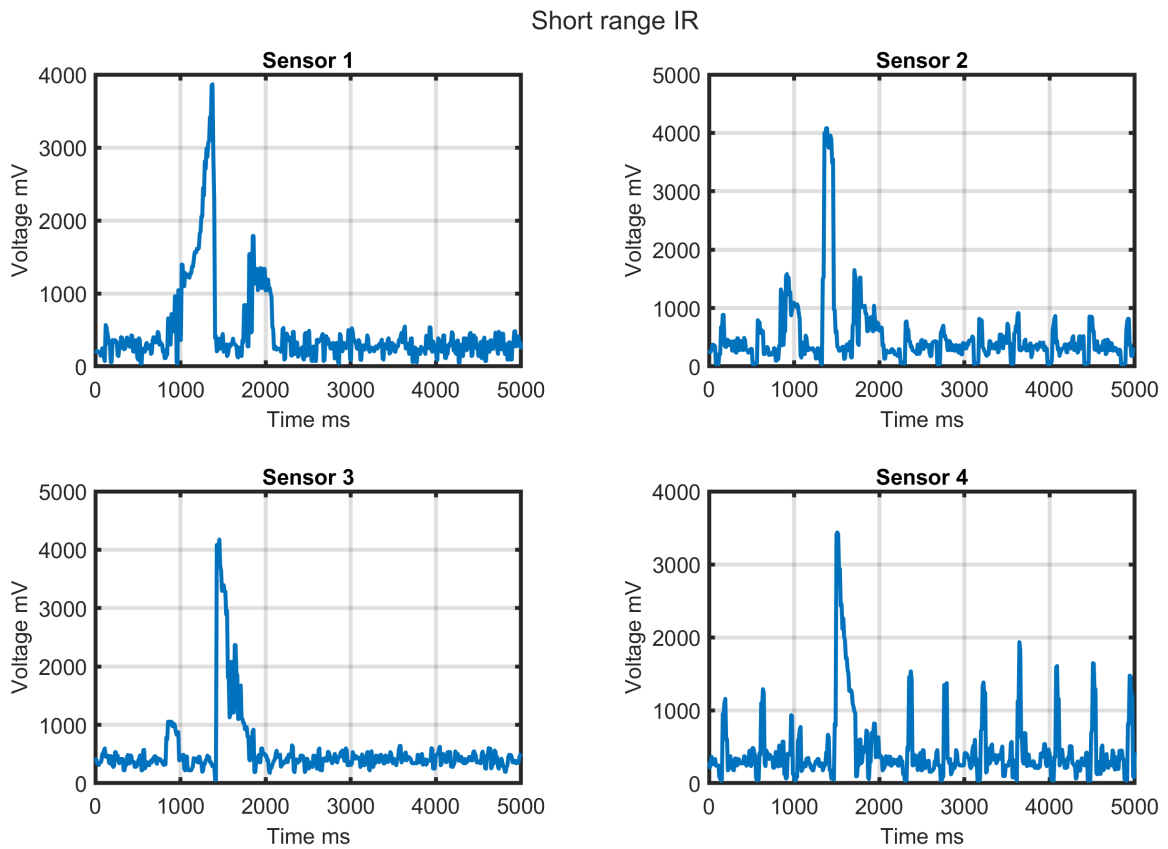


Figure 1.31. Raw sensor readings, 4 short-range IR sensors

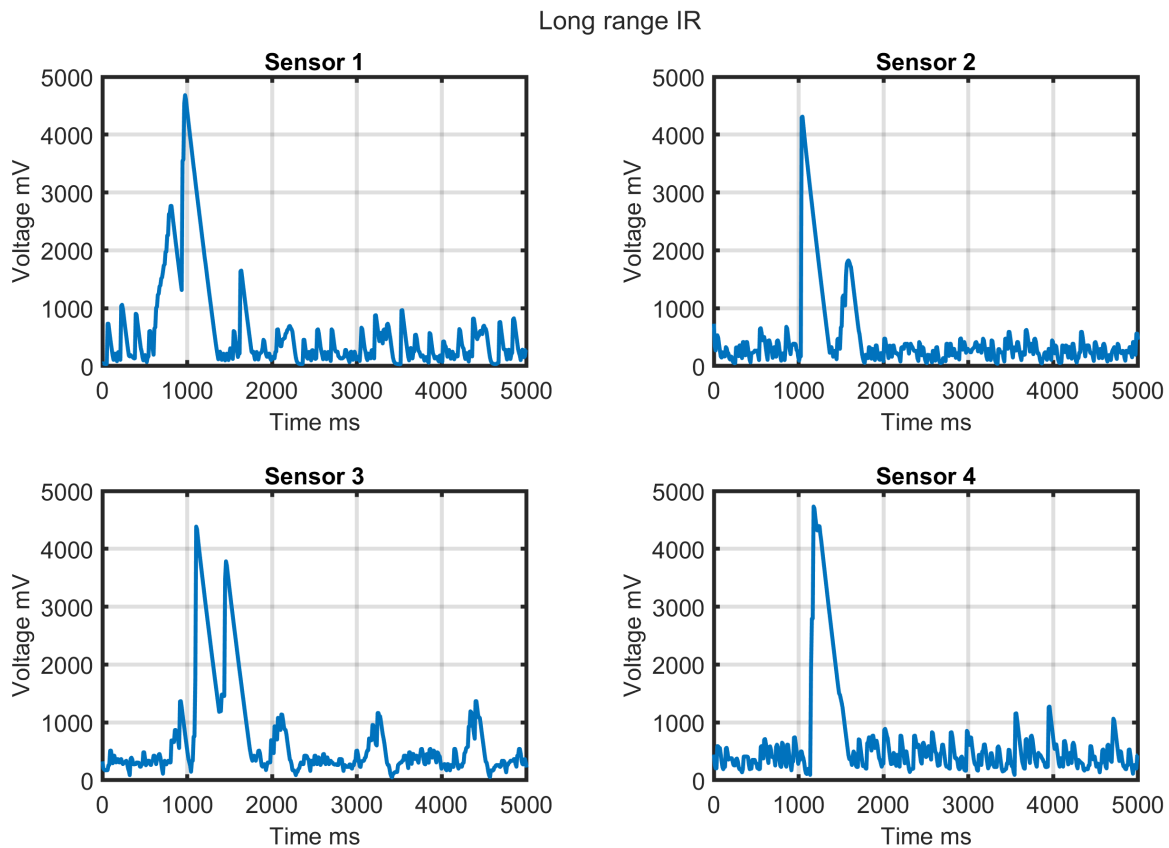


Figure 1.32. Raw sensor readings, 4 long-range sensor

In the context of Smart toys and human-computer interaction, the utilization of SVM, Random Forest, and Neural Networks provides valuable options for effective gesture recognition and interaction analysis. These algorithms have been successfully applied in diverse domains, such as educational technologies, gaming, and robotics. Their ability to handle complex and multi-modal input data, along with their robustness and scalability, makes them well-suited for the development of intelligent systems capable of accurately and real-time interpretation and response to children’s gestures [129, 130, 131].

In order to maximize the performance of the SVM, RF, and NN models for gesture recognition, a parameter grid search was conducted [132, 133, 134]. The grid search approach is employed to systematically explore various combinations of hyperparameters, allowing for the identification of the optimal configuration [132]. The selected parameters for each model are presented in their respective tables below (Table 1.21, Table 1.22 and Table 1.23).

Table 1.21. SVM Parameter Grid Search

Kernel	C	Gamma
Linear	0.1	Scale
RBF	1	Auto
Poly	10	Scale

Table 1.22. Random Forest Parameter Grid Search

Max Samples	N Estimators	Criterion	Max Depth	Max Features	Min Samples Split
0.1	50	Gini	3	Auto	2
0.2	100	Entropy	5	Sqrt	4
0.3	150	Gini	7	Auto	6
0.4	200	Entropy	9	Sqrt	10

Table 1.23. Neural Network Parameter Grid Search

Hidden Layer Sizes	Activation	Solver	Alpha	Learning Rate
(50,)	ReLU	SGD	0.0001	Constant
(100,)	Tanh	Adam	0.001	Adaptive
(50, 50)	ReLU	SGD	0.01	Constant
(1024, 512, 64, 6)	ReLU	Adam	0.001	Adaptive

The Table 1.24 presents the initial results of Machine Learning models applied to raw data from different sensor types for gesture recognition. The accuracy of the models varied across sensor types. The IR long-range sensor demonstrated higher accuracy in gesture recognition compared to other sensor types due to its ability to capture more detailed and precise data. The extended range of the sensor allows for capturing a wider range of hand movements, resulting in improved classification accuracy. The longer range also helps in reducing occlusion and interference, leading to more reliable and consistent recognition of gestures.

Table 1.24. ML results for raw data

Table Sensor type	Accuracy		
	<i>SVM</i>	<i>RF</i>	<i>NN</i>
Capacitive	0.80	0.78	0.80
IR short-range	0.82	0.70	0.82
IR long-range	0.91	0.89	0.91

Raw sensor data were pre-processed before to improve the accuracy and efficiency of Machine Learning models. Due to the nature of the sensors used, the raw data is noisy and inconsistent, making it difficult to extract meaningful information. Data preprocessing helps to address these issues and prepares data for analysis through data transformation, data cleaning, and data reduction. Data transformation converts data into a more suitable format by doing linear or non-linear scaling and normalization of numerical values. As a notable example, the IR distance sensor outputs non-linear analog data that could be transformed to a linear distance [135].

By performing non-linear scaling before feeding data to an ML model, the first layers of our Machine Learning model do not have to find relations between non-linear voltage input and actual linear distance and can focus on resolving hand gesture form transformed linear

distance data. The data cleaning technique removes or corrects errors and inconsistencies and predicts missing values. This requirement is again presented on the IR distance sensor, which internally updates readings with a 25 Hz refresh rate, while our system is set to a fixed 50 Hz refresh rate. The faster refresh rate was required as the IR distance sensor outputs faulty readings during short periods of internal distance recalculation, and there is the possibility of reading the sensor output during that exact moment. By having more readings than required, simple data filtering can be performed, and outliers are simply removed and replaced with mean neighboring values (using a mean filter). The data reduction technique effectively reduces the size of the dataset while still preserving important information. As reported in the literature, human self-paced movements are within the 3.3 Hz bandwidth (ref), thus the system's 50 Hz sampling rate is excessive for recognizing complex hand gestures. Additionally, the training and inference time of any ML model is significantly reduced by reducing the input size. By our conservative estimation and general experience, a 10 Hz refresh rate was selected as optimal that balances the performance of the ML model and the complexity of the ML model. Data reduction was performed by resampling 500 inputs per sensor (for 10 s measurement time) to 100 inputs using cubic spline interpolation. When data are resampled at a five-fold lower rate, noisy sensor inputs are filtered, and readings are smoothed. By resampling data to a 1:5 rate we have effectively achieved low-pass filtering and simplification (reduction) of the ML model. With this approach, we are effectively reducing the 50Hz sensor acquisition rate to a 20Hz acquisition rate, which is still suitable to recognize complex hand gestures. If a lower acquisition rate were to be used, some faster movements may be tracked with an inadequate number of samples thus preventing accurate recognition. Additionally, when the original input vector (4 x 500 samples) is used for ML training with a similar ML model (only input size was modified) categorical accuracy on the test is significantly reduced to 0.86, and with the model size around 6.3 MB (1.4 MB for resampled inputs) which may be inadequate for ML implementation on microcontrollers.

An additional pre-processing step was also considered, where only data belonging to the performed gesture are extracted and forwarded to an ML model. This is usually done by observing the first and the last samples where the object is detected by sensors and extracting all samples in between. This approach was shown to be unreliable in practice, as the subject may place the hand in the sensed area long before or keep it long after the required gesture is performed.

In order to enhance the accuracy and smoothness of the hand gesture trajectories, cubic interpolation was applied to the raw sensor data. Cubic interpolation is a mathematical method that estimates data points between known data points using a smooth curve. It helps to overcome the limitations of discrete sensor data and improve the representation of the hand gestures for further analysis and modeling [136]. Figure 1.33 shows an example of cubic interpolated data for a long-range IR sensor.

The utilization of cubic interpolation significantly improved the accuracy of the NN

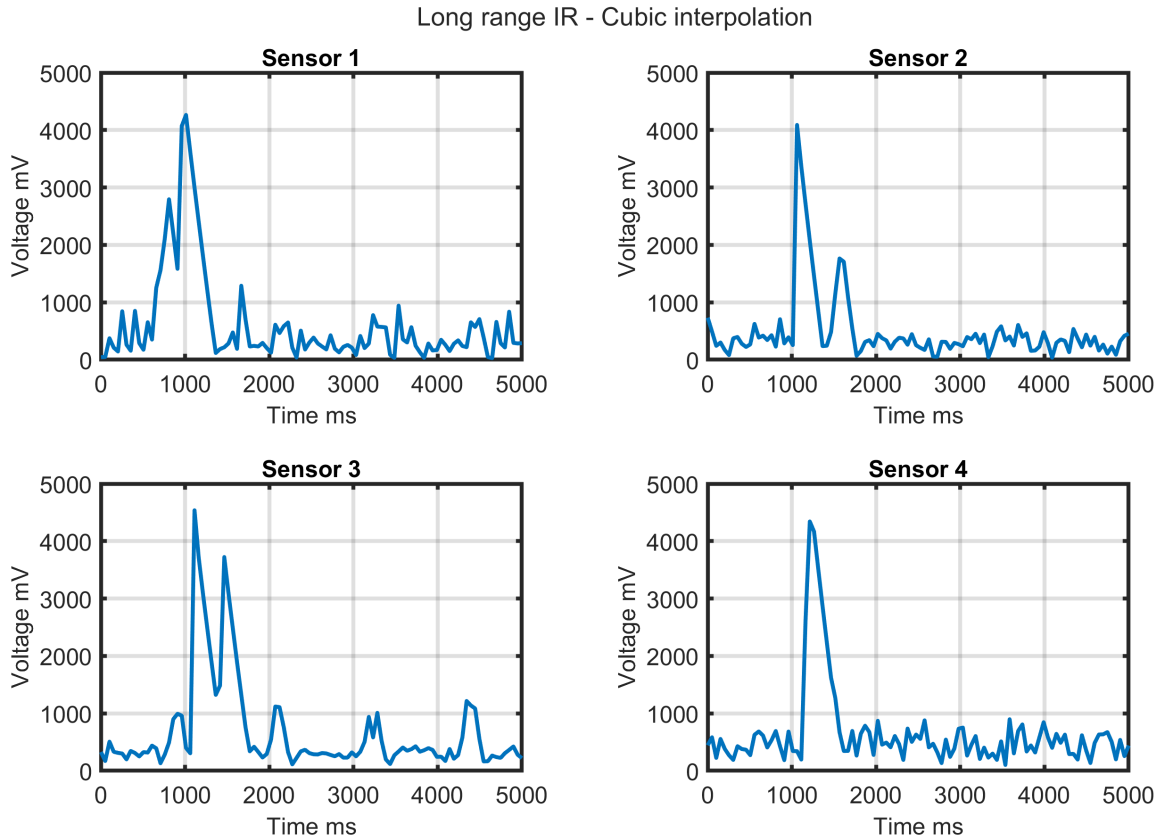


Figure 1.33. IR long interpolation

Table 1.25. ML results for cubic interpolated data

Table Sensor type	Accuracy		
	<i>SVM</i>	<i>RF</i>	<i>NN</i>
Capacitive	0.84	0.80	0.79
IR short-range	0.82	0.69	0.75
IR long-range	0.88	0.83	0.94

model when applied to interpolated sensor data. The results presented in the Table 1.25 reveal notable enhancements in accuracy for certain sensor types, such as capacitive and IR long-range sensors. However, a slight decrease in accuracy was observed for the IR short-range sensor. These findings underscore the substantial impact of cubic interpolation in augmenting the performance of the NN model for accurate gesture recognition.

The final architecture of the NN model displayed in this investigation is constructed of eight layers, as depicted in Figure 1.34. The first is the input to a 1D convolution layer with 32 filters and 16 kernel sizes. The convolution layer is followed by a flattening, which is then followed by a dense layer with 32 neurons. The dense layer is followed by a Dropout layer with a dropout rate of 0.2 and another dense layer with 32 neurons, which is again followed by another dropout layer with a dropout rate of 0.1. The last two layers are the Dense layer with 32 neurons and the final output layer. The applied activation functions were

ReLU (in dense layers) and Softmax (in the output layer). Our proposed architecture is a 1D Convolutional Neural Network (cNN), where the convolution layer extracts characteristic features from the signal input, and where Dense layers try to find relations between extracted features to classify signals. The dropout rate (probability of setting output from the hidden layer to zero) must be included because of the small training dataset, which prevents the overfitting of the network to a training dataset.

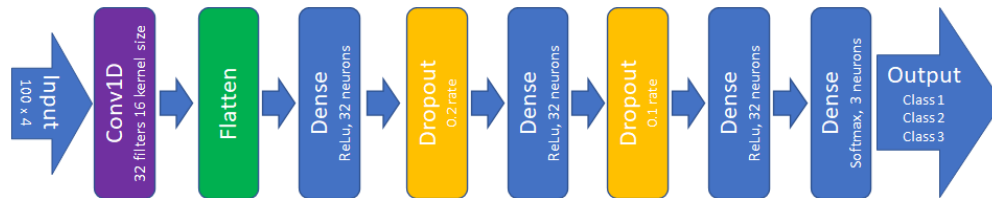


Figure 1.34. ANN Architecture

Since the classification of the hand gestures is a multiclass classification problem, the Categorical Cross-Entropy Loss function was applied as the loss function. Another key aspect of the ANN model architecture that was thoroughly examined is the selection of optimizers, learning rates, number of epochs, and batch size. Adam provided the most accurate estimation results on the test dataset with a 0.0005 learning rate. Optimal training results were obtained with 100 epochs and 128 batch size.

ML models were originally trained in all shapes, but the performance of the test data set showed unsatisfactory results with a categorical precision of 87.3%. By removing one gesture from the training and test dataset, categorical accuracy with the remaining four gestures increased to 89.8%, which was also unsatisfactory. Finally, by keeping only three gestures (namely circle, square, and pentagon), we have achieved better categorical accuracy. After performing several repetitions of the classification, the accuracy ranged from 93.8 % up to 98.3%, depending on the repetition. The results in the form of a confusion matrix for all three models are presented in Figure 1.35, and do not show which shape or gesture is to be blamed for the poor performance of the model with the five gestures in the training dataset.

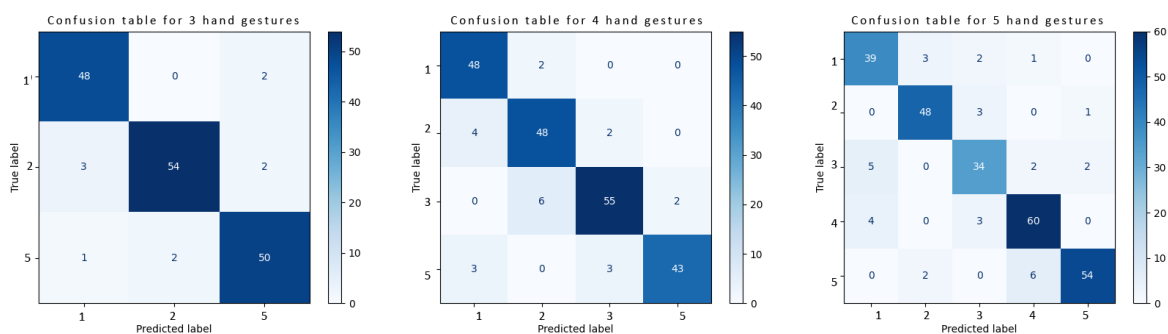


Figure 1.35. Confusion matrix for models that include 3 shapes (left), 4 shapes (middle), and 5 shapes (right)

We have analyzed raw training data from different subjects in search of a solution that

could eventually improve performance. An example of the analysis is presented in the form of a plot on Figure 1.36, for all five gestures-shapes for a single sensor and the same subject. As the system is capturing raw data in 10s intervals, useful data (when the user is performing a gesture) takes only a few seconds and can be found anywhere inside the original signal. As seen from the sample data presented in Fig.1.36, useful data takes only 2 seconds intervals per sensor while the rest of the data is extremely noisy. Relative timings and shapes of slopes between sensors capturing the same gesture are actual features that have to be extracted and used for gesture recognition and classification. By visual inspection of raw data for several examples (same person performing same gesture) some obvious similarities between signals cannot be easily found. Thus, this non-trivial task is delegated to our proposed ML model, which extracts those features and decides which gesture is performed. More detailed analysis of measured raw data from all four sensors on several subjects in the training set suggested that shapes 3 and 4 (namely triangle and rhombus) are similar to shapes 1 (circle). We presume that acquiring larger training data would improve the performance of a 5-shape ML model, by allowing it to find more specific features for each shape and consequently build a better model. Due to the aforementioned reasons, we have removed shapes 3 and 4 from the training and test data set.

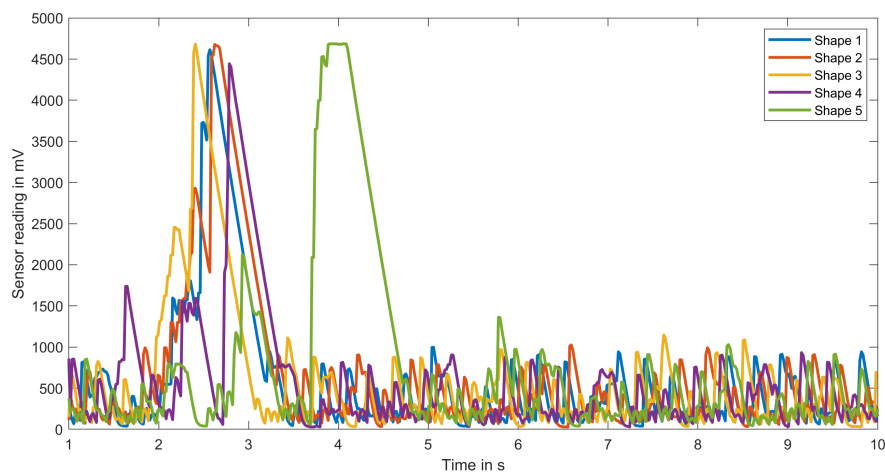


Figure 1.36. Raw sensor data for five hand gestures -shapes recorded with a single sensor

### Exploratory Pilot Study

This pilot evaluation collected data on children's experiences and perceptions of using IoT technology for educational purposes, focusing on usability, engagement levels, and motor aspects of interactions with a Smart Toy for early childhood geometry education. Exploratory pilot studies with children are crucial for identifying usability issues before larger-scale studies. Pilot testing and small sample sizes in child-related research have been highlighted in previous studies, emphasizing the need to involve children in the design process [137, 138].

Small sample sizes allow for iterative design processes and can reveal design flaws not apparent in larger studies [139].

In order to conduct the pilot study, the faculty Ethics Committee gave their positive opinion on the experiment procedure, stating that the proposed scientific research is carried out in accordance with the ethical principle of scientific integrity. All parents had signed a consent form before their children participated in the experiment. This exploratory study involved a small group of children (ages 4-7) interacting with the toy prototype and providing feedback. Hand movements during interaction were analyzed to guide future design and movement-based feedback. The collected data will inform toy design and the performance of the Machine Learning model in future research.

### ***Experiment Design and Procedure***

The study used a mixed-method approach, combining quantitative data from pre- and post-test tasks and usability testing with qualitative and quantitative data from video recordings, questionnaires, and interviews with children. The study was carried out in a controlled laboratory setting, with one-on-one interaction between the researcher, participants, and the proposed Smart toy. Experimental design along with materials and methods is further described.

The assessment process was based on a set of criteria that includes several quantitative and qualitative measures, which are expressed in terms of:

- Time-related aspects of interaction (time taken by the user to draw a shape and overall interaction duration);
- Hand gestures used to interact with the toy;
- Perceived ease of use (mapping of the particular shape);
- User mapping accuracy per particular shape;
- Engagement;
- Returnance (as one of the durability dimensions);
- Fun and Subjective satisfaction;
- Obtained knowledge.

Several measuring instruments were used to acquire the aforementioned quantitative and qualitative measures:

- Pre-test and Post-test: employed to evaluate the level of information acquisition as an indicator of the educational value.



- Attitude questionnaires (Smileyometer and The Again Again table) [140]: used to measure children’s fun and subjective satisfaction.
- Structured interview: used as an instrument to measure children’s fun and subjective satisfaction, level of engagement and as well as their perceived ease of use (mapping of the particular shape).
- Video recording: used as an instrument to measure motor aspects of interaction (hand gestures), time-related aspects of interaction, and engagement.
- Observation checklist: used as an instrument during the assessment process to record notes, document identified problems, and fill in additional information related to task completion accuracy.

Figure 1.37 represents the overall framework of the experiment.

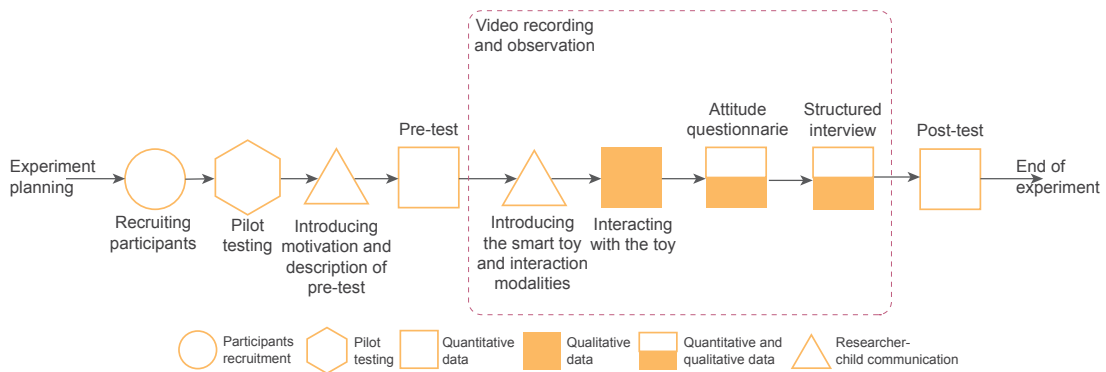


Figure 1.37. Overall framework of the experiment

Laboratory equipment utilized for the experiment were as follows:

- Cardboard geometric shapes and boxes
- Smart toy for geometry learning
- Computer for data collection
- Consent forms for parents/guardians

Figure 1.38 gives the graphical representation of the laboratory setup and equipment applied in the experiment.



Figure 1.38. Visualization of laboratory setup

Experiment Procedure was as follows:

- **Recruitment:** The study employed a convenience sampling strategy, which is a non-probability sampling method. Convenience sampling entails selecting study participants who are readily available and willing to participate. The preschool children were recruited from the University staff, including non-scientific personnel and personal networks of research aiming to ensure that the study sample was as diverse as possible. The study was explained to parents/guardians, who were asked to consent to their child's participation. Overall, 14 children (7 girls and 7 boys) aged from 4 to 7 years old participated in the pilot study. Inclusion criteria will include no previous exposure to the smart toy used in the study, as well as no history of developmental or learning disabilities.
- **Pre-test task:** Before interacting with the Smart toy, each child was given a pre-test task to assess their current knowledge of basic geometric shapes. Children were given 30 simple cardboard geometric shapes (namely 10 circles, 10 squares and 10 pentagons) of different color and size and were asked to put them in appropriate box for each of the shapes. The evaluation was administered orally by the researcher.
- **Interaction with the Smart Toy:** Each child had 30 minutes to play with the toy. The researcher observed the child and documented their level of participation, motor aspects of interaction, interest, and overall behavior while interacting with the smart toy.
- **Data Collection:** A video camera was used to record the participants during the experiment. It recorded the duration of the interaction, the accuracy of the completed task,

and any errors made by the participants. It also captured the levels of engagement and other aspects of interaction that children had with the toy. Furthermore, the data was also collected by the Smart toy in terms of sensor output data obtained from gesture movement.

- **Post-Test task:** After interacting with the smart toy, each child completed a post-test task the same as the one in the pre-test. They were again given 30 (new) simple cardboard geometric shapes (namely 10 circles, 10 squares, and 10 pentagons) of different color and size and were asked to put them in appropriate box (new) for each of the shapes. The evaluation was administered orally by the researcher. The pre- and post-test task were further utilized to examine the effectiveness of the Smart toy for geometry learning.
- **Follow-up Interview and Questionnaire:** The researcher asked close-ended questions about the child's engagement with the smart toy, ease of use, their learning experience, and subjective satisfaction while interacting.
- **Data Analysis:** Analyses of the overall collected data included statistical analysis while focusing on several aspects such as fun and subjective user satisfactions, ease of use, engagement, returnance, and motor aspects of interaction. The pre- and post-test results were compared to see if the interaction with the Smart toy significantly improved geometry knowledge. The results of the questionnaire, interviews, and video recordings were also be analyzed in order to gain insight into the child's level of engagement and overall satisfaction with the smart toy.

This pilot study utilized several techniques for evaluating children's experiences with a Smart Toy for early childhood geometry education. Firstly, simple cardboard geometric shapes were used for pre-test and post-test assessments. These shapes have been shown to be a valid tool in user evaluation studies and provide a tangible representation of geometric concepts [141, 142, 143, 144, 145]. Secondly, a structured interview, as presented in Table 1.26, was conducted to gather insights into children's fun, subjective satisfaction, and perceived ease of use while interacting with the Smart Toy. Interviews have been proven effective in investigating the user experience in studies involving children [146, 147, 148, 149, 150, 151].

To assess subjective satisfaction and fun, two instruments from the Fun Toolkit were employed. The Smileyometer, which uses a visual scale of smiley faces, was used to measure children's subjective experiences with the toy. The tool is based on a 5-point Likert scale (as presented in Figure 1.39), with responses ranging from 1 (awful) to 5 (excellent) (brilliant) [152]. The Again-Again table was used to assess children's desire to repeat an activity [140].

Table 1.26. Structured Interview

Questions	Aspects of exploration
1. Did you like the game?	Fun; Subjective satisfaction
2. Which shape was the easiest for you to draw?	Ease of mapping
3. Which shape was the hardest for you to draw?	Ease of mapping
4. Was the game boring?	Fun; Subjective satisfaction
5. Was the game difficult?	Fun; Subjective satisfaction
6. Would you like to play this game again?	Fun; Subjective satisfaction
7. What else would you like to teach the giraffe?	Engagement

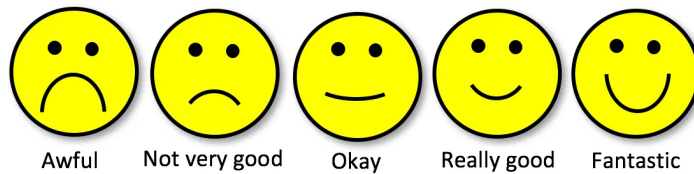


Figure 1.39. Smileyometer rating scale

The “Again-Again Table” was derived (presented in Table 1.27) from the original presented in [152]. The table was filled applying by the researcher asking the research question: “Would you like to draw this shape again?”

Table 1.27. The Again-Again Table

	Would you like to draw this shape again?		
	Yes	Maybe	No
Circle			
Square			
Pentagon			

Video recording was used to capture important aspects of interaction, including engagement, time-related aspects, and hand gestures. Video recording allows for detailed analysis of children’s behavior and movements, enabling researchers to identify patterns and areas for improvement in the user experience [153, 154, 155, 156]. These evaluation methods provided valuable data for assessing the effectiveness of the Smart Toy and informing future design improvements.

**Results**

Over the course of three consecutive days, 14 children participated in the pilot study. Among them were 7 girls and 7 boys. Ten preschool children were 6 years old, three were 4 and 5 years old and went to kindergarten, and one was 7 years old and is a first grader.

**a) Results regarding the objective aspects of interaction**

The pre-test was designed to assess children’s knowledge of a variety of geometric shapes appropriate for their ages in order to study the change after using the proposed Smart toy.

Children can touch, feel, and manipulate cardboard cutouts, which provide a tangible and physical representation of geometric shapes. This enables children to grasp and internalize geometric concepts and relationships. Most children do not have a thorough understanding of all geometric shapes at a young age so it was important to examine if they can appropriately distinguish and name them. Table 1.28 shows how children performed in the pre-test stage.

*Table 1.28. The number of correct and incorrect answers given by children in the pre-test stage when identifying geometric shapes.*

	Circle	Square	Pentagon
<b>Correct Answers</b>	100%	99.3%	99.3%
<b>Incorrect Answers</b>	0%	0.7%	0.7%

As can be seen, the children were good at distinguishing circles from squares and pentagons. However, due to the fact that cardboard geometric shapes were of different colors and sizes, on two occasions, a square was mistaken for a pentagon and vice versa.

Following the pre-test, the children were taken to a separate area of the laboratory where the Smart toy giraffe was placed, as shown in Figure 1.38. The entire interaction process was recorded on video and the researcher let the child become acquainted with the toy without intervention or specific instructions. The children were then asked if they wanted to "teach the giraffe" the geometric shapes they had been playing with in the pre-test. Each child had 30 minutes to interact with the Smart Toy. The researcher instructed them to draw (map) the shape from the LCD screen above the giraffe's back with their hands as if they were drawing on a canvas or a board. During interaction with the smart toy, the researcher observed the child and recorded motor aspects of the interaction, their level of participation, interest, and overall behavior. The researcher labeled each gesture made by the child as correct or incorrect. This was later verified by analyzing the video recording. For each shape, the researcher asked the child if they wanted to play a bit more. When the child expressed a desire to stop playing, he or she was interviewed and encouraged to take the post-test.

Firstly, four children did not establish the appropriate manner of interaction with the Smart Toy. Two of them were ages 4 and 5 (kindergarten) and eager to touch and cuddle the toy. They showed their emotions by smiling. The other two children were six-year-olds and tried to interact with the toy, however, they did not manage to do it. One of them did not show interest in the toy. This was especially evident in the fact that the child did not touch the giraffe at all. The other tried to do the gestures but gave up and continued to play with the toy in his own way. This child was interested in the toy and expressed emotions by smiling.

In total, ten children managed to interact with the toy in a suitable way. The primary aspect of the interaction observed was the formation of the gesture. According to the results, five children performed the interaction with a single finger (index finger). Four children

interacted with two fingers (thumb and index finger), while one child used the entire fist. Children who used one finger had longer interactions because they performed more gestures, while those who used two fingers or a fist had shorter interactions and performed fewer gestures, as exhibited in Figure 1.40. No child interacted with the toy for a planned period of 30 minutes. The majority of interactions lasted from around five to ten minutes. A child, a first-grader, engaged with the toy longest and managed to make a significant number of gestures.

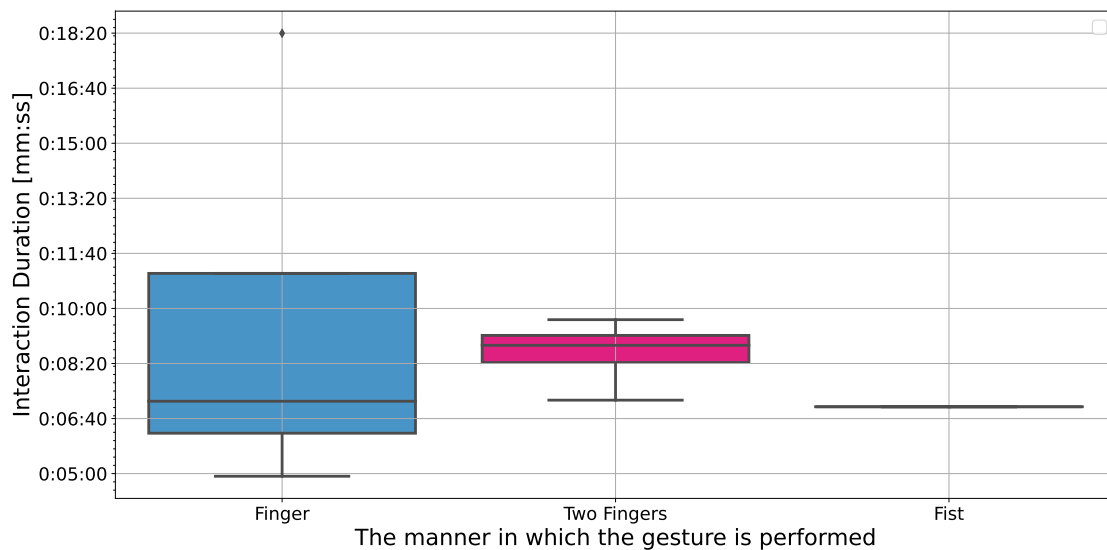


Figure 1.40. Interaction duration related to the established manner of the performed gesture

The time required to form a specific shape was the second aspect of the observed interaction. Figure 1.41 shows the distribution of the time required to perform a particular gesture. There is an evident and reasonable increase in complexity correlated with the time required to perform a given gesture, with a circle requiring the least time and a pentagon demanding the most, which was to be expected. In the case of the square shape, there is an outlier caused by one child's playfulness even though the gesture was correctly performed.

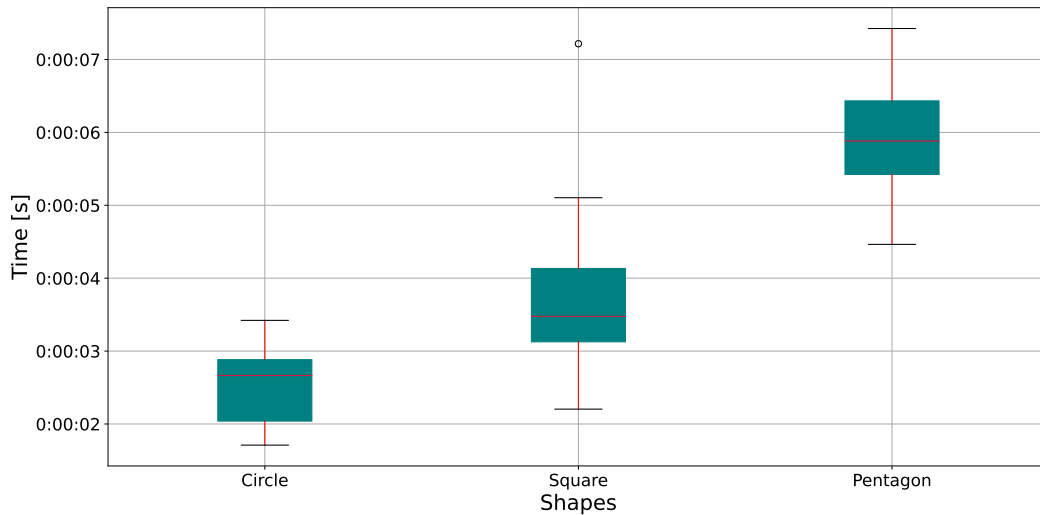


Figure 1.41. Gesture time per particular shape

To identify any potential confounding variables in our limited data sample size, we conducted a search for variables that are correlated with both the independent variable and the dependent variable. Through our investigation, we discovered that age was highly positively correlated with the number of user gestures, the number of correct user gestures per particular shape, and the number of correct user gestures. Specifically, the Pearson correlation coefficient for age and the number of user gestures was 0.77, while for the number of correct gestures for circle, square, and pentagon shapes, it was 0.74, 0.7, and 0.83, respectively. Additionally, the Pearson correlation coefficient for age and the number of overall correct user gestures was 0.77, indicating that the age may be a confounding variable that needs to be controlled for in analysis. We have therefore calculated partial correlation coefficients between the number of correct user gestures for per particular shape and the number of performed gestures, while controlling for the effect of age. We have found strong positive correlations between the number of correct user gestures and the number of performed gestures for the circle, square, and pentagon shapes, even after controlling for the effect of age. Specifically, the partial correlation coefficients were 0.942 (p-value=0.0001), 0.84 (p-value=0.004), and 0.899 (p-value=0.001) for the circle, square, and pentagon shapes, respectively. The statistically significant relationship between correct user gestures and performed gestures even after controlling for age suggests that age may not be a significant factor in predicting user performance for these shapes. This result may have implications for the future design of gesture-based interfaces, for instance for older children.

The final part of assessing the motor aspect of interaction was the accuracy of the child's gesture mapping. This will provide a subjective measure of the ease of mapping while interacting with the toy, which is an important aspect of user experience design. A gesture is considered correct if drawn on a virtual canvas above the sensors in the following way:

- a circle is drawn in 360 degrees, without overwriting the previous trajectory;

- the starting vertex for a square and pentagon is the same as the ending one, without repetition of previous edges.

This was evaluated in real-time by the researcher during the experiment, and validated by examining the video footage. The results presented in Figure 1.42 show a somewhat different and unexpected order of complexity among different shapes. That is, a circle has a higher failure rate than a square. This is most likely the result of outlining multiple circles on existing ones. As assumed, the failure rate for a pentagon is the highest.

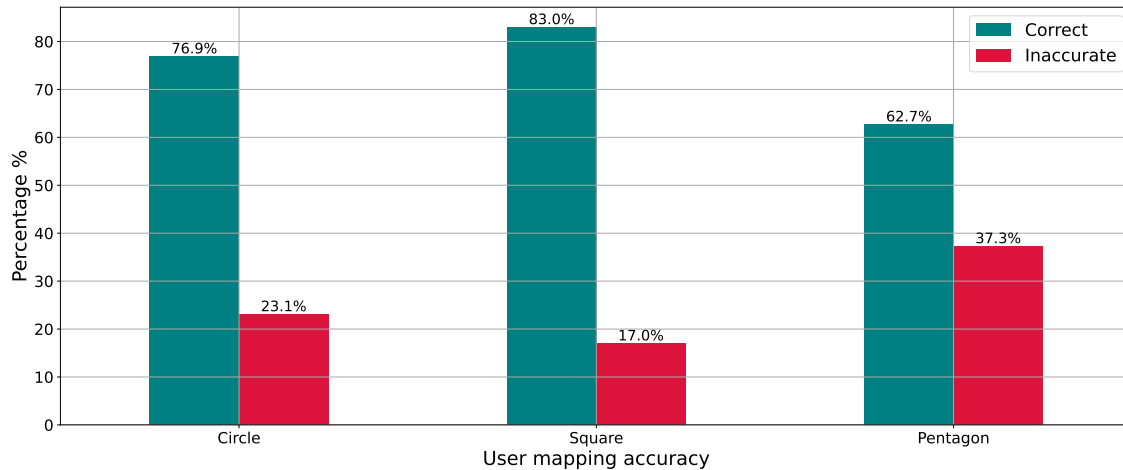


Figure 1.42. User mapping accuracy per particular shape

**b) Results regarding the subjective aspects of interaction**

These results are onward compared with the child’s subjective experience related to the ease of mapping. Based on the answers provided from the interview questions "2. Which shape was the easiest for you to draw?" and "3. Which shape was the hardest for you to draw?" the following results were obtained and presented in Figure 1.43.

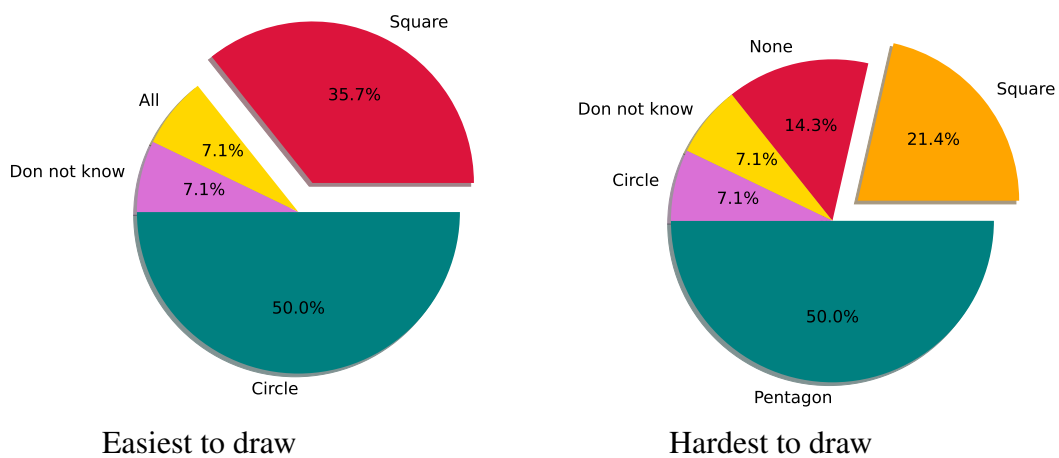


Figure 1.43. The results of the answers to interview questions 2 and 3.



As can be seen, the children perceived the circle to be the easiest shape to map, as opposed to the pentagon, which they perceived to be the most difficult. This result correlates with the distribution of the time required to perform a specific gesture, with the circle requiring the least time and the pentagon requiring the most. However, these results are in contrast to the objective user mapping accuracy, as the square was the most accurately mapped shape. As was previously mentioned, this is probably due to the fact that a great number of children draw the circle by outlining multiple circles over existing ones.

Furthermore, the relationship between the perceived difficulty of different shapes and the actual time required to draw them was examined. The Mann-Whitney U test was used to compare the time taken to draw the hardest/most time-consuming shape (pentagon) with the time taken to draw the easiest/least time-consuming shape (circle). The null hypothesis, which stated that there would be no significant difference in time taken between the two shapes, was rejected based on the results of the test. The statistic was calculated to be 0.000000 and the p-value was found to be 0.00041, indicating a significant difference in time taken between the two shapes. This suggests that the perceived difficulty of the shapes corresponds to the actual time required to draw them. These findings have implications for the design of educational materials and activities that involve drawing shapes, as they suggest that the time required to draw a shape can be used as an objective measure of its difficulty.

Regarding the results from the children's subjective impressions of fun and satisfaction, valuable feedback from the children about their subjective experiences with the smart toy was obtained. Table 1.29 provides information on children's responses to question "Can you show me, using these pictures, how you felt while playing this game?"

Table 1.29. Fun and subjective satisfactions measured with the Smileyometer rating scale.

The Smileyometer rating scale results					
	Awful	Not Very good	Okay	Really good	Fantastic
<b>Number of children</b>	0 (0%)	1 (7.1%)	2 (14.3%)	2 (14.3%)	9 (64.3%)

As can be seen, the dominating majority of children expressed a feeling of "Really good" and "Fantastic" while interacting with the smart toy. These results indicate that the children are enjoying the activity and experiencing positive subjective satisfaction. This may also imply that, in future interactions, children are more likely to fully engage in toy play. These implications are supported by the results obtained from children's responses to interview questions "4. Was the game boring?" and "5. Was the game difficult?", presented in Table 1.30.

Table 1.30. The results of the answers to interview questions 4. and 5

	Yes	No
Was the game boring?	3	11
Was the game difficult?	1	13

The children perceived the play with the giraffe to be engaging and easy. Such positive experiences indicate that the toy is meeting expectations, which can be an important factor in promoting children’s learning since they are more likely to continue using the toy. The latter might overall result in greater technology adoption and success.

These implications are in correlation with the results obtained by analyses of video recordings of children’s expressions and behavior during toy interaction. The majority of children (12) smiled and were happy while interacting with the toy, one child danced and others bounced excitably. They were also keen on touching, petting, and exploring the toy, while at the same time communicating with the researcher. It was also noticed, that some children, four of them, were more concentrated on the task itself, rather than on the toy itself. Although they said they feel good interacting with the toy, they did not engage in other types of play with the toy apart from the proposed interaction. They were more interested in the toy’s educational features. When asked, "7. What else would you like to teach the giraffe?" the majority of children just smiled and were unsure what to say other than "I don’t know." However, some children provided rather interesting answers like "I would like to teach her letters", "I would like the draw hearts", and one child answered "I would like to teach her about good behavior."

Finally, the results of the returnability aspect based on the responses from Again-Again Table 1.27 are presented in Figure 1.44.

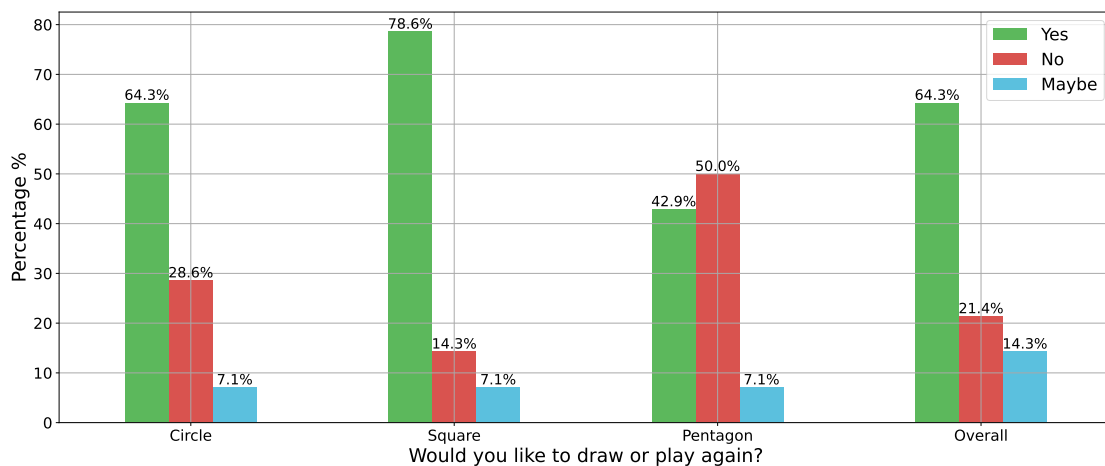


Figure 1.44. Results from the responses from Again-Again Table 1.27

Results indicate that the majority of children would like to play with the toy again. Furthermore, findings suggest that the children found the square shape to be the most engaging

and interesting to play with, as evidenced by their desire to play with it again and their preference for drawing the square. This preference may be related to the objective user mapping aspect, in which the square was the most accurately mapped shape. It is also worth noting that, despite the children's subjective assessment that the circle was the easiest shape to draw, they preferred drawing the square. This suggests that a child's interest in the toy was not solely determined by its ease of use. Overall, these findings suggest that future enhancements to the toy's design should consider not only the ease of use but also the toy's engagement factor. The objective user mapping aspect can also be considered to increase engagement. As was to be expected, half of the children would not want to draw the pentagon again. It's possible that the children's lack of interest in drawing the pentagon again is related to their level of motor skill development, as the pentagon has more sides and angles than the other shapes, potentially making it more difficult to draw. They may also feel less confident or interested in attempting to draw the pentagon again or that they found the pentagon more challenging to understand or remember compared to the other shapes. This implication is supported by the researcher's observations as well as the video analyses, as none of the children were familiar with the shape or knew its name, and usually referred to it as the "house shape".

An immediate post-test followed the interaction with the toy. The results of the test are presented in Table 1.31. Only the results of children who interacted with the toy were taken into account. As can be noticed, the accuracy of recognizing and classifying the pentagon seems to decline. This was probably an immediate result of the fatigue of one child who incorrectly classified the pentagon as a square several times since this child has interacted with the toy the longest and has performed a great number of gestures. Overall, due to the small sample size, a definitive conclusion about the impact of the toy on children's performance on the post-test cannot be drawn. Therefore, in the future, it is important to ensure that sample sizes are adequate to make accurate claims about the impact of the toy on children's educational performance.

*Table 1.31. The number of correct and incorrect answers given by children in the post-test stage for identifying geometric shapes.*

	Circle	Square	Pentagon
<b>Correct Answers</b>	99%	99%	95%
<b>Incorrect Answers</b>	1%	1%	5%

### *c) Machine Learning performance*

Finally, the performance of the Neural Network in experimental scenarios is presented. Children performed an overall amount of 111 different gestures and Table 1.32 provides insight into gesture classification accuracy.

As can be observed, the classification accuracy is quite low, especially for the pentagon shape. There are several possible reasons for such a bad performance. To begin with, our

Table 1.32. Machine Learning gesture classification accuracy results

	Circle		Square		Pentagon	
	Gussed	Missed	Gussed	Missed	Gussed	Missed
<b>Number of gestures</b>	25	15	19	20	6	26

experimental results have shown that children’s gestures differ from adult gestures in terms of frequency and execution. As demonstrated, children performed gestures primarily with their index fingers, while the data used to build the model came from adult users who primarily used their entire fists. Furthermore, of those 111 gestures, half came from a single user, the first grader, that is, 46, who made gestures with his index finger, while the other 55 gestures were distributed among the other children, indicating an imbalance in the test set. With that regard we have later done a comparison of raw sensor data from a child and an adult subject while performing gestures for a same geometrical shape, as presented in Figure 1.45.

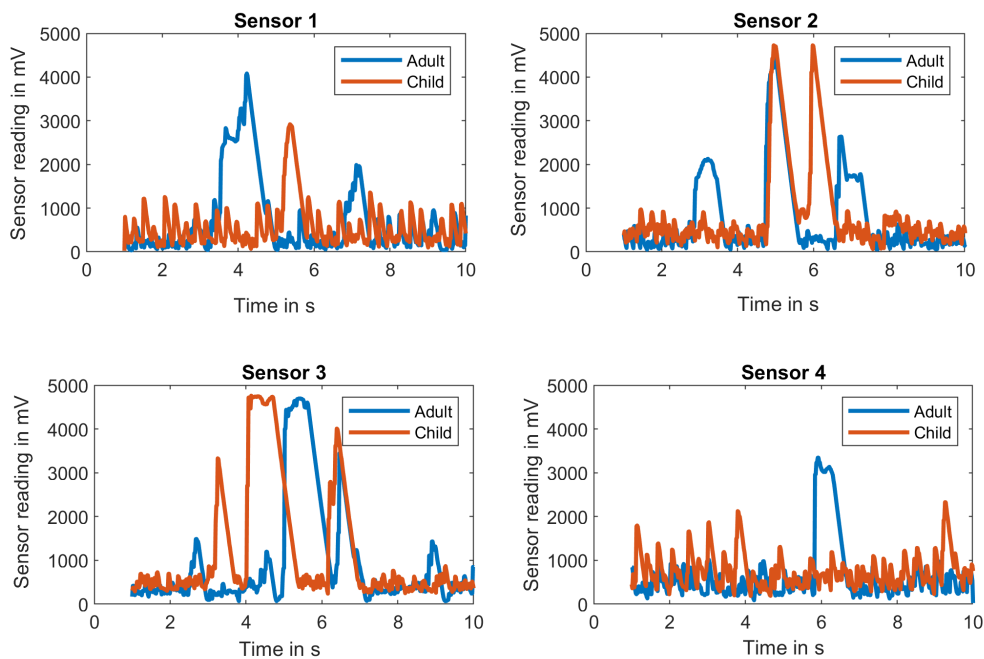


Figure 1.45. Comparison of raw sensor data of child and adult subject while performing gestures for a same geometrical shape

It can be observed that children’s gestures greatly differ from adult gestures, both in terms of their frequency and the way they are executed. Adults generally performed gestures with the entire fist, where, as we have seen, the children primarily used their index finger. Therefore, the feature set used to train the Machine Learning model was unable to accurately capture the variation in children’s gestures, leading to poor classification accuracy.

As this was a pilot study, it provided valuable information on the performance of the smart toy for geometry learning and identified areas for improvement. This information will be used to guide future design iterations, helping to refine Smart the toy and improve the

accuracy of the Machine Learning algorithm. In that regard, it will be necessary to collect a large data set of children's gestures and train the Machine Learning model specifically on this data set. This can involve collecting data from a range of ages and developmental stages to ensure that the model can capture the variation in children's gestures. Additionally, it may be necessary to develop new feature sets or modify existing ones to better capture the unique features of children's gestures. Finally, it may be necessary to test the model on a separate validation data set to ensure that it generalizes well to new examples of children's gestures.

### **1.2.5 The fundamental scientific contributions**

The fundamental scientific contribution of this research lies in the utilization of Machine Learning techniques to optimize and improve the performance of Internet of Things stack services. This contribution is exhibited through three main domains:

1. Enhancement of the Perception Layer service within the IoT architecture through a novel approach that leverages the signal strength of LoRaWAN devices and utilizes Machine Learning algorithms to detect states and changes in the environment. The improvement is exhibited in better cost-effectiveness and robustness with high precision compared to existing solutions.
2. Enhancement of the Network Layer service within the IoT architecture with a new model for frame size estimation and tag estimation in RFID Gen2 systems that utilize the ALOHA protocol. A model based on machine learning algorithms is proposed. The improvement is exhibited in better throughput compared to the state-of-the-art algorithm, with appropriate execution time to meet the protocol requirements.
3. A new interface for interaction on the Application Layer of the IoT architecture that utilizes Machine Learning techniques to recognize complex human gestures from sensor output data.

The Figure 1.46 visually represents the embodiment and potential implementation of the proposed solution. It serves as an illustration of the fundamental scientific contribution, demonstrating the utilization of Machine Learning techniques to optimize and enhance the performance of Internet of Things for specific use case scenarios explored within research conducted within this doctoral dissertation.

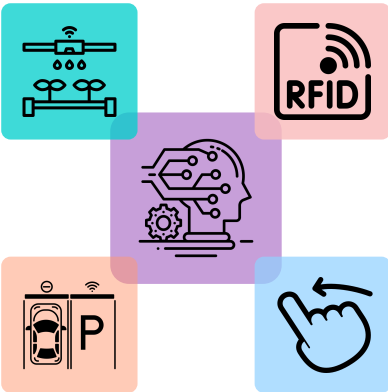


Figure 1.46. Realization and potential implementation of solutions

### 1.3 List of Published Papers Upon Which the Dissertation's Scientific Contribution was Founded

The proposed doctoral dissertation aims to improve the perception, communication, and application layers of the IoT architecture by utilizing Machine Learning algorithms. Through these novel approaches, the dissertation provides original scientific contributions to the field. These fundamental contributions cover all three layers of the IoT stack and are confirmed by the publication of scientific articles that collectively form a comprehensive and innovative scientific contribution to the field of information processing in the Internet of Things environment. A list of the published papers forming the basis of this dissertation's contribution is provided below.

- [1] Dujić Rodić L, Stančić I, Čoko D, Perković T, Granić A. Towards a Machine Learning Smart Toy Design for Early Childhood Geometry Education: Usability and Performance. *Electronics*. 2023; 12(8):1951. <https://doi.org/10.3390/electronics12081951>
- [2] Dujić Rodić, Lea; Perković, Toni; Škiljo, Maja; Šolić, Petar, Privacy leakage of LoRaWAN smart parking occupancy sensors. *Future generation computer systems*, 138 (2022), 142-159, doi:10.1016/j.future.2022.08.007
- [3] Dujić Rodić, Lea; ; Stančić, Ivo.; Zovko, Kristina; Perković, Toni; Šolić, Petar, Tag Estimation Method for ALOHA RFID System Based on Machine Learning Classifiers. *Electronics (Basel)*, 11 (2022), 16; 2605, 20, doi:10.3390/electronics11162605
- [4] Dujić Rodić, Lea; Granić, Andrina, Tangible Interfaces in Early Years' Education: A Systematic Review. *Personal and ubiquitous computing (2021)*, doi:10.1007/s00779-021-01556-x
- [5] Dujić Rodić, Lea; Perković, Toni; Županović, Tomislav; Šolić, Petar, Sensing Occupancy through Software: Smart Parking Proof of Concept. *Electronics*, 9 (2020), 12; 2207, 28. doi:10.3390/electronics9122207
- [6] Dujić Rodić, Lea; Županović, Tomislav; Perković, Toni; Šolić, Petar; Rodrigues, Joel J. P. C. Machine Learning and Soil Humidity Sensing: Signal Strength Approach. *ACM Transactions on Internet Technology (2020)* doi:10.1145/3418207

### 1.3.1 Other publications

In addition to the research presented in the aforementioned journal papers, the author has undertaken a significant volume of supplementary work, which has been disseminated through journal and conference papers, encompassing diverse domains of computer science, particularly in the field of Internet of Things. This work extends the breadth of the author's contributions to various aspects of IoT research. Furthermore, the author's scholarly output includes publications in the wider discipline of computer science, reflecting their interdisciplinary approach and engagement with different facets of computational techniques and IoT applications.

- [1] Perković, Toni; Dujić Rodić, Lea; Šabić, Josip; Šolić, Petar. Machine Learning Approach towards LoRaWAN Indoor Localization. *Electronics* (Basel), 12 (2023), 2; 1-23 doi:10.3390/electronics12020457 (international peer review, article, scholarly) 2023;12(8):1951.<https://doi.org/10.3390/electronics12081951>
- [2] Škiljo, Maja; Šolić, Petar; Blažević, Zoran; Dujić Rodić, Lea; Perković, Toni. UHF RFID: Retail Store Performance. *IEEE Journal of Radio Frequency Identification*, 6 (2022), 481-489 doi:10.1109/JRFID.2021.3129694 (international peer review, article, scholarly)
- [3] Čoko, Duje; Stančić, Ivo; Dujić Rodić, Lea; Čošić, Dora. TheraProx: Capacitive Proximity Sensing. *Electronics* (Basel), 11 (2022), 3; 393, 16 doi:10.3390/electronics11030393 (international peer review, article, scholarly)
- [4] Škiljo, Maja; Blažević, Zoran; Dujić-Rodić, Lea; Perković, Toni; Šolić, Petar Self-Sensing Antenna for Soil Moisture: Beacon Approach. *Sensors*, 22 (2022), 24; 9863, 14 doi:10.3390/s22249863 (international peer review, article, scholarly)
- [5] Dujić Rodić, Lea; Stančić, Ivo; Zovko Kristina; Šolić, Petar Machine Learning as Tag Estimation Method for ALOHA-based RFID system. 2021 6th International Conference on Smart and Sustainable Technologies (SpliTech) Bol (Brač), Hrvatska: IEEE, 2021. pp. 1-6 doi:10.23919/SpliTech52315.2021.9566455 (lecture, international peer review, full paper, other)
- [6] Čoko, Duje; Dujić Rodić, Lea; Perković, Toni; Šolić, Petar Geometry from Thin Air: Theremin as a Playful Learning Device. *Proceedings of the 16th International Conference on Telecommunications ConTEL 2021 / Antonić, Martina ; Babić, Jurica (ed.)*. Zagreb: Faculty of Electrical Engineering and Computing, 2021. pp. 89-96 doi:10.23919/contel52528.2021.9495979 (lecture, international peer review, full paper, scholarly)



- [7] Dujčić Rodić, Lea; Perković, Toni; Zupanović, Tomislav; Solić, Petar Markov Model as Approach to Parking Space Occupancy Prediction. IEICE Proceedings Series 2021 Niš (Serbia), (On-line): The Institute of Electronics, Information and Communication Engineers, 2021. 9, 4 doi:10.34385/proc.64.ICTF2020\_paper\_9 (lecture, international peer review, full paper, scholarly)
- [8] Dujčić Rodić, Lea; Perković, Toni; Šolić, Petar; Škiljo, Maja; Blažević, Zoran RFID Performance Evaluation in a Retail Store. Proceedings of the 5th International Conference on Smart and Sustainable Technologies (SpliTech) Bol i Split, Hrvatska (virtual), 2020. pp. 1-3 doi:10.23919/SpliTech49282.2020.9243698 (lecture, international peer review, full paper, scholarly)
- [9] Dujčić Rodić, Lea; Granić, Andrina Tangible User Interfaces for Enhancement of Young Children's Mathematical Problem Solving and Reasoning: A Preliminary Review of Relevant Literature. CECIIS 2018: 29th Central European Conference on Information and Intelligent Systems: Proceedings / Strahonja, Vjeran ; Kirinić, Valentina (ed.). Varaždin: Faculty of Organization and Informatics, University of Zagreb, 2018. pp. 77-84 (lecture, international peer review, full paper, scholarly)

## **1.4 Overview of the Dissertation Structure**

The introductory chapter of this doctoral dissertation presents the motivation for the study and the corresponding hypotheses. The following section of the introduction describes and explains the scientific methods employed to verify the hypotheses and realize scientific contributions. Furthermore, an overview of the scientific literature upon which the dissertation's contributions are based is provided. The second chapter is a review article that focuses on the doctoral candidate's contributions across three main research areas. The third chapter offers a detailed review of the scientific contributions of the papers upon which the dissertation is founded, with special emphasis on the candidate's contributions to each individual work. The fourth chapter consists of the concluding remarks of the entire doctoral dissertation, including a discussion of future research. Finally, relevant literature is cited and the published scientific papers (Appendix A, B, C, D, E, F) upon which the scientific contribution of this dissertation is based are included

## 2 STATE OF THE ART

The State of the Art chapter of this dissertation provides a comprehensive overview of scientific literature, divided into four parts. The first part presents an overview of the advancements and key concepts in Machine Learning, highlighting its relevance in the context of IoT systems. Additionally, this part will incorporate the scientific definitions of the applied algorithms that are pertinent to the research, enriching the understanding of the subject matter. The subsequent three parts are focused on the Perception layer, Network Layer, and Application Layer of the three-layer IoT stack, respectively. Each part delves into the specific challenges and application domains that have been the primary focus of scientific research within this dissertation. This structured approach ensures a thorough exploration of the contemporary scientific achievements in each layer, shedding light on the advancements and areas of interest in the field.

### 2.1 Machine Learning: General Overview

In the IoT paradigm of numerous smart connected devices, Machine Learning has emerged as an essential field of research and application aiming at providing computer programs the ability to automatically improve through experience [157]. The most distinguished attribute of a learning machine is that the trainer of learning machine is ignorant of the processes within it [158]. Origins of Machine Learning have its foundation in the study of pattern recognition and Computational Learning Theory and can be considered an integral part of Artificial Intelligence utilized for development of algorithms based on relationships between data [159]. Machine Learning generally includes data processing, training, and testing phases with the aim of making the system able to carry out decisions based on the input received from the training phase [160]. In order to archive the learning process, systems use various algorithms and statistical models to analyze the data and gain information about the correlation between the data features [161]. The algorithms that are used in these processes can be divided into four distinctive groups, as Supervised, Unsupervised, Semi-supervised, and Reinforcement learning algorithms:

- Supervised learning algorithms demand external monitoring by a supervisor with the goal of learning how to map input values to the output values where the accurate values are given by a supervisor [162].

- Unsupervised learning algorithms make computers learn how to perform a specific task only with the provided unlabeled data. These types of algorithms need to find existing relationships, irregularities, similarities, and regularities in provided input data [163].
- Semi-supervised learning is a hybrid approach of the previous two categories that uses both labeled data and unlabeled data. These algorithms generally act like the unsupervised learning algorithms with the improvements that are brought from a portion of labeled data [164].
- Reinforcement learning algorithms operate with a restricted insight of the environment and with limited feedback on the quality of the decisions. In order to operate effectively and provide the most positive outcome, these algorithms have the ability to selectively ignore irrelevant details [165].

Machine Learning has proven to be highly effective in addressing a wide range of problems, including classification, clustering, prediction, and pattern recognition, among others [161]. The selection of the most suitable ML algorithm depends on factors such as the speed of the technique and its computational intensity, tailored to the specific application requirements [161].

In recent years, Deep Learning has emerged as a prominent approach within the field of ML, particularly in IoT applications [166]. DL exhibits robust capabilities in mining real-world IoT data, even in noisy and complex environments, surpassing traditional ML techniques. This makes DL a powerful analytical tool, particularly in handling vast amounts of data and achieving superior performance in various tasks [167]. Unlike traditional ML techniques that heavily rely on the quality and accuracy of manually engineered features, DL has the ability to automatically extract and organize multiple levels of information, thereby effectively capturing complex relationships within the data [168]. ML, including DL, has found widespread applications across different fields, such as computer vision, computer graphics, natural language processing (NLP), speech recognition, decision-making, intelligent control [169], and even intrusion detection systems [170]. Within the realm of IoT devices, the application of ML techniques enables users to uncover deeper insights into data correlations and extract hidden information and features that might otherwise remain obscured [171]. This capability facilitates enhanced data analysis and decision-making processes in IoT environments.

## 2.1.1 Algorithms

### 1. Support Vector Machine

The idea of Support Vector Machine (SVM) was introduced by Vapnik in mid 1990-ties and today this a well known machine learning algorithm used in various applications from

classification, forecasting to pattern recognition. The SVM implements the idea of mapping input vectors into a high-dimensional space  $\mathcal{F}$ , which is furnished with a dot product, using a non-linear mapping selected a priori [172]. This idea has been generalized to become applicable to regression problems using Support Vector Regression (SVR) briefly presented in the following.

Let us consider a training set  $T = \{(x_i, y_i) \mid x_i \in \mathbb{R}^n, y_i \in \mathbb{R}, i = 1, \dots, n\}$ , where  $X = (x_1, \dots, x_n)$  are sampling data and  $Y = (y_1, \dots, y_n)$  target vaules. The objective of SVR is to find function  $f(x)$  that has at most  $\varepsilon$ - deviation from the observed target  $y_i$  for all training data, enforcing flatness. This function can be defined as a linear function

$$f(x) = \omega\Phi(x) + b, \quad (2.1)$$

where  $\Phi : \mathbb{R}^n \rightarrow \mathcal{F}$  is the map into the higher dimensional feature space,  $\omega$  represents vector of wights of the linear function and  $b$  is the bias. Desired function which is optimal is chosen by minimizing the function

$$\Psi(\omega, \xi) = C \cdot \sum_{i=1}^n (\xi_i + \xi_i^*) + \frac{1}{2} \|\omega\|^2, \quad (2.2)$$

where  $\xi, \xi^*$  are non negative slack variables that measure the upper and lower excess deviation,  $\|\cdot\|$  is the Euclidean norm ( $\frac{1}{2} \|\omega\|^2$  represents regularization term),  $C$  is a regularization parameter which allows the tune of the trade-off between tolerance to empirical errors and regularization term.  $\Psi(\omega, \xi)$  must satisfy following constraints:

$$\begin{cases} y_i - \omega\Phi(x_i) + b_i \leq \varepsilon + \xi_i \\ \omega\Phi(x_i) + b_i - y_i \leq \varepsilon + \xi_i^* \\ \xi, \xi^* \geq 0, i = 1, \dots, n \end{cases} \quad (2.3)$$

Furthermore, the most prominent feature of SVR is the ability to establish correlation between data using non-linear mapping. This is achieved using kernel functions for generating the inner products, know as kernels, which satisfy Mercer's theorem. One of the broadly used kernels are polynomial and Gaussian radial basis function (RBF) kernels. The RBF kernel is given with the formula:

$$K_\gamma(|x - x_i|) = \exp \left\{ -\gamma \cdot |x - x_i|^2 \right\} \quad (2.4)$$

The necessary parameters  $\gamma, C$  and  $\varepsilon$  can be selected with grid search process of performing hyper parameter tuning in order to determine the optimal values for a given model.

## 2. Random Forest

Random Forest as an ensemble learning Machine Learning approach to classification and regression was first introduced by Breiman in 2001. [173]. It has successfully been utilized in many research and application domains and has become a standard in non-parametric classification and regression Machine Learning technique for making predictions based on different types of variables without making any prior assumption of how they are associated with the target variables [174]. Its application ranges from bioinformatics [175], intrusion detection systems [176] computer vision [177], RS land cover classification [178], as well traffic accident detection [179], crop classification based on object-based image analysis [180] and DDoS attack detection [181].

Formally, RF can be defined as a classifier constructed out of a collection of tree-structured classifiers  $\{c_k(x, T_k)\}, k = 1, \dots, L$ , where  $T_k$  are independent identically distributed random samples (vectors) and for a input  $x$ , each of the trees casts a unit vote for the most popular class [182] as depicted in Figure 2.1.

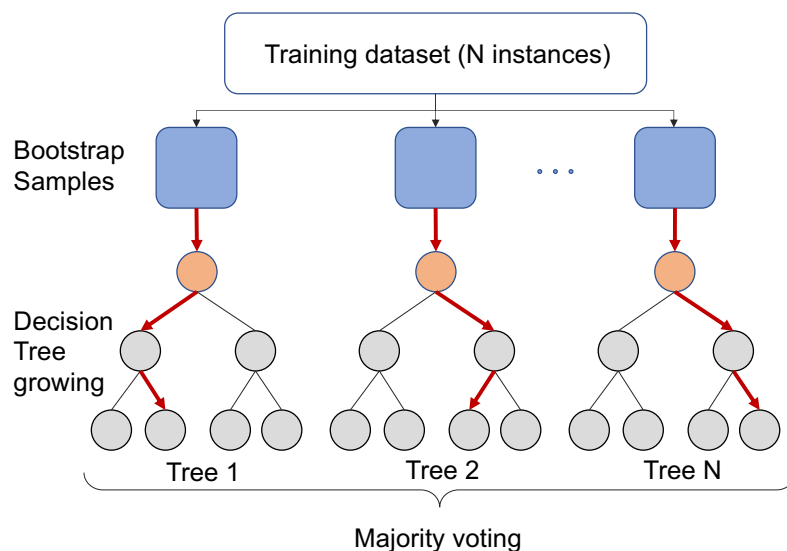


Figure 2.1. Example of an Architecture of Random forest model.

The trees are generated using a bagging approach, that is by producing random samples of training sets through replacement, where some samples can be taken several times and others may not be taken at all [183]. For a given training set  $T$  constructed classifiers  $\{c_k(x, T_k)\}$  cast a vote and make the *bagged* predictor and for each  $y, x$  in the train set the votes from classifiers for which the  $T_k$  did not contain  $y, x$  are stored as *out-of-the-bag* classifiers [173]. Generally, samples used for training the trees are taken from two thirds of the instances and the remaining one third are used in an inner cross-validation technique that estimates the resulting RF model performance [183]. The *out-of-the-bag* estimate for the generalization error is the error rate of the *out-of-the-bag* classifier on the training set and this estimate is as accurate as using a test set of the same size as the training set, thus remov-

ing the need for a set aside test set [173]. Commonly, user defines the number of trees and the algorithm creates trees that have high variance and low bias where each decision tree is independently produced without any pruning, and each node is split using a user-defined number of features that are randomly selected [183]. The error rate will decrease as the number of combinations increases and therefore the *out-of-the- bag* estimates will tend to overestimate the current error rate and therefore it is necessary to run past the point where the test set error converges [173]. Final classification is done by taking the average of class assigned probabilities calculated by all generated trees and new data given as an input is accordingly evaluated against all decision trees produced in the ensemble and each tree votes for a class membership [183]. The class which has the biggest overall number of votes is the one that is chosen in the end.

### 3. Hidden Markov Models

Hidden Markov Models (HMMs) have been known for decades and today are making a large impact with regard to their applications, especially in form of Machine Learning models and applications in reinforcement learning. They are widely being used for pattern recognition [184], i.e. namely speech recognition [185] as well as in biological sequence analysis [186], gene sequence modeling, activity recognition [187] and analyses of ECG signal [188, 189]. Markov Chains and process were introduced by the Russian mathematician Markov in 1906 when he obtained a theoretical result for a stochastic process. Markov process can be considered a time-varying random phenomenon for which Markov properties are attained. Its practical importance is the use of the hypothesis that the Markov property holds for a certain random process in order to build a stochastic model for that process [190]. Such a process has a fixed number of states, and it randomly evolves from one state to another at each step. The probability for it to evolve from state  $a$  to a state  $b$  is fixed, and it depends only on the pair  $(a, b)$ , not on past states (the system has no memory) [191].

In the broadest sense, a Hidden Markov model (HMM) is a Markov process that can be divided into two parts: an **observable** component and an unobservable or **hidden** component. The observation is a probabilistic function of the state, i.e. the resulting model is a doubly embedded stochastic process, which is not necessarily observable, but can be observed through another set of stochastic processes that produce the sequence of observations. A machine learning algorithm can apply Markov models to decision making processes regarding the prediction of an outcome.

In 1986 Rabiner and Juang [87] gave the structure of the first order Hidden Markov Model denoted as  $\lambda(A, B, \pi)$ , where  $A = \{a_{ij}\}$  is the matrix of transition probabilities,  $B = \{b_j(k)\}$  is the matrix of observation probability distribution in each state and  $\pi$  is the initial state distribution. Rabiner (1989) presented [192] three different types of problems in HMM: The Evaluation Problem, Decoding problem and Learning.

1. **The Evaluation Problem.** Given the observation sequence  $O = o_1, o_2, \dots, o_T$ , and the model  $\lambda(A, B, \pi)$ , how to compute  $\mathbb{P}(O \mid \lambda)$ , the probability of the observation sequence.
2. **Decoding.** What is the most likely state sequence in the given model that produced the given observations.
3. **Learning.** How to adjust the model parameters  $\lambda(A, B, \pi)$  to maximize  $\mathbb{P}(O \mid \lambda)$ .

The first problem is commonly solved by using the Forward or Backward algorithm, where as the last problem is, the most difficult of the three problems, usually solved using Baum-Welch method. With regards to the second problem the central issue is to find the optimal sequence of states to a given observation sequence and model used. Most common method to this is by using the **Viterbi algorithm**, introduced by Andrew Viterbi in 1967 as a decoding algorithm for convolution codes over noisy digital communication links. It is the answer to the decoding problem resulting in the Viterbi path, since the algorithm can be interpreted as a search in a graph whose nodes are formed by the states of the HMM in each of the time instant [190].

**Definition 2.1.1.** Let  $\lambda(A, B, \pi)$  be a HMM and  $O = (o_1, o_2, \dots, o_T)$  given observations. The Viterbi algorithm finds single best state sequence  $q = (q_1, q_2, \dots, q_T)$  for the given model and observations. The probability of observing  $o_1, o_2, \dots, o_t$  using the best path that ends in state  $i$  at the time  $i$  given the model  $\lambda$  is:

$$\delta_t(i) = \max_{q_1, q_2, \dots, q_{t-1}} \mathbb{P}(q_1, q_2, \dots, q_{t-1}, q_t = i, o_1, o_2, \dots, o_t \mid \lambda) \quad (2.5)$$

$\delta_{t+1}(i)$  can be found using induction as:

$$\delta_{t+1}(i) = b_j(o_{t+1}) \max_{1 \leq j \leq N} [\delta_t(j) a_{ij}] \quad (2.6)$$

To return the state sequence, the argument that maximizes Equation (2) for every  $t$  and every  $j$  is stored in a array  $\psi_t(j)$  [87]. It is important to point out that The Viterbi algorithm can be implemented directly as a computer algorithm. Moreover, the algorithm succeeds in splitting up a global optimization problem so that the optimum can be computed recursively: in each step we maximize over one variable only, rather than maximizing over all  $n$  variables simultaneously.

Hidden Markov models have been used now for decades in signal-processing applications, such as speech recognition, but the interest in models has been broaden to fields of all kind of recognition, bioinformatics, finance etc. [193].

With regards the first order Markov model, if the past and the present information of the process is known, the statistical behaviour of the future evolution of the process is determined



by the present state. Thus, the past and the future are conditionally independent (the system has no memory) [194]. Therefore, it is reasonable to ask can there be a model which can gather and somewhat keep information from the past. The answer lies within a higher-order Markov models, where the hidden process is a higher order Markov chain and it is dependent on previous states. This gives memory to the model and such a modeling is more appropriate for processes in which memory is evident and important, for example a stock market time series.

#### 4. $k$ - Nearest Neighbour ( $k$ -NN)

One of the most straightforward, fundamental and extensively used algorithms for classification (although it can be applied for regression as well) is the  $k$ - Nearest Neighbour algorithm. It was first introduced by Fix and Hodges in 1951 as non-parametric method for pattern classification and was later formally elaborated and defined by Cover and Hart in 1967 [195]. For almost half a century,  $k$ -NN has been explored and implemented in numerous problems related to pattern classification such as pattern recognition, ranking models, text categorization as well as object recognition and has been recognized as one of the top ten data mining techniques [196].  $k$ -NN needs no prior knowledge about data distribution and has been labeled as a “lazy learning” or “instance-based learning” algorithm [164]. The algorithm will use raw training instances to make decision and no learning of the model is needed, i. e. it will not construct a mapping function or an internal model- the computational outcome will be derived directly from training data set stored in the memory [197].

The basic operation of the  $k$ -NN is founded on the calculation of distances among the tested and the training data samples for identification of its nearest neighbours thus assigning the tested instance to a particular class of its nearest neighbour[198]. One of the  $k$ -NN advantages are simplicity of implementation, robustness to noisy training data and its ability to effectively process large training data [199]. In the following some formal definitions will be provided alongside with the mathematical concepts for  $k$ - Nearest Neighbour that employs distance as a method for classifying data.

**Definition 2.1.2.** Let  $S$  be a set. A **metric** on  $S$  is a function  $d : S \times S \rightarrow \mathbb{R}$  that satisfies the following properties :

- i) (Positivity) For all  $x, y \in S$ ,  $d(x, y) \geq 0$ ; equality holds if and only if  $x = y$ ;
- ii) (Symmetry) For all  $x, y \in S$ ,  $d(x, y) = d(y, x)$ ;
- iii) (Triangle inequality) For all  $x, y, z \in S$ ,  $d(x, y) \leq d(x, z) + d(z, y)$ .

The couple  $(S, d)$  is called a **metric space**, the elements of  $S$  are called **points** and the number  $d(x, y)$  is called the **distance** between  $x$  and  $y$ .

Commonly used metrics for  $k$ - Nearest Neighbour technique are Euclidean, Manhattan, Minkowsky and Chebyshev distances with Euclidean distance being is the most frequently applied [198]. Some of these are defined in the following.

**Definition 2.1.3.** Let  $x = (x_1, x_2, \dots, x_n)$  and  $y = (y_1, y_2, \dots, y_n)$  be points in the  $n$ -dimensional metric space  $(X, d)$ .

- The Minkowski distance, commonly know as the  $L_p$  norm is defined as:

$$L_p = \sqrt[p]{\sum_{i=1}^n |x_i - y_i|^p}. \quad (2.7)$$

- The Euclidean distance norm, also known as the  $L_2$  norm is defined as:

$$L_2 = \sqrt{\sum_{i=1}^n |x_i - y_i|^2}. \quad (2.8)$$

- The Manhattan distance or the City block distance, otherwise known as the  $L_1$  norm is the special case of Minkowski distance for  $p = 1$  and is given with formula:

$$L_1 = \sum_{i=1}^n |x_i - y_i|. \quad (2.9)$$

- The Chebyshev distance, also known as chessboard distance or maximum value distance is defined as:

$$D_{Chebyshev}(x, y) = \max_i |x_i - y_i|. \quad (2.10)$$

From the two above formulas it can be seen that the Euclidean distance is just a special case of the Minkowski distance for  $p = 2$ .

**Principle of operation:** Let  $k$  be a positive integer and let set  $C$  be a set of different classes  $C = \{C_1, \dots, C_l\}$ . For a given  $m$ -dimensional metric space  $(X, d)$  let  $\{(x_1, y_1), \dots, (x_n, y_n)\}$  be a set of points  $x_i \in X$  with their appropriate classes, i.e.  $y_i \in C, \forall i = 1, \dots, n$ , (or in regression case let  $y_i \in \mathbb{R}$ ). For a given query point  $x \in X$  the  $k$ -NN algorithm determines the  $k$  closests points to  $x$  with respect to metric  $d$ , permutes the values  $y_i$  and orders them with respect to the given metric obtaining an ordered set  $\{(x_{1'}, y_{1'}), \dots, (x_{k'}, y_{k'})\}$ . In the case of classification, the corresponding class  $y(x)$  of the point  $x$  will be determined by majority voting rule, where as in the regression case,  $y(x)$  is calculated using formula:

$$y(x) = \sum_{i'=1}^k \omega_{i'} y_{i'}, \quad (2.11)$$

where number  $\omega_{i'}$  represents weight of each point, commonly calculated as  $\omega_{i'} = \frac{1}{d(x, x_{i'})}$ .

The majority voting rule for classification problems can also be expressed using Formula 2.11, where all weights are set to be  $\frac{1}{k}$  and

$$y(x) = \begin{cases} 1, & \text{if } \sum_{i=1}^k y_i > \frac{1}{2} \\ 0, & \text{if } \sum_{i=1}^k y_i < \frac{1}{2} \\ \text{a tie otherwise.} \end{cases} \quad (2.12)$$

Figure 2.2 represents an example of  $k$ -Nearest Neighbour classification. The test sample (the green star within the two circles) needs to be classified as class 1 of purple squares or class 2 of red circles. For  $k = 1$  it is assigned to class 1 since there is only one square within that circle. For  $k = 3$  (outside the first circle) it is assigned to the second class since there are 2 red circles and only 1 purple square inside the inner circle. For the case of  $k = 5$  the green star will be classified belonging to the first class since there are 3 purple squares vs. 2 red circles outside the outer circle.

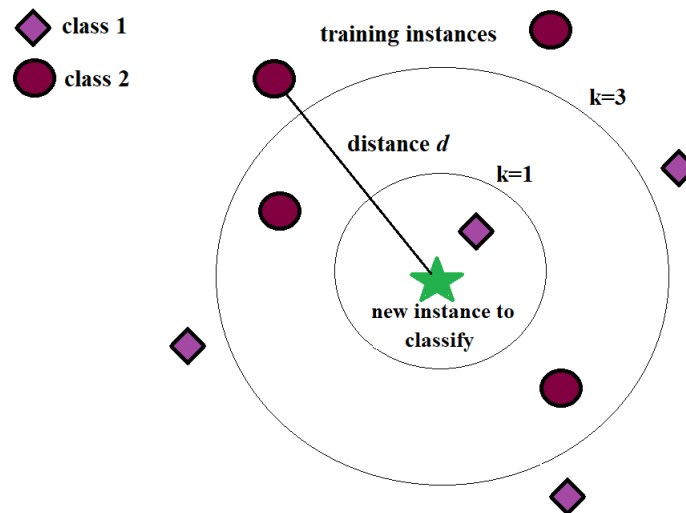


Figure 2.2. Example of  $k$ -Nearest Neighbour classification.

The main disadvantages of  $k$ -Nearest Neighbour algorithm are its dependence of the appropriate selection of distance (metric), high computational time since for every new instance, all the distances from  $k$ -neighbors need to be calculated all over [199], [196]. What is more, the  $k$ -NN inherits instability with respect to the order in which the data are presented to the algorithm which is not desirable and can be avoided by labeling the data, but at the cost of expanding the computational time [200]. What is more, if the class probability estimation is based on a simple voting, it can be a drawback if the nearest neighbors vary extensively in their distance and the closer neighbors more reliably indicate the class of the points, indicating that it may be prudent to give weight to each of the points based on their distance [201].

## 5. Neural Network

The core concept of DL has been displayed throughout an increasing research interest regarding Neural Networks or Artificial Neural Networks (ANN). They can be described as computational models and architectures that simulate the true functionality and structures of biological neural networks [202]. ANN have been found in a wide variety of areas that require classification or some form of prediction in many applications such as science, engineering, agriculture, mining, business, technology etc. [203]. Generally, the neural network is comprised out of an input layer, an output layer, and at least one hidden layer between the input and output with interconnections [202], as depicted in Figure 2.3.

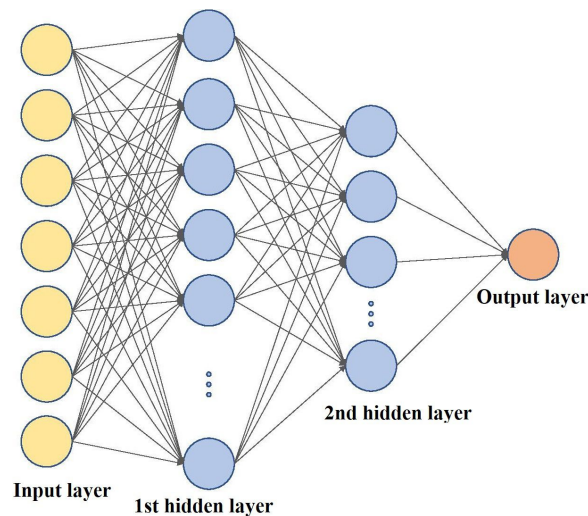


Figure 2.3. Example of a Neural Network Architecture

An important feature of a neural network is that it can hold more than one hidden layer, which actually represents the depth of the neural network. In one such a neural network, the simulation of the learning process is based on finding hidden connections among the sequences of input data through layers of neurons, in such a way that the output from a neuron in one layer represents the input to a neuron located in the next layer. An artificial neuron can be mathematically defined as non-linear mapping that is applied to a weighted sum of its input values and a bias, producing an output  $\hat{y}$  according to formula 2.13:

$$\hat{y} = \sigma \left( \sum_{i=1}^m w_i x_i + b \right); \quad (2.13)$$

$m$  represent the number of inputs,  $x_i$  represents the inputs, and  $w_i$  are the weights. Weights are assigned based on the inputs relative importance with regards to the other inputs, where as the bias provides a constant value to the mapping which can be crucial for a successful learning [88]. The non-linear mapping  $\sigma(\cdot)$  is called the activation (or transfer) function and it controls the output of the neuron by keeping it within acceptable values, usually between  $[0, 1]$  or between  $[-1, 1]$  [204]. Activation functions are characterized as linear and

non-linear, where non-linear ones are the most frequently used. Commonly utilized non-linear functions include (ReLU)  $\psi(x) = \max(0, x)$ , often used in recent years, as well as the more conventional Sigmoids function, like the hyperbolic tangent  $\Phi(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$  and logistic function  $S(x) = \frac{1}{1 + e^{-x}}$  [205]. ReLU is mainly used as an activation function in the hidden layer, while Sigmoid is usually employed in the output layer [206].

Since the training procedure of a Neural Network is an iterative process, the loss (cost) function is utilized to determine the quality of the network, aimed at giving weights to neurons during the training procedure. To minimize loss function during the training phase in which the weight of neurons is determined, a good deal of optimization algorithms have been implemented, many of which are first-order iterative optimization algorithms such as: Stochastic Gradient Descent (SGD), Adaptive Moment Optimization (Adam), and Root Mean Square Propagation (RMSProp). It is also important to determine the best learning rate in the model i.e., how much the model need to be changed in response to the estimated error each time the model weights are updated. With a high learning rate it may not enable find global minimum and the model might not converge at all, whereas with low learning rates model might take to long to converge [88].

## 6. Long-Short Term Memory (LSTM) neural network

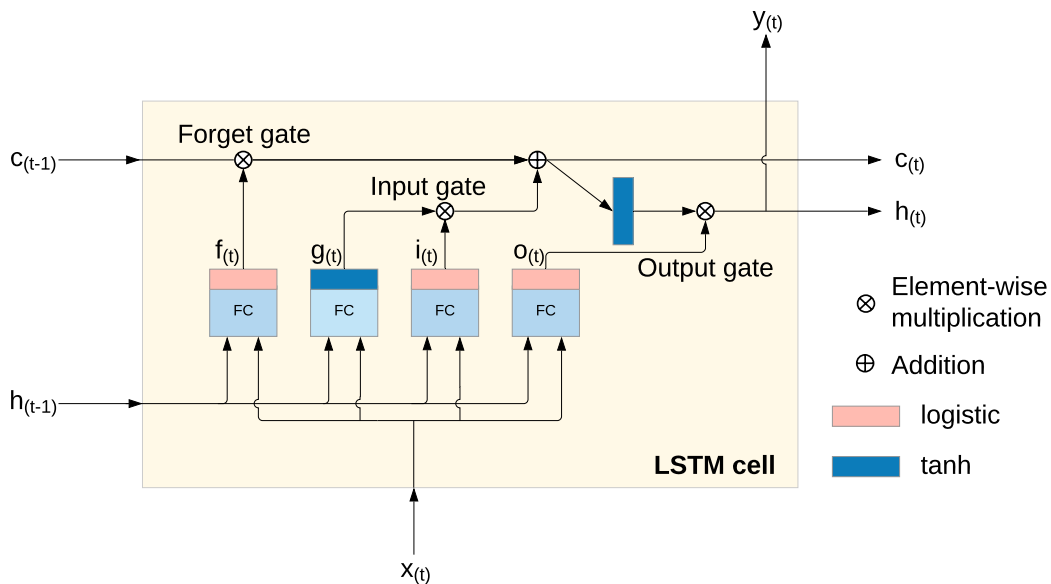


Figure 2.4. Long Short-Term Memory (LSTM) cell.

Recurrent Neural Networks are based on the recursive structure in which the one-step model with a time-step is trained first and then recursively used to return the multi-step prediction [207]. A special type of RNN is Long-Short Term Memory (LSTM) neural network constituted out of a set of recurrently connected memory blocks – LSTM cells (depicted in

Figure 2.4). LSTM cell consists out of four layers, main layer and three layers which are gate controllers each computing values between 0 and 1 based on their input [166].

Layers operate in the following way:

- Main layer - analyses the current inputs  $x_{(t)}$  and the previous (short-term) state  $h_{(t-1)}$  then outputs the  $g_{(t)}$  vector;
- Forget gate  $f_{(t)}$  decides parts of the long-term state  $c_{(t-1)}$  that need to be erased;
- Input gate  $i_{(t)}$  controls parts of  $g_{(t)}$  that are added to the long-term state  $c_{(t)}$ ;
- Output gate  $o_{(t)}$  determines which parts of long-term state should be read  $c_{(t-1)}$  and given to the output  $y_{(t)}$  and short-term state  $h_{(t)}$  at the current time step(t).

The states of the cell are calculated using equations given below:

$$i_{(t)} = \sigma(W_{xi}^T \cdot x_{(t)} + W_{hi}^T \cdot h_{(t-1)} + b_i) \quad (2.14)$$

$$f_{(t)} = \sigma(W_{xf}^T \cdot x_{(t)} + W_{hf}^T \cdot h_{(t-1)} + b_f) \quad (2.15)$$

$$o_{(t)} = \sigma(W_{xo}^T \cdot x_{(t)} + W_{ho}^T \cdot h_{(t-1)} + b_o) \quad (2.16)$$

$$g_{(t)} = \tanh(W_{xg}^T \cdot x_{(t)} + W_{hg}^T \cdot h_{(t-1)} + b_g) \quad (2.17)$$

$$c_{(t)} = f_{(t)} \otimes c_{(t-1)} + i_{(t)} \otimes g_{(t)} \quad (2.18)$$

$$y_{(t)} = h_{(t)} = o_{(t)} \otimes \tanh(c_{(t)}) \quad (2.19)$$

where  $\sigma$  represents logistic activation function,  $\tanh$  is hyperbolic tangent function,  $W_{(x)}$  are weight matrices for each of the four layers for input vector  $x_{(t)}$ , and  $W_{(h)}$  are matrices of the previous short-term state  $h_{(t-1)}$ . Finally,  $b$  denotes the bias term of each layer. Difference between the LSTM and the standard RNN is within their structure to memorize. With traditional RNN parts of information are lost in the process of each feedback resulting in RNN not being able to have long time memory in contrast to LSTM which has a long term memory. LSTM is able to remove or add information to the cell state, unlike the mechanism that completely overrides cell states like in standard RNN [208]. Long dependency in time can be observed in IoT applications such as environmental monitoring, human activity recognition, or machine translation and LSTM models have proven to perform better than RNN for such data [166]. LSTM cells are very successful at capturing long-term patterns in time series data and that was one the reasons for their selection as Deep Learning approach for prediction.

## 7. Convolutional Neural Networks

Convolutional Neural Networks have arisen as very effective Deep Learning models intended particularly for computer vision problems. These networks have shown remarkable perfor-

mance in a variety of visual identification tasks, such as picture classification, object detection, sentence classification, and image segmentation, making them an essential component of current computer vision systems [209]. Yann LeCun, Yoshua Bengio, and Geoffrey Hinton introduced CNNs in [210], laying the foundation for the application of CNNs in computer vision tasks. Convolutional Neural Networks are a specially designed for processing data with a known grid-like topology, such as time-series data or image data [211]. CNNs have demonstrated remarkable success in practical applications, and their name stems from the utilization of the mathematical operation known as convolution [162]. Convolution is a specialized linear operation, and in CNNs, it replaces the general matrix multiplication commonly used in traditional neural networks, being employed in at least one layer of the network architecture [212].

Convolutional Neural Networks exhibit a hierarchical architecture where the computation of each subsequent layer, denoted as  $x_j$ , builds upon the input signal  $x$  in the following manner:

$$x_j = \rho W_j x_{j-1}, \quad (2.20)$$

where  $W_j$  is a linear operator and  $\rho$  is a non-linearity [213]. This hierarchical structure enables the network to extract increasingly complex and abstract features from the input data. In the context of a CNN,  $W_j$  is commonly implemented as a convolution operation, while the activation function  $\rho$  is often chosen as either the rectifier function  $\max(x, 0)$  or the Sigmoid function  $1/(1 + \exp(-x))$  [209]. Conceptually, it is helpful to envision  $W_j$  as a collection of stacked convolutional filters. Consequently, the layers in a CNN can be regarded as maps of filters, with each layer expressed as the sum of convolutions performed on the preceding layer as follows [213]:

$$x_j(u, k_j) = \rho \left( \sum_k (x_{j-1}(\cdot, k) * W_{j, k_j}(\cdot, k)) (u) \right), \quad (2.21)$$

where,  $*$  is a discrete convolutional operator [162]:

$$(f * g)(x) = \sum_{u=-\infty}^{+\infty} f(u)g(x - u). \quad (2.22)$$

In general, The fundamental CNNs is composed of interconnected layers which include convolutional layers, pooling layers, and fully connected layers, which collectively enable CNNs to automatically learn and extract meaningful features from raw input data [209]. Convolutional layers employ filters that perform local receptive field operations, capturing spatial dependencies and detecting patterns at various scales [214]. Pooling layers, on the other hand, downsample the feature maps, reducing their spatial dimensions while retaining the most salient information [212]. Finally, fully connected layers combine the learned features and generate predictions based on them [209]. An example of Convolutional Neural Network

Architecture as given in [211] is presented in Figure 2.5.

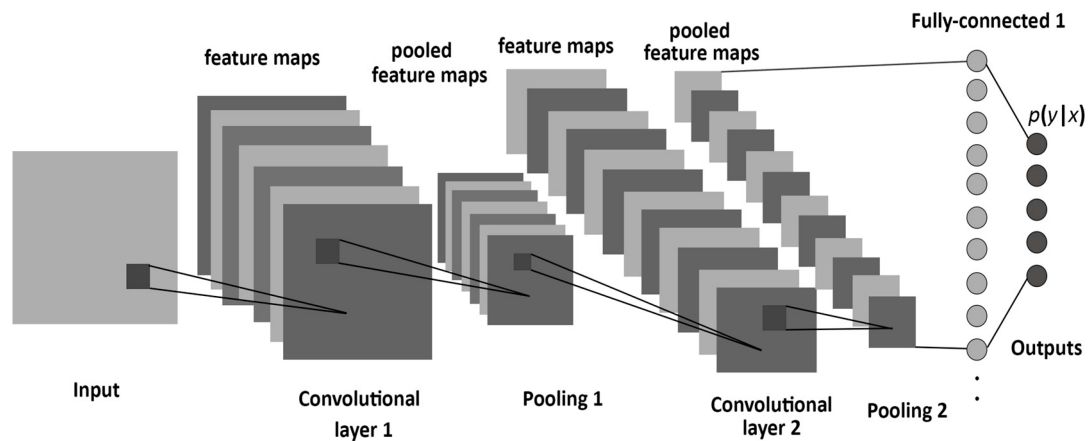


Figure 2.5. Example of a Convolutional Neural Network Architecture [211].

The field of CNNs has seen significant advancements through the introduction of influential architectures and techniques. LeNet-5, proposed by LeCun et al. [210], played a pivotal role in early CNN development, specifically for handwritten digit recognition. AlexNet, introduced by Krizhevsky et al. [?], marked a breakthrough by winning the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2012, showcasing the power of deep CNNs. VGGNet, proposed by Simonyan and Zisserman (2014) in [215], emphasized the importance of deeper networks by introducing a series of stacked convolutional layers. GoogLeNet, introduced by Szegedy et al. (2015) [216], proposed the inception module, enabling efficient network architectures with multiple paths. ResNet, presented by He et al. (2016) in [217], introduced residual connections to address the challenge of training very deep networks.

## 2.2 Perception Layer: Overview of Scientific Literature

### 2.2.1 Soil Humidity Sensing

Soil moisture plays a significant role in agricultural production and hydrological cycles, and accurate prediction is crucial for the effective utilization and management of water resources [218]. However, predicting soil moisture accurately is challenging due to its complex structural characteristics and the influence of meteorological factors [219]. Developing an ideal mathematical model for soil moisture prediction is a difficult task. Existing prediction models face issues such as limited accuracy, generalization capability, and the ability to process multiple features simultaneously, necessitating improvements in prediction performance [218]. The dynamics of soil moisture, encompassing both its growth and regression, exert a direct influence on the water consumption and growth of crops. It serves as a crucial indicator for various aspects of agricultural production, including drought resistance, flood control



[220, 221], and precision irrigation decisions [222, 223]. Accurate prediction of the temporal patterns of soil moisture regression is of paramount importance in effectively managing agricultural water resources and fostering improvements in crop yields [218]. Wide variety of battery-operated sensing devices already exist which are based on measuring the electrical properties of the soil, while data is delivered through some wireless interface. Existing wireless technologies have been either designed for high-throughput applications (e.g., 3G, WiFi, LTE) with high power consumption or are characterized by low power consumption (e.g., ZigBee, Bluetooth Low Energy) but limited in the achievable coverage area. Low power wide area networks (LPWA), such as LoRa, Sigfox NB-IoT are emerging as the enabling wireless technology especially for the development of precision farming, flood monitoring [224], precision livestock farming and/or smart irrigation systems [225, 226]. LPWA leverage the need for only intermittent or sporadic transmissions of small data packets, making them suitable for battery-operated devices. Existing commercial sensors for irrigation systems are quite expensive, while sensor lifetime can reach up to couple of years.

Application of machine learning techniques for agriculture use and specifically soil moisture estimation and prediction have interested researches for over two decades. One such research is presented in [227], where Soil moisture is estimated using remote sensing data from Tropical Rainfall Measuring Mission Precipitation Radar (TRMMPR). The aim of the study was the estimation of soil moisture content for the Lower Colorado River Basin area. The authors developed Support Vector Machine (SVM), Artificial Neural Networks (ANN) and multivariate linear regression (MLR) models for the estimation and showed that the SVM model is better in capturing interrelation between soil moisture, backscatter and vegetation in comparisons to the ANN and MLR models. Study presented in [228] investigated relationship between soil moisture, Precipitation-Evapotranspiration Index (SPEI) and climate indices in Xiangjiang River basin. Incorporating the climate impact on drought, the Support Vector Regression SVR model is built to predict the SPEI from climate indices. The results showed that the SVR model could improve prediction accuracy of drought in comparison to solely using the drought index as the only input parameter. In [229] authors used precipitation, daytime and nighttime land surface temperature, potential evapotranspiration (PET) estimated using mean temperature, the Normalized Difference Vegetation Index, the Normalized Difference Water Index data combined with large-scale climate indices and long-range forecast climatological data for drought forecasting in the area of South Korea. They developed Decision trees (DT), Random forest (RF) and Extremely randomized trees (ERT) models as both classification and regression models. Results show that the regression models gave better performance in majority of cases. Soil moisture estimation from meteorological data using SVR is given in [230]. The authors also compare the SVR model with ANN to validate capabilities of the SVR model and conclude that SVR outperforms ANN in all cases. A Deep Learning model for soil moisture estimation is presented in the study [231]. The authors use deep belief networks (DBN) to predict Soil moisture content from topo-

graphic properties, environmental and meteorological data such as evapotranspiration, leaf area index (LAI) and land surface temperature in the Zhangye oasis in Northwest China. A novel macroscopic cellular automata (MCA) model by combining DBN is given and compared with widely used neural network, multi-layer perceptron (MLP). The result show that the DBN-MCA model led to a reduction in root mean square error by 18% in comparison with the MLP model. The authors conclude that the MCA model is promising for modeling the temporal and spatial variations of Soil Moisture content.

Review of Machine Learning dedicated to applications of machine learning in agricultural production systems is presented in [232]. The authors emphasise key and unique features of popular ML models and conclude that the ML models will be even more widespread in the future providing production improvement. Generally, a variety of ML algorithms have been exploited for agricultural purposes, such as [233], where soil moisture, crop biomass and Leaf Area Index are estimated from X-band ground-based scatterometer measurements using two variants of Radial basis function neural networks (RBFNN) algorithms, namely conventional radial basis function neural network and generalized regression neural network (GRNN). Results show that good performance was obtained from both networks in retrieving soil moisture content. Furthermore, in [234] support vector regression (SVR) technique was used and compared with multi layer perceptron neural network (MLP NN) algorithm for soil moisture estimation using C-band scatterometer field measurements and considering various combinations of the input features (i.e., different active and/or passive microwave measurements acquired using various sensor frequencies, polarization, and acquisition geometries). The authors present a comparison of SVR model performance and the MLP NN model and conclude that the SVR provides higher accuracy in prediction for the given data sets and for all the input feature configurations. They imply that the SVR model has a better generalization ability than the MLP NN model, i.e., the SVR model is more capable to learn mapping that provides higher accuracy in the prediction of unknown real samples. Furthermore, in [235] Long Short-Term Memory (LSTM) has been applied to predict water depth in agricultural Hetao Irrigation District in arid northwestern China using monthly water diversion, evaporation, precipitation, temperature, and time as input data to predict water table depth. The model was evaluated using RMSE and coefficient of determination  $R^2$ . The authors conclude that the proposed model is suitable for predicting water table depth and especially can be used in areas with complex hydro-geological characteristics. In [236] neural networks, multiple regression, and fuzzy logic were used for spatial soil moisture retrieval using active microwave data. The study area was located in Oklahoma, USA and models sensitivity was estimated by measuring the change of RMSE when an input variable is added (or deleted) from the models. The obtained results suggest that soil texture and vegetation highly influence soil moisture retrieval. The authors conclude that the fuzzy logic and neural network models out-performed multiple regression in terms of validation. Table 2.1 gives a short comparison of above mentioned researches regarding ML models for soil moisture and

drought estimation and forecasting.

Soil moisture estimation from other measurements such as RSSI only recently started to attract the research community. In [237] authors present a case study of how variations in meteorological conditions with four selected meteorological factors, air temperature, absolute air humidity, precipitation and sunlight influence IEEE 802.15.4 network based on six months of sensor data. Amongst the obtained results, they conclude that temperature is the most dominantly correlated with RSSI. Similarly, in [238] an impact of both air temperature and air humidity on performances of signal strength variations of 802.15.4 networks is shown. The authors conclude that air temperature has a significant negative influence on signal strength in general, while high relative air humidity may effect the signal on lower temperatures.

*Table 2.1. Comparison table of Machine Learning models and applications for Soil moisture and drought estimation.*

Paper	Prediction model	Application	Best performance model
A. Sajjad et al. [227]	SVM, ANN, MLR	Soil moisture estimation	SVM
Y. Tian et al. [228]	SVR, drought index	Prediction of agricultural drought	SVR
J. Rhee and J. Im [229]	DT, RF, ERT (classification and regression models)	Drought forecasting	regression
M. Gill et al. [230]	SVM, ANN	Soil moisture prediction	SVM
X-D.Song et al. [231]	DBN, MLP	Soil moisture content prediction	DBN
R. Prasad et al. [233]	RBFNN, GRNN	Soil moisture estimation	both
L. Pasolli [234]	SVR, MLP NN	Soil moisture estimation	SVR
J. Zhang et al. [235]	LSTM	Water depth	LSTM
T. Lakhankar et al. [236]	neural networks, multiple regression, fuzzy logic	Spatial soil moisture retrieval	fuzzy logic, neural network

In [239], authors present a soil moisture monitoring system that uses UHF RFID tags in order to provide a wireless and battery-less field sensor. The paper presents the conceptual design of the system and provides experimental results showing that RSSI signal correlates with soil moisture using ANN. Further on, Artificial Neural Network was used for soil moisture prediction based on the RFID tag signal analyses giving coefficient of determination  $R^2 > 0.9$  in majority of cases. However, since buried UHF RFID tags can be only read from short distances (up to 50 cm), this work proposes a mobile robot that travels across the field and navigates above the buried UHF RFID tags to collect RSSI data. Such a solution is time consuming and challenging especially when a large number of RFID sensors is scattered over a large and possibly uneven crop field, which requires a robot to travel to every tag to collect RSSI data. In [240] authors propose a passive UHF RFID tags sensors integrated with a monopole probe for soil moisture monitoring. Their experimental results show that changes in soil permittivity cause changes in RSSI of the back-scattered signal. Therefore, we conclude that further exploration of Received Signal Strength in the context of soil moisture estimation is required, especially for the purpose of reducing size, cost and battery efficiency of sensor device. With regards to LoRa based systems, research community has recently began to research the potential of correlating RSSI changes in LoRa signal with specific environmental changes. In study presented in [241] a publicly available

dataset of LoRaWAN RSSI measurements is utilized to compare different machine learning methods for fingerprinting localization. The authors present the k-Nearest Neighbours (kNN) method, the Extra Trees method and a neural network approach using a Multi-layer Perceptron. They conclude that the MLP performs best achieving highest accuracy. In [242], the authors present results of signal strength measurements and simulations based on Wireless InSite radio propagation software and imply that parking occupancy can be estimated by detecting the change in RSSI at the receiver side. With regards to soil moisture and its correlation to RSSI researches, [243, 244] show existence of correlation between RSSI signals from LoRa-based devices and soil moisture, for sensors and gateways buried fully into the ground. In [244] authors present design and experimental validation of the developed Soil Moisture Sensing System (SoMoS) based on a Software Defined Radio (SDR) approach using LoRa in the laboratory. The system showed valid behavior and is able to detect soil moisture via the radio field. Furthermore, authors in [243] have done a long term evaluation of the previously proposed SoMoS system showing a high correlation between measured Receive Signal Strength Indicator and precipitation events. Table 2.2 compares the above mentioned papers and this paper in terms of used radio technology for variety of applications.

Table 2.2. Comparison table of various radio technologies and applications based on signal strength variations.

Paper	Radio technology	Application	ML model	Best performance model	RSSI values mapped with specific soil moisture values
Aroca et al. [239]	UHF RFID (short range)	soil moisture prediction	ANN	ANN	YES (MSE=0.00152)
Hasan et al. [240]	UHF RFID (short range)	soil permittivity	none	none	NO
Anagnostopoulos et al. [241]	LoRa (long range)	localization	kNN, Extra Trees, MLP	MLP	/
Solic et al. [242]	LoRa (long range)	parking space occupancy	none	none	/
Wennerström et al. [237]	802.15.4 (short range)	change of meteorological factors	none	none	NO
Luomala et al. [238]	802.15.4 (short range)	air temperature and air humidity	none	none	NO
Liedmann et al. [244]	LoRa (long range)	soil moisture estimation	none	none	NO
Liedmann et al. [243]	LoRa (long range)	soil moisture estimation	none	none	NO

In the context of soil humidity estimation, employing LoRa technology in combination with Machine Learning techniques offers several advantages. LoRa's long-range capabilities enable devices to communicate over extended distances, reaching up to 10 km. This surpasses the communication range of other technologies such as UHF RFID and 802.15.4 radio technology. The extended communication range of LoRa allows for the deployment of underground beacons distributed across large crop fields, while utilizing a single overground gateway device for data collection. This setup could enable simultaneous gathering of signal strength measurements from multiple beacons, providing a comprehensive view of soil conditions across the entire field. By harnessing Machine Learning techniques, the collected signal strength data can be analyzed to extract meaningful insights. Through the training of Machine Learning models, we can establish relationships between the signal strength measurements and corresponding soil humidity levels. These models can then be utilized to predict soil humidity based on future signal strength measurements.

### 2.2.2 Smart Parking Solutions

In the age of the Internet of Things and Smart City ecosystems, Smart Parking solutions have gained prominence due to their ability to optimize time, reduce fuel consumption, and lower carbon emissions [245]. With a defined architecture comprising sensors, communication protocols, and software solutions, smart parking addresses the challenge of parking space and management in congested cities. The increasing number of vehicles and limited parking spaces result in congestion, driver frustration, and environmental pollution. For instance, in New York, drivers spend an average of 107 hours per year searching for parking spots, leading to increased emissions and fuel consumption [13]. Addressing these challenges through effective and sustainable smart parking solutions is a significant endeavor.

Diverse research efforts have been done that focus on enhancing parking space detection and predicting future occupancy for optimal utilization. It particularly solutions extensively investigate and test the three key components of Smart Parking Solutions: sensor types, communication protocols, and the utilization of Machine Learning techniques.

An IoT based smart parking system presented in research [246] exhibits a solution that provides parking lot occupancy information, parking slot reservation and payments using a mobile app. The solution is based on the Passive Infrared and Ultrasonic Sensors to sense parking slot availability where information is send using Wi-Fi and a raspberry pi acts as an intermediate between the sensors and cloud allowing communication with the cloud to process collected data. Finally, the mobile application serves as an interaction interface between the end user and the system.

In [247], the authors designed a prototype of a parking occupancy monitoring and visualization system that uses an ultrasonic sensor being controlled by an Arduino Uno which uses a Wireless XBee shield and an XBee Series 2 module for communication. The data collected from the sensor is then given as an input to a algorithm that detects parking space statues and reports to a database in a real-time basis.

Research presented in [248] describes a system based on ultrasonic sensors for detection of parking spaces. Information about occupancy is then sent via Zigbee protocol to an information center, where as a Bluetooth station is employed to identify a user within the parking lot. By employing a proposed “Shortest path search algorithm”, the user will be able to find the swiftest ways to a free parking space. All the collected data is onwards sent via Wi-Fi to a parking a management menu.

In [249], the authors used a light detection and ranging optical sensor in order to measure the distance between a car and an object next to it. They have combined this sensor with a GPS receiver to determine the speed of a vehicle in a particular pair of geographic coordinates and a web camera to track tests. The information were then sent to a Raspberry Pi connected to the cloud via LTE-IEEE 802.11p protocol for further data processing and analyses. Parking situations were estimated by applying machine learning (not explicit which

one) obtaining accuracy above 95%.

Research that was conducted in [250] uses video camera sensors for detecting multiple parking space occupancy. Using image processing techniques: the Histogram of oriented Gradient (HOG) descriptor, the Scale-invariant feature transform (SIFT) corner detector, and Metrics on Color Spaces YUV, HSV, and YCrCb authors achieved an accuracy rate of over 93% for parking lot occupancy detection.

Employment of RFID sensors in a smart parking scenario has been explored in [251]. The solution is based on RFID readers, passive RFID tags, barriers, retractable bollards, Wi-Fi spots and a database. Readers are placed at the entrance and exit of the parking area allowing entrance to cars that have RFID tags, where the parking space is assigned with the same identification number as the passive RFID tag. This way, information about parking occupation is collected at the entrance along with the time of occupancy as well as exit time. Gathered data is sent through Wi-Fi to a cloud server that saves it in a database for future study. The main novelty of the proposed solution is the adoption of RFID tag-to-tag communication, which can support a more energy-efficient collection of information from the RFID tags compared to the conventional direct type of communication. Research presented in [92] demonstrates a proof-of-concept of a smart parking solution based on Ultra-High Frequency (UHF) RFID and WSN technologies. The infrastructure of the system consists of Zigbee network, Smart Gateway (SG), Central Server (CS) and two different mobile applications, named Parking App and Policeman App, designed for vehicle drivers and traffic cops, respectively. The information about occupancy is collected using RFID tags, where CS receives the information about occupancy if an appropriate RFID tag has been read by the reader that is placed on poles located near the reserved parking spaces. The system onwards directs the drivers to the nearest empty parking space by using a customized software application.

A novel vehicle detection sensor design based on a dual microwave Doppler radar transceiver modules is presented in [252]. By employing a motion recognition algorithm for vehicle behavior identification and parking occupancy detection, the proposed sensor is able to detect the vehicle movement clearly with the parking space occupancy detection accuracy higher than 98%. Research elaborated in [253] provided a radar based real-time algorithm, which detects, classifies, and evaluates parking spaces in a vehicle's immediate vicinity. Their approach processed data obtained from radar in a form of a particularly designed target list of 2D vectors. Using this method, computation burden was decreased and quantization errors were evaded. Experimental results show that more than 95% of all parking spaces were classified correctly in several test drives, indicating that the proposed algorithm is suited for both parallel and perpendicular parking spaces in urban scenarios.

Recently, an extensive research elaborated in [254] presented an IoT-driven vehicle detection method by combining the data feature of magnetic signals with that of Ultra-wideband (UWB) radio channels aiming to improve wireless vehicle detectors that are based on

the IoT technology and magnetic sensor. The proposed method obtains vehicle detection by examining the length of the propagation path and signature of the channel impulse response with respect to the vehicles, which can be obtained from UWB modules. The experimental results indicate that the sensor can achieve a detection accuracy of 98.81% when the sampling rate of the magnetic sensor is 1 Hz. Research presented in [255] provides a comparison between inductive loop and magnetic sensors for vehicle detection that can be employed for traffic or parking systems. The overall measurements provided good comparison results between the two technologies and when considering all the values gathered, the inductive loop detector reported a total of 13,713 vehicles, while the magnetic sensor reported 13,407 vehicles, resulting in an overall detection difference of 306 vehicles (2.28%). Furthermore, a street parking system proposed in [256], employs a magnetic sensor node on the space to monitor the state of every parking space. Authors of the study propose a vehicle detection algorithm based on the magnetic signal along with an adaptive sampling mechanism to reduce energy consumption. Evaluation of the system was performed on a street parking spaces where eighty-two sensor nodes were implemented and collected the data for over a year. Their results indicate a vehicle detection accuracy of over 98% and the lifetime of the sensor node of more than 5 years with a pair of 2500mAh Li batteries.

Implementation of an acoustic sensors parking space surveillance system is exhibited in [257]. Authors elaborate on the design, implementation, and evaluation of the system equipped with low-cost microphones that are able to localize acoustic events. The system is constituted out of the Acoustic Source Localization (ASL) system, the surveillance camera system, and the server system. Once an acoustic event occurs, the ASL sends estimated position of the acoustic event to the server, which then displays the estimated position on the map and sounds an alarm. The camera surveillance system then rearranges the cameras pointing to the estimated position to capture the event scene and onward performs motion detection to locate a more precise position. The data is transmitted over 802.15.4 wireless network protocol. The proposed system can efficiently supervise a large parking lot of 100 vehicles using only a dozen sensor nodes. Feasibility of the system was validated in a real parking lot, where experimental results show that detection rate in the region with the alarm using a camera is 94.29%.

Research presented in [258] proposes a smart parking solution based on the advantages of NB-IoT technology. The system is comprised of the sensor node made geomagnetic vehicle detectors, smart parking cloud server, application for mobile device and the third-party payment platform. The cloud server implements basic information management, charge management, sensor node surveillance, task management and business intelligence modules. The proposed system has already been deployed two cities in Zhejiang province, China. The author however, do not discuss the accuracy of detection of the proposed system.

Rather recently, authors in [259] presented a smart parking system based on ultrasonic sensors and the received signal strength indicator (RSSI) in Bluetooth communication. Park-

ing occupancy is determined by using sensor nodes located in the parking spaces whereas the parked vehicle location is based on the BLE communication between the user's smartphone and the sensor nodes. What is more, the authors have done RSSI transformation into a distance range using triangulation in order to improve the location awareness for users. System was further evaluated by employing the sensor nodes for the ultrasonic sensor and the BLE modules in parking spaces. Experimental results confirmed that the ultrasonic sensors successfully detected the available parking spaces with the 83% accuracy. A similar idea was explored in work presented in [260]. Authors present a smart parking solution for both indoor and outdoor parking areas that is based Bluetooth low energy beacons and which uses particle filtering to improve its accuracy with the goal to develop a smartphone application for parking users enable them to securely and easily find and pay for parking, while also providing management capabilities for the parking facility owners. The system builds an RSSI path loss model for the desired parking region and further implements the RSSI-based distance estimation on the smartphone. Based on the experimental results, the solution has achieved accuracy ranging from 87% to 100% for outdoor and 74% to 100% accuracy in indoor parking availability estimation.

Employment of LoRa technology for smart parking solution was examined in research [202]. Authors propose a smart vehicle parking system architecture is made out of four layers: sensor nodes, edge gateway, LoRa gateway and Cloud and provide a proof-of-concept with a specific realization. Employed sensors were accelerometer, magnetometer and temperature, humidity and barometric pressure sensor for weather conditions. Sensor data is sent over to the edge layer before transmitting it to the cloud layer, and consequently to the end users. The LoRa gateway layer provided communication to ensure robust connection between the edge and cloud layers. The authors examined the latest LoRa and nRF communication technology to effectively increase the energy-efficiency and coverage area. They also propose a dynamic pricing algorithm for maximizing profit for the parking management authorities. Although the presented system is energy-efficient, secure, and provides a multi-parametric data about the parking slots the authors do not discuss the accuracy of detection.

In the last decade, a number of solutions aiming at predicting the occupancy in the future have emerged with the goal of simplifying the search of free parking spaces. These solutions are based on Machine Learning techniques that involve learning, predicting, and the exploiting of cloud based architectures for data storage [261]. Generally, data regarding occupancy are the history of occupancy for a parking lot, containing date-time information with a specific occupancy status. For instance, in the work [262], while using ML, the authors present two smart car parking scenarios based on real-time car parking information that has been collected from sensors in the City of San Francisco, USA, and the City of Melbourne, Australia. The historic data contained features, like area name, street name, side of street, street marker, arrival time, departure time, duration of parking events (in seconds), sign, in



violation, street ID, and device ID. From these data, the occupancy rate was calculated. The evaluation revealed that the Regression Tree, when compared to NN and SVR, using a feature set that includes the history of the occupancy rates along with the time and the day of the week performed best for prediction of a free parking space on both the data sets. Moreover, in research [261], the authors applied a Recurrent Neural Network-based approach for the prediction of the number of free parking spaces. They have used parking data of Birmingham, U.K., which contained the parking occupancy rate for each parking area given the time and date. They achieved the median of mean absolute error of 0.077 for prediction of occupancy. The results show that the approach used is accurate to the point of being useful for being utilized in Smart Parking solutions. In [263], the authors discuss the problem of predicting the number of available parking spaces in a parking lot by regarding the vehicle's arrival as a Poisson distribution process. They model the parking lot as a continuous-time Markov chain. With the predicted occupancy status, each parking lot can provide availability information to the drivers via vehicular networks.

Research presented in [264] proposes an urban smart parking management platform based on the NB-IoT and wireless sensor network. The presented solution employs automatic license plate recognition (LPR) device to obtain images from the video stream and determine the license plate information whereas the vehicle location within the parking space is acquired by geomagnetic sensors. The employed image recognition algorithm for LPR is the Back Propagation (BP) Neural Network Algorithm. In the overall architecture of the system a personal digital adaptor (PDA)/mobile app acts as the management tool, and the NB-IoT wireless communication is used for data transmission. The authors elaborate and compare power consumption of NB-IoT with that of Zigbee to verify the performance of the proposed platform, concluding that the NB-IoT consumed less energy than Zigbee, indicating that the technique is cheaper to maintain, considering the long-term maintenance cost. The authors do not comment or discuss the system performance in terms of accuracy of detection of a free parking space.

A recent study exhibited in [265] attempted to realize a low-cost smart parking system utilizing several BLE beacon devices, a smartphone owned by a pedestrian/driver, a gateway, and a server. The idea is that a pedestrian's smartphone measures the received signal strength when it receives radio waves transmitted from the beacon device, and then estimates its own position in the parking lot by inputting the time series data to a learning model of the machine learning based on deep learning. Once the server gets the status of each slot in the parking space from the gateway, it would provide a driver outside the parking lot information about parking availability. What is more, the server has a function of constructing a new learning model based on the measurement results of the smartphones, and applying the updated learning model to the smartphones. The authors utilized Deep Neural Networks and Convolutional Neural Network (as deep learning approaches for parking occupancy estimation). Experimental results show that DNN obtained 98% accuracy in parking slot estimation

in contrast to CCN that had 99% estimation accuracy. What is more, estimation accuracy of the pedestrian's position is around 70% and the system is able to position of vehicles / pedestrians and to send the estimation results in the parking lot in less than 0.1 seconds. A similar idea of employment BLE beacon devices (a mesh network topology) and localization technique based on radio fingerprinting was presented in research [266]. Author propose a smart parking solution in which nodes listen for broadcasts of RSSI values from a custom beacon placed in every vehicle that parks in the lot. The RSSI values are then validated, encrypted, and sent back to a designated central node where space prediction occurs using ML; namely Decision Tree, Random Forest, Naive Bayes, Support Vector Machines and  $k$ -Nearest Neighbors were employed. Experimental results indicate prediction model obtains a high accuracy using radio training data (90.7% correctly identified) where the evaluation shows a promising result of 69.17% accuracy up to and including 3 spaces away), even without employing tuning and data filtering techniques for the RF classifier.

Two possible solutions for a smart parking deployment are presented in a research exhibited in [267]. Authors propose a design of an adaptable and affordable smart parking system using distributed cameras, LIDAR sensors edge computing, data analyses, and utilization of advanced deep learning algorithms. One solution would use a network of cameras as a sensing technique and the other network of LIDAR sensors, where the data is sent for further processing via Wi-Fi mesh technology. Both solution utilize three types of Neural Networks, namely Standard AlexNet, AlexNet with two convolution and a custom designed network model with one convolution layer. Their results show that camera model obtained 99.8584 % accuracy of detection of empty parking spots, whereas for the LIDAR model result vary depending of the spot from 30% to up to 93% of accuracy.

Moreover, in [268], the authors presented a novel system for detecting the cruising behavior in vehicle journeys and developed a real-time parking information system. The system uses GPS sensors as an application that sends the user's location and allows for the system to create a heat map with the acquired information showing free and unavailable parking lots. The proposed method relies on the principle of detecting a significant local minimum in the GPS trace with respect to the distance from the destination. In addition to GPS data, other sensing data from the driver's smartphone, such as accelerometer, gyroscope, and magnetometer, were also collected. Classification using Decision Trees, Support Vector Machines and  $k$ -Nearest Neighbors is used to detect cruising behavior. The system then automatically annotates parking availability on road segments based on the classified data and displays this information as a heat-map of parking availability information on the user's smartphone. Using this approach, the researches were able to detect cruising on average 81% of the time.

The work presented in [269] investigates the changing characteristics of short-term available parking spaces. The availability data were collected from parking in several off-street parking garages in Newcastle. This forecasting model is based on the Wavelet Neural Network (WNN) method and it is compared with the largest Lyapunov exponents (LEs) method

in the aspects of accuracy, efficiency, and robustness. They conclude that WNN gives a more accurate short-term forecasting prediction with a average mean square error (MSE) is  $6.4 \pm 3.1$ .

More recently, the authors in [270] presented a framework that is based on LSTM in order to predict the availability of parking space with the integration of Internet of Things. They have also used the previously mentioned Birmingham parking sensors data set for performance evaluation of free parking space prediction that is based on location, days of a week, and working hours of a day. The authors show that, from all performance measurement parameters, the minimum prediction accuracy is 93.2% and maximum prediction accuracy is 99.8%. They present the experimental results that show that their proposed model outperforms the state-of-the-art prediction models. Finally, they point to some limitations of the study regarding the decision support system: it predicts the availability of parking lots only considering the parking occupancy information.

Recent works of researches incorporated Markov models for parking space occupancy predictions. For instance, in [271], the authors propose a model-based framework in order to predict future occupancy from historical occupancy data. The foundation of this predictive framework is continuous-time Markov queuing model, which is employed to describe the stochastic occupancy change of a parking facility. The model was evaluated while using a mean absolute relative error (MARE), ranging from 5.23% to 1.86% for different case studies. Furthermore, in [272], an agent-based service combined with a learning and prediction system that uses a time varying Markov chain to predict parking availability is proposed. Agents predict the parking availability in a given parking garage and communicate with other agents in order to produce a cumulative prediction achieving prediction accuracy of about 83%.

Neural Networks have also been used for in prediction of future occupation of parking space such as in [273, 274]. Researches in [273] have exploited the data concerning the availability of a free parking state depending on the duration of a particular occupancy status. Therefore, they have deployed a long term and short term occupancy prediction system based on neural network that achieves good performance with only a 0.004 Mean Absolute Error. They concluded that temporal changes of parking occupancy status was appropriately encompassed by the NN model that can provide an rather precise occupancy prediction up to thirty minutes ahead.

Authors in [274] have utilized a DL neural network for classification of a free parking space. Their model is based on images of a parking lot and it achieves a exceptionally good classification with 93% accurately classified occupancy status for a particular data set. Work presented in [275] proposed an occupancy prediction model using a deep neural network model which includes various data sources such as weather conditions, traffic conditions as well as parking meter transactions. Using Graph-Convolutional Neural Networks (GCNN) model is able to extract the spatial relations of traffic flow in large-scale networks and further

captures the temporal features by applying Recurrent Neural Networks along with Long-Short Term Memory. Evaluation of the model's performance was done on a case study for the downtown area in Pittsburgh and it achieved mean absolute percentage error (MAPE) of 10.6% when predicting block-level parking occupancy half an hour in advance.

Researches in [269] have explored the altering properties of parking spaces that available for a short-term period. Data about occupancy in a particular period have been collected in a couple of Newcastle off-street parking garages. A model has been designed based on Wavelet Neural Networks (WNN) and it provides a short-term predictions of occupancy. The proposed model was evaluated in the terms of efficiency, accuracy and robustness and compared with the largest Lyapunov exponents (LEs) method. They conclude that WNN gives a more accurate prediction achieving an average mean square error of  $6.4 \pm 3.1$ . More recently, authors in [276] have explored the use of deep convolutional neural networks, namely ResNet, based on the two different data sets containing parking lot images. They have been able to obtain an high accuracy rate ranging from 97,36% up to 99,82% for the test set and have optimized the increase of the learning error that occurs when the network becomes deeper thus providing swifter training. Research presented in [277] depicts a parking space occupancy monitoring software solution based on video and image processing and interpretation methods. Authors have employed five different models for classification, namely Logistic Regression, Radial Basis Function Support Vector Machine, Linear Support Vector Machine, Decision Tree and Random Forest. Based on classifier comparison Logistic regression achieved highest classification score of 93.5% out-performing other classifiers.

Authors in [278] have carried out a study that utilizes several future parking occupancy prediction models such as Multi-Layer Perceptron,  $k$ -Nearest Neighbour, Random Forest, Linear Regression and KStar (instance-based model) for the campus location Charles Sturt University (CSU), Australia. The algorithms are based on car park occupancy data collected for a period of five weeks. They have done a performance comparison for all of the algorithms based on the the simple mean as criterion of good performance. Authors conclude that although majority of algorithms provide rather precise prediction in stable conditions, for highly variable conditions the KStar has achieved the best results.

Work presented in [279] explored a concept of using the smartphone's sensors readings such as sound, pressure levels and luminosity to obtain the information about the users transportation mode. By using the pervasive Wi-Fi and cellular infrastructure they were able to automatically detect users which are going out from a parking spot. Researches have utilized the Random Forrest algorithm to classify sensor readings, in real time, and determine which form of the most frequent transportation modes used in city areas the user is applying (for example walking, bus driving, car riding, cycling, train riding etc.) Evaluation was carried out on 7 smartphones and 3 different cities showing an accuracy of over 95% in transportation mode classification and in return-to-vehicle scenarios.

Rather recently, researchers in [280] have proposed a parking space detection system that

uses parking lot images captured under different weather conditions as input and detects the empty slots in a particular images. They have employed combination of canny edge detection as well as LUV based colour variation detection methods to accurately derive the edges for each parking slot. Over a 942 images showing 37,680 parking spaces were used and Random Forest classifier has been utilized achieving accuracy of 98.31% compared to the existing methods. Authors point out to RF's good ability to solve the over-fitting problem with regards to training data and conclude this to be the reason of its accuracy. Not long ago, work presented in [281] provided a comparative analysis of Multilayer Perceptron,  $k$ -NN, Random Forest, Decision Tree, and Voting Classifier for the prediction of parking space availability. Data set used for the analysis was obtained by collecting the measurements of sensors deployed in city of Santander, Spain and it contained information about parking spot ID, day of the week, parking duration and status. Algorithms were evaluated in terms using K-fold cross-validation and numerical results obtained for Accuracy, Precision, Recall and F1-score. Authors conclude that the simpler algorithms such as DT, KNN and RF outperform more complex algorithms like Multilayer Perceptron, achieving higher prediction accuracy, giving better information about the prediction of parking space occupancy that can be compared.

The literature reviewed reveals important insights into current smart parking solutions. Many studies focus on implementing sensor and communication technologies for obtaining and sharing information about parking space occupancy. Some researches achieve high detection rates without employing machine learning techniques, while others utilize machine learning, particularly Neural Networks, for accurate prediction and classification of parking availability. Additionally, certain studies emphasize the performance of sensor technology alone, while others concentrate on historical data analysis for predicting parking occupancy. Overall, the literature highlights the significance of Neural Networks in achieving accurate results in parking prediction and classification.

Amongst 41 overall examined researches for Smart Parking solutions, only 8 exhibited all three components of the technological architecture for Smart Parking solutions examined within this paper. Table 2.3 gives a comparison of identified researches regarding the technological architecture of these existing Smart Parking solutions and the concept that is presented in this paper. Within this comparison only researches that have incorporated all three components of the technological architecture for Smart Parking solutions have been considered since only such solutions can be considered equivalent for comparison.

Table 2.3. Comparison table of various sensing technologies and its applications in Smart Parking.

Paper	Sensing Device (Network Protocol)	Data Type	Application	ML Model	Detection Rate
Ebuchi and Yamamoto[265]	BLE beacon and smartphone(BLE)	RSSI	parking occupancy detection	DNN, CNN	98%, 99%
Seymer et al.[266]	BLE beacon	RSSI	parking occupancy detection	DT,RF, Naive Bayes, SVM, $k$ -NN	69.17%-90.7%
Bura et al.[267]	camera, LIDAR (Wi-Fi mesh)	images, distance	parking occupancy detection	NN	30%-99.8%
Vlahogianni et al. [273]	ferromagnetic parking sensor (802.15.4)	occupancy history	parking occupancy prediction	NN	0.004 MAE
Farag et al. [274]	camera	parking spaces images	parking occupancy classification	NN	93% classification rate
Jones et al. [268]	GPS sensors	location data	detection of cruising behaviour	DT, SVM, $k$ -NN	81% detection accuracy
Hiesmair et al. [249]	LIDAR(LTE-IEEE 802.11p), GPS	distance, speed	estimation of parking situation	NN, DT, $k$ -NN, SVM	95% accuracy
Krieg et al.[279]	smartphone sensor (Wi-Fi)	sound, pressure levels and luminosity	users transportation mode	RF	95%

As can be seen from the table, there is an overwhelming dominance of short range technologies which will be further discussed. Data type used for building the Machine Learning models vary depending on the sensor technologies, which indicates that on a base level, the sensing technology greatly influences the choice of an appropriate Machine Learning model that is further utilized. What is more, the researches report on extremely high accuracy of detection/ prediction of availability of finding a free parking space when ML algorithms are adequately applied. It can also be noticed that traditional classifiers like  $k$ - NN or RF are able to compete with deep learning approaches like DNN or CNN.

Second major observation obtained from research of literature is the immense dominance of short-range communication technologies. Only researches presented in [258, 202, 264] have reported to have examined the long-range technologies, which is only 7% of all papers considered in this research. This trend has already been confirmed in a rather recent review of literature presented in [245], that has elaborated that only 10% of researched papers employed long-range communication technologies. What is more, amongst these none have employed Sigfox technology within their research. This is rather unusual, since it has been reported by [282, 283] that Libelium<sup>1</sup>, a WSN platform provider, has used both LoRaWAN and Sigfox in their Plug & Sense platform, which uses magnetic sensors to detect vehicles

<sup>1</sup>Libelium: <https://www.libelium.com/iot-products/smart-parking/>, (accessed on 8 October 2021)

in parking spots for commercial purposes. Author in [283] claim that Lora has not yet been popular within smart parking solutions, whereas SigFox has been highly commercially employed in large cities like Moscow and Barcelona, but not explored for scientific purposes.

Thirdly, only some studies have elaborated the cost and energy efficiency of the proposed solutions in general, like the ones presented in [256, 264]. This a rather small percent of research, and such aspects should be examined more. An appropriate solution must consider all technological aspects of a system in terms of accuracy, cost and energy savings. One such good approach was presented in [284], in which the authors have discussed state-of-the-art by a systematic in-depth overview of technologies used for the smart parking detection realization consuming mW of power. The researchers have conducted a real-scenario performances and power consumption of most popular sensor devices and LoRa, Sigfox and NB-IoT communication technology. Based on their results, lowest consumption is for LoRa devices. They have also conducted an analysis of power consumption of commercial LPWA-based Smart parking sensor device along with battery estimation lifetime.

To conclude, there is a need for comprehensive analyses of parking occupancy sensor accuracy and reliability through benchmarking experiments. The choice of sensor technologies significantly impacts communication and machine learning deployment in smart parking solutions. Further research is required to explore the performance and cost-effectiveness of long-range communication technologies. Moreover, Machine Learning implementation should focus on achieving high detection accuracy while reducing the cost and power consumption associated with multiple sensors and communication peripherals. Alternative solutions based on machine learning can offer cost-effective and accurate detection without the need for complex hardware configurations.

### **2.3 Network Layer: Overview of Scientific Literature**

The use of RFID technology drives numerous applications in the Internet of Things. However, the high production cost and short service lifetime due to periodic battery replacement pose challenges. To address these issues, it is crucial to focus on energy efficiency in RFID systems [285, 286, 287]. By accurately tracking information such as expiry dates and item leakage, RFID technology contributes to waste reduction and energy conservation in various operations, including monitoring, packaging, and refrigeration. This, in turn, facilitates the wider deployment of RFID systems [288]. As the integration of RFID into IoT systems progresses, the traditional fixed reader format is no longer the sole option. Mobile readers and battery-powered wireless sensor nodes are becoming viable reader devices. Therefore, energy efficiency becomes a vital metric for evaluating the overall performance of RFID systems [285, 286, 287]. An energy-efficient RFID protocol plays a significant role in prolonging the operating lifetimes of readers and, if applicable, active tags, thereby supporting the growth of environmentally friendly RFID technology and its envisioned applications.

Achieving energy efficiency requires the adoption of an energy-efficient anti-collision algorithm by the reader, which optimizes tag cardinality estimation, adaptively modulates transmission power levels, and reduces tag collision and eavesdropping [287].

Over the past few years, various methods and approaches have been employed for tag estimation. In [289], Vogt presented a method based on Minimum Squared Error (MSE) estimation by minimizing the Euclidean norm of the vector difference between the actual frame statistics and their expected values. The number of empty, successful, and collision slots is taken into account. However, the predicted values are calculated under the assumption that the tags in each slot have independent binomial distributions, which leads to unreliable findings. In the research presented by Chen in [102] the authors apply probabilistic modeling of the tag distribution within the frame, which they consider to be a multinomial distribution. By doing so, they obtain the tag number estimation. For each slot, binomial distributions provide occupancy information, however, it does not consider the fact that the number of tags in the interrogation area is limited [290]. An improvement of the previous model was given by research in [291], although this improvement tends to have a high computational cost of implementation for genuine RFID systems [290]. Furthermore, a study presented in [292] offered a unique tag number estimation scheme termed ‘Scalable Minimum Mean Square Error (sMMSE) which enhanced accuracy and reduced estimation time. The efficient modification of the frame size is derived from two principal parameters: the first one puts a limit on the slot occupancy whereupon frame size needs to be expanded, and the second determines the frame size expansion factor. In the research presented in [293], authors provided an in-depth analysis of some of the most relevant anti-collision algorithms taking into account the limitations imposed by EPCglobal Class-1 Gen-2 for passive RFID systems. The study classified and compared some of the most important algorithms and optimal frame length selection. Based on their research results, the authors point out that the maximum-likelihood algorithms achieve the best performance, in terms of mean identification delay. Finally, researchers conclude that the algorithm performance also depends on the computation time for estimating the number of tags.

A study presented in [294] introduced a new MFML-DFSA anticollision protocol. In order to increase the accuracy of the estimate, it uses a maximum-likelihood estimator that makes use of statistical data from many frames (multiframe estimation). The algorithm chooses the ideal frame length for the following reading frame based on the anticipated number of tags, taking into account the limitations of the EPCglobal Class-1 Gen-2 standard. The MFML-DFSA algorithm outperforms earlier suggestions in terms of average identification time and computing cost (which is lower), making it appropriate for use in commercial RFID readers. Rather novel research given in [295], proposes an RFID tag anti-collision method that applies adjustable frame length modification. The original tag number is estimated based on the initial assumption that the number of tags identified in the first frame are known. The authors present a nonlinear transcendental equation-based DFSA (NTEBD) algorithm and



compare it to the ALOHA algorithm demonstrating the error rate for experimental results to be less than 5% and improved tag identification throughput by 50%. Authors in [296] present an extension for an anti-collision estimator based on a binomial distribution. They have constructed a simulation module to examine estimator performance in diverse scenarios and have shown that the proposed extension has enhanced performance in comparison to other estimators, no matter if the number of tags is 1000 or 10 000.

As can be observed, previously mentioned algorithms tend to have high computational costs, since they are commonly funded by calculating probabilities. This might present an issue for standard microprocessors that are not adjusted to perform computations of factorials that produce large numbers. To solve the issue, a diverse method for tag estimation has been introduced by researchers in [95], namely the Improved Linearized Combinatorial model. Their approach bypasses the conditional probability calculations by doing them in advance. Onward, the estimation model is uncomplicated and provides an effective tag estimation  $\hat{n}$  based on linear interpolation. The results obtained by the authors have indicated that the ILCM shows a comparable behavior to state of art algorithms regarding the identification delay (slots), but is not computationally complex. Extension of their study was done in [113] by presenting a C-MAP anti-collision algorithm for RFID system which has lower memory demands.

Novel research, like the one presented in [297] present a comprehensive review and analysis of tag identification protocols in UHF RFID systems, focusing on the aspects of time and energy consumption. The authors examine prior works in the field and propose a novel DFSA-based algorithm called TES-FAS for EPC C1 Gen2, which employs low-cost cardinality estimation using look-up tables and adaptive frame configuration based on different parameters. The algorithm incorporates a slot-by-slot mechanism within a sub-frame observation phase, allowing for fine-grained frame size adjustment and improved efficiency. The TES-FAS algorithm improves reading performance without requiring modifications to the hardware. Simulation results highlight the enhanced slot efficiency, time efficiency, and energy efficiency achieved by TES-FAS compared to prior approaches. Additionally, the researches have developed a prototype RFID system, and experimental results indicate that TES-FAS outperforms commercial solutions, improving the average identification rate by 22.4% and 28.9% in different scenarios. These promising outcomes establish TES-FAS as a viable option for commercial and industrial RFID systems.

Paper given in [298] presents a novel approach to decrease the average time required to receive the Electronic Product Code (EPC) and read a sensor data packet from a tag in UHF RFID systems. The proposed approach, called Fuzzy Frame Slotted Aloha (FFSA) protocol, outperforms existing DFSA strategies compliant with the current standard, resulting in improved efficiency. Authors imply that the FFSA protocol that meets the requirements of the EPC C1G2 standard and can be easily implemented using existing RFID technology infrastructure. They present the development of a frame update policy and FFSA protocol that

significantly reduce the time for the reader to receive sensor data packets. Furthermore, they utilize the Mean Minimum Square Error estimator to calculate the frame size in FFSA, leading to improved performance compared to existing anti-collision protocols. FFSA conforms to the UHF RFID standard, allowing it to recognize commercial sensor tags, particularly in scenarios where low sensor data read time is essential. Through extensive experimentation, this paper demonstrates the superiority of FFSA over current anti-collision protocols, highlighting its potential for enhancing the performance of UHF RFID systems with varying numbers of sensor tags.

In research presented in [299] authors introduce a novel cross-layer anti-collision algorithm for slotted ALOHA-based UHF RFID systems. The proposed algorithm, called Cross Layer Anti-Collision Algorithm (CLAA), addresses the limitations of existing approaches by considering a complete slot-by-slot (SbS) estimation method and reducing the computation requirements in the MAC layer. The contributions of their work are twofold. Firstly, a modified Bayesian inference algorithm is proposed for accurate estimation of the number of tags in the MAC layer. Unlike existing approaches, the proposed SbS estimator maintains its accuracy even when a new frame is set up. This improves system efficiency and reduces the sensitivity to the initial frame length and the number of tags. Secondly, the computation requirement in the MAC layer is significantly reduced to a fixed number of floating-point operations per time slot, approaching the efficiency of the ILCM model. Additionally, the proposed algorithm introduces an additional computation part in the physical layer, which can be implemented in hardware for faster and more stable calculations. Simulation results demonstrate the effectiveness of the CLAA algorithm and its superior system efficiency compared to state-of-the-art DFS approaches. The proposed method achieves an average system efficiency of 35.64% for EPC systems and 36.67% for non-EPC systems, considering a uniform distribution of tags ranging from 0 to 1000. The CLAA algorithm exhibits robust performance independent of the number of tags and the initial frame length, thanks to its slot-by-slot nature. Furthermore, the algorithm strikes a good balance between performance and computation cost, reducing the computational requirements in the MAC layer while utilizing the available resources in the physical layer for stable and efficient calculations.

Effective anti-collision algorithms are essential in RFID applications that involve a large number of tags. These algorithms play a crucial role in reducing communication overhead, thereby enhancing the energy and time efficiency of RFID systems. While existing MAC algorithms primarily focus on improving system throughput and reducing identification time, the increasing significance of embedded systems and mobile applications demands the integration of energy consumption considerations in the design of new anti-collision algorithms [300].

## 2.4 Application Layer: Overview of Scientific Literature

Toys play a significant role in facilitating children's expression, fantasy, and cognitive development [301, 302]. With the advancement of technology, technology-based toys have gained popularity, with Smart Toys emerging as a favored choice [66]. The application of technology to toys and its impact on children's interaction with them has become an area of increasing focus for the scientific community due to the vital role that toys play in the development of children [67]. Smart toys, which incorporate digital features such as software or sensors, provide a more interactive environment than traditional toys, fostering the development of cognitive, social and behavioral skills in children [66, 303]. According to toys manufacturers and marketers, the possibilities of using smart and connected toys in education appear to offer rich, interactive, innovative, and mobile learning experiences in preschool children [304]. Smart toys distinguish themselves by integrating tangible objects with electronic components, enabling interactive child-toy interactions and purposeful tasks, thus offering a dynamic environment that promotes cognitive, social, and behavioral development[67]. Smart toys can be categorized based on the tasks they initiate, such as behavioral tasks or cognitive tasks [66]. Behavioral Smart Toys, like the Furby or Fisher Price's Learning Kitchen, aim to enhance children's behavioral skills through caring for and interacting with the toy [305]. Cognitive Smart Toys, such as StoryMat or Fisher Price's Learning Lantern, focus on developing cognitive skills through activities like storytelling or teaching numbers and concepts [306]. Moreover, Smart Toys can be classified based on their interactions, either with computers or as self-contained devices [66]. Examples of smart toys that interact with computers include Rosebud and StoryTech, which integrate stuffed animals or plush toys with a computer to enable storytelling and imaginative play [307, 308]. In contrast, self-contained Smart Toys, such as Sifteo blocks, have integrated digital features within their structure and provide opportunities for creativity and learning through physical interactions [309]. Two-way child-toy interactions are a significant characteristic of smart toys, combining physical and virtual realities to provide richer experiences [310]. Electronic sensors in smart toys facilitate meaningful interactions, allowing children to engage in purposeful tasks and build knowledge relationships [311, 312]. As children interact with the toys' technical and educational components, interaction with Smart Toys fosters learning, creativity, and imagination. For example, StoryTech enables children to improve storytelling skills and imagination through interactions with plush toys and multimedia features, while curlybot encourages computational and mathematical thinking through play [308, 313].

Futhermore, Smart Toys leverage Tangible User Interfaces to provide children with physical interactions that bridge the gap between the physical and digital worlds, enhancing their engagement and learning experiences [314]. TUIs enable children to manipulate and interact with digital elements through tangible objects and physical gestures, facilitating a deeper understanding of abstract concepts[315, 316].

Studies have shown the positive impact of TUIs in Smart Toys, promoting collaborative learning, problem-solving skills, and cognitive development [317, 318]. For example, the TUI-based Smart Toy platform, Osmo, integrates physical objects with digital activities, fostering spatial reasoning, mathematical thinking, and creativity [319]. Similarly, the Cubelets robotic construction kit utilizes TUIs to allow children to physically assemble and program robots, promoting computational thinking and engineering concepts [320]. By incorporating TUIs into smart toy design, these studies demonstrate how the physicality and tangibility of interactions contribute to children's engagement and learning outcomes.

Smart toys have also emerged as a promising tool for STEM education in preschool children [321]. For successful STEM education, research has emphasized the importance of improving mathematical skills, programming skills, and problem-solving skills.

The design and implementation of technology for learning cannot take place without taking into account the psychological aspects of a child's development that affect their ability to learn and interact with technology, on the one hand, and the pedagogical practices that improve those abilities, on the other [65, 62, 322]. STEM education for children is based on the principle of introducing them to programming through a high-level language, which was pioneered by Seymour Papert [323] with his development of Logo Turtles. This approach is based on Piaget's theory of cognitive constructivism [324]. In recent decades, educational technology research has been influenced by Piaget's theory of cognitive development and Montessori's educational approach, which emphasize the importance of hands-on learning and manipulation of objects in the development of logical-mathematical knowledge [324, 325]. Researchers and practitioners have embraced Piaget's theory of cognitive development and the significance of play in learning. Studies have shown that interactive smart toys, such as robotic kits, foster children's problem-solving skills and enhance their spatial cognition [137, 326] providing hands-on experiences that align with Piaget's view of children as active constructors of knowledge. On the other hand, Papert's concept of constructionism has influenced the design of smart toys that promote creative thinking and programming skills. For example, programmable robots like LEGO Mindstorms and KIBO enable children to build and code their own creations, fostering computational thinking and problem-solving abilities [327, 328]. The Montessori approach, emphasizing self-directed learning and independent exploration, has also been integrated into the design of Smart Toys. For instance, interactive learning platforms, such as tablet-based applications, have been found to support children's independent learning, creativity, and self-regulation skills [329, 330]. These Smart Toys provide opportunities for children to engage in autonomous exploration and personalized learning experiences.

Studies have shown that physical manipulation plays a critical role in the development of thinking skills, enabling the transition between physical and virtual representations and simplifying abstract concepts for young children [331]. Interactive features such as sound, animation, and movement-initiated feedback can also provide rich contextual information to

enhance learning and motivate children to complete tasks successfully [61].

A study presented in [67] provides a review of Smart toys from the last 30 years, focusing on toys for children in early childhood and primary school by analyzing and categorizing Smart toys based on their technological and educational capabilities. One of the major findings of the study emphasizes that in recent years smart toys have focused on special sciences (programming) and some skills of the 21st century (STEM and computational thinking). On the contrary, in the first 20 years, greater emphasis was placed on cross-disciplinary skills such as collaboration, emotional thinking, symbolic thinking, storytelling, and problem solving. Another novel research presented in [332] aimed to review 30 computational toys and kits designed for children aged 7 years and under, including physical, virtual, and hybrid kits. Qualitative analysis examines the kits' design, support for exploring computational concepts and practices, participation in projects and activities, and exploration of other domains of knowledge. The study presents design suggestions and opportunities to expand the exploration of computational concepts, new modes of expression, and expanded support for children from underrepresented groups in computing. The findings reveal commonalities between existing kits and suggest ways for designers and researchers to improve the possibilities for children to create, explore and play with computing.

Smart toys are now being scientifically researched in the STEM context for preschool education. For example, the KIBO Robot Demo is an educational robot designed to teach young children (ages 4 to 8) programming and engineering concepts [333]. The children can program the robot using wooden blocks with barcodes, learning basic programming concepts such as sequencing, loops, and conditional statements. The system has been tested in a variety of settings and has been shown to effectively engage children in programming and engineering. A research presented in [334] focuses on the development of a Smart toy called ABBOT, designed to motivate children to become outdoor explorers. ABBOT is equipped with sensors that allow it to collect environmental data such as temperature, humidity, and light levels. The device is also designed to encourage children to participate in outdoor activities and learn about their environment by providing feedback and rewards. The research study presented in [335] describes the anthropomorphic design, development, and testing of a prototype called OBSY, which is an Observation Learning System aimed at facilitate learning of science concepts for primary school children in Thailand. The system consists of an ubiquitous sensor-based device that resembles an octopus and a mobile web application hosted on the device. Sensors attached to the OBSY device collect environmental data, which is then interpreted using the web application accessed through tablet computers. The system was developed through a user-centered design approach and aims to promote science learning in an engaging and interactive way. The study presented in [336] describes the design and interactive behavior of a tangible augmented reality toy kit that teaches preschool children about color mixing, mathematics and geometric 2D-3D shapes. The game allows children to interact with both physical and in-screen objects, using touch-screen and AR

interactions. Researches conclude that the game has the potential to improve the learning experience for young children and promote interest in STEM fields.

In pilot study presented in [326], focus was on exploring the potential of the Kindergarten Social Assistive Robot (KindSAR) as an innovative tool for promoting children's development through social interaction. The study aimed to investigate how KindSAR could assist educational staff in teaching geometric thinking and promoting metacognitive development by engaging children in interactive play activities. The findings revealed that most children demonstrated positive interaction with the robot and expressed a high level of enjoyment. Furthermore, the results indicated that children's performances in terms of geometric thinking and metacognitive tasks improved while they engaged in play activities with the robot. To assess the children's learning, a unique measure called "velocity of learning" was developed and employed. Overall, this study showcased the feasibility of incorporating KindSAR into preschool education and highlighted the anticipated benefits it can bring in terms of children's cognitive development.

Through tangible programming, the study presented in [337] investigates the use of IoT technology in the smart farming education of children. It involves creating a tangible programming kit that simulates a smart farming system using sensors and Internet of Things devices. User testing revealed that the kit was effective in promoting engagement and learning in young children and has the potential to improve learning in the fields of agriculture and technology.

Research in [338] and [339] investigates coding with two commercial Smart toy robots *Dash* and *Botley* as part of playful learning in the context of Finnish early education. The results of our study show how Finnish preschoolers aged 5-6 years approached, conducted and played coding with two toy robots. The study's main conclusion is that preschoolers used toy robots with coding affordances primarily in developing gamified play around them by designing tracks for the toys, programming the toys to solve obstacle paths, and competing in player-generated dexterity, speed, and physically mobile play contests.

A rather recent study presented in [340] examines the effects of didactic approaches in guiding early childhood children learn computational logic and programming concepts. To develop the students' cognitive skills, a teaching methodology was developed that utilizes a commercial smart mBot Arduino robot. mBot is a beginner-friendly educational robot that makes programming and learning robots simple and enjoyable. mBot also helps develop logical thinking and design skills. The study concludes that the developed method enhances learning processes and computational thinking abilities.

In recent years, due to the development of Smart Toys enhanced with the Internet of Things (IoT) that can connect to the Internet, there has been a growing body of research on cyber security and privacy risks of smart toys. The studies presented in [341, 342] focus on reviewing major smart toys-related children's privacy risks and major mitigation so to such risks.

Despite requests from the scientific community to investigate how to best incorporate new technologies into the formal and informal learning contexts of young children, the design and development periods of new smart toys have not been adequately emphasized, as highlighted by the research in [305]. Therefore, they have applied the design and development research method to create guidelines for designing and using Smart toys for preschool children. Research examines a smart toy developed in a pilot study, holds focus group meetings with early childhood teachers, creates two prototypes, and tests them with preschool children, teachers, and scholars. The study divides the design guidelines into three categories: content, visual design, and interaction.

Based on the literature, the use of smart toys in preschool education represents a promising approach to fostering STEM skills in young children, and with that regard learning geometry at an early age is crucial for the development of spatial reasoning skill. Studies indicate that it is critical to introduce geometry in preschool period, when first critical geometrical observations are made [343, 344]. With that regard, recent studies on gestures emphasize the body's significance in spatial and geometric reasoning highlighting the importance of kinetic movement in the genesis and development of abstract geometrical cognition in early years [122, 345, 123, 124].





### **3 OVERVIEW OF THE SCIENTIFIC CONTRIBUTION OF PUBLICATIONS**

This chapter comprises abstracts of the publications, along with an overview of the scientific contributions upon which the dissertation is grounded. It specifically emphasizes the distinct contributions made by the doctoral candidate in each respective work.

#### **3.1 Paper 1: Machine Learning and Soil Humidity Sensing: Signal Strength Approach**

##### **Abstract:**

The Internet-of-Things vision of ubiquitous and pervasive computing gives rise to future smart irrigation systems comprising the physical and digital worlds. A smart irrigation ecosystem combined with Machine Learning can provide solutions that successfully solve the soil humidity sensing task in order to ensure optimal water usage. Existing solutions are based on data received from the power hungry/expensive sensors that are transmitting the sensed data over the wireless channel. Over time, the systems become difficult to maintain, especially in remote areas due to the battery replacement issues with a large number of devices. Therefore, a novel solution must provide an alternative, cost- and energy-effective device that has unique advantage over the existing solutions. This work explores the concept of a novel, low-power, LoRa-based, cost-effective system that achieves humidity sensing using Deep Learning techniques that can be employed to sense soil humidity with high accuracy simply by measuring the signal strength of the given underground beacon device.

##### **Overview of the scientific contributions**

A novel, low-power, LoRa-based, cost-effective system was introduced which achieves humidity sensing using Deep learning techniques that can be employed to sense soil humidity with high accuracy simply by measuring signal strength of the given underground beacon device. Machine Learning approach was proposed that leverages RSSI data from LoRaWAN devices to estimate soil moisture conditions. This aligns with the hypothesis, which suggests that Machine Learning algorithms can be used to estimate IoT environment conditions using

RSSI data. Data analysis collected from LoRaWAN-based Soil Moisture Sensor device was employed to uncover anomalies, define necessary data preparation approaches and determine potentially useful Machine Learning algorithms for the desired estimations. Such data analysis enabled the discovery of characteristic properties of the data with a goal to exploit how soil humidity is related to the signal strength. Indeed, it was shown that the correlation between RSSI and soil humidity is substantial, which supports the hypothesis that RSSI data can be used to estimate environmental conditions with high precision. The proposed Machine Learning approach can provide accurate estimates of soil moisture conditions in a time-series dataset. Use of Long Short-Term Memory (LSTM) Neural Network as a Deep Learning approach provided significant results in terms of accuracy of estimation in contrast to traditional ML techniques of Support Vector Regression. This aligns with hypothesis where Machine Learning algorithms can be used to estimate IoT environment conditions from the Perception Layer of the three-layer IoT architecture with high precision.

### **The doctoral candidate's contribution**

The doctoral candidate made significant contributions to the research thorough data analysis on the collected data from the LoRaWAN-based Soil Moisture Sensor device, uncovering anomalies and determining appropriate data preparation approaches. By establishing a substantial correlation between soil humidity and the received signal strength indicator and signal-to-noise ratio, the candidate provided valuable insights into the relationship between these variables. The doctoral candidate used Machine Learning approaches to evaluate soil moisture levels, first using the Support Vector Regression model and then applying the Long Short-Term Memory Neural Network. Furthermore, the candidate enhanced the Support Vector Regression (SVR) model by incorporating values from both the current and previous time steps, thus providing it with a "hybrid short-term memory." This improvement resulted in enhanced accuracy of the model in estimating soil moisture conditions, as it effectively considered the temporal relationship between measurements. The LSTM model outperformed the SVR model in terms of accuracy, making it more useful in assessing soil humidity. These contributions not only support the hypothesis that RSSI data can be used for precise environmental estimations, but also demonstrate the candidate's expertise in data analysis, correlation analysis, and the application of Machine Learning algorithms in the context of IoT environmental sensing.

## 3.2 Paper 2: Sensing Occupancy through Software: Smart Parking Proof of Concept

### Abstract:

In order to detect the vehicle presence in parking slots, different approaches have been utilized, which range from image recognition to sensing via detection nodes. The last one is usually based on getting the presence data from one or more sensors (commonly magnetic or IR-based), controlled and processed by a micro-controller that sends the data through radio interface. Consequently, given nodes have multiple components, adequate software is required for its control and state-machine to communicate its status to the receiver. This paper presents an alternative, cost-effective beacon-based mechanism for sensing the vehicle presence. It is based on the well-known effect that, once the metallic obstacle (i.e., vehicle) is on top of the sensing node, the signal strength will be attenuated, while the same shall be recognized at the receiver side. Therefore, the signal strength change conveys the information regarding the presence. Algorithms processing signal strength change at the receiver side to estimate the presence are required due to the stochastic nature of signal strength parameters. In order to prove the concept, experimental setup based on LoRa-based parking sensors was used to gather occupancy/signal strength data. In order to extract the information of presence, the Hidden Markov Model (HMM) was employed with accuracy of up to 96%, while the Neural Network (NN) approach reaches an accuracy of up to 97%. The given approach reduces the costs of the sensor production by at least 50%.

### Overview of the scientific contributions

The paper proposes a hardware sensing solution through software that uses signal strength information to achieve cost savings. LoRa-based parking sensors were used to gather occupancy/signal strength data, while Hidden Markov Model and Neural Network approaches were used to estimate the presence of vehicles with high accuracy. This approach leverages the Received Signal Strength Indicator data from LoRaWAN devices to estimate the occupancy status of parking slots. The contributions are in line with the proposed hypothesis of utilizing RSSI data from IoT devices to accurately estimate the conditions of the IoT environment. Moreover, data analysis approach was employed to detect the relationship between Occupancy Status, Received Signal Strength Indicator, and Signal to Noise Ratio of LoRa-based parking sensors. The analysis showed that parking lots are free considerably more than they are occupied, indicating the stochastic behavior of the parking place. The Hidden Markov Model of the second order was designed and used to classify the occupancy status of a parking space while using RSSI and SNR values. The accuracy achieved by the HMM approach was up to 96%, while the NN approach reached an accuracy of up to 97%. The find-

ings demonstrate the feasibility of implementing a cost-effective beacon-based mechanism for detecting vehicle presence, leading to significant cost reductions in sensor production.

### **The doctoral candidate's contribution**

The doctoral candidate made significant contributions to the field of smart parking through their research on sensing occupancy through software. They conducted extensive data analyses and observed that higher Received Signal Strength Indicator values indicate a free parking space, while lower values indicate an occupied one. They also noted that changes in the parking status corresponded to significant variations in RSSI and Signal to Noise Ratio values. To estimate vehicle presence accurately, the candidate employed two Machine Learning models: the second-order Hidden Markov Model and a Neural Network model with two hidden layers. In the HMM approach, the candidate introduced a novel "hybrid short-term memory" technique by defining the observable states as the changes in RSSI values between two previous states. The Viterbi algorithm was used to determine the best state sequence for the given model and observations. The NN model, on the other hand, utilized sensor ID, RSSI, SNR, gateway ID, and timestamp as input variables. The NN model outperformed the HMM model, achieving superior accuracy in classifying parking spaces compared to other researched literature. Overall, the candidate's research showcases the feasibility of a cost-effective beacon-based mechanism for detecting vehicle presence in smart parking systems. These findings have significant implications for reducing sensor production costs and improving the efficiency of smart parking technologies. The candidate's research contributes to the field by investigating the feasibility of estimating the conditions of the Internet of Things environment with high precision.

## **3.3 Paper 3: Privacy leakage of LoRaWAN smart parking occupancy sensors**

### **Abstract:**

Development of smart cities is enabled by its core concepts of smart and sustainable mobility, where Low Power Wide Area Network (LPWAN) such as Long Range Wide Area Network (LoRaWAN) became one of the most important Internet of Things (IoT) technologies. Due to its low power consumption, simple setup, and large communication range, LoRaWAN smart parking devices are already employed to reduce congestion and improve quality of life. This paper studies privacy leakage of LoRaWAN smart parking communication devices. Namely, when a vehicle as a metallic obstacle obscures the LoRaWAN smart parking device, the signal strength will be significantly reduced on the receiver side. Consequently, the variation in the signal strength of LoRaWAN parking systems transmits information about

parking space occupancy, allowing the implementation of a passive side-channel attack at large distances. Using supervised machine learning techniques based on Neural Network, the attacker can estimate parking lot occupancy with accuracy up to 97%, while Random Forrest approach reaches the accuracy over 98%.

### **Overview of the scientific contributions**

The primary scientific contribution involves the extensive investigation of Machine Learning algorithms' capacity to harness RSSI data from LoRaWAN devices for the purpose of achieving accurate detection of parking space occupancy. The paper explores the utilization of this approach while delving into the implications of potential privacy breaches that may result from the exploitation of this information by malicious actors. Data analysis techniques were employed to identify potential data correlations, uncover anomalies, and determine appropriate approaches for data pre-processing. Specifically, analyzed RSSI data collected from the LoRaWAN devices was used to identify patterns and trends that could be used to accurately estimate parking space occupancy. Exploratory data analysis was also used to identify any outliers or anomalies in the data that could affect the performance of the Machine Learning algorithms. Finally, statistical techniques such as feature selection and dimensionality reduction was used to pre-process the data before training and testing the Machine Learning models. Namely, using supervised machine learning methodology based on Random Forest and Neural Networks, it was shown that parking space occupancy can be estimated by using RSSI data from LoRaWAN devices. The Random Forest algorithm achieved an accuracy of 98%, while the Neural Network algorithm achieved an accuracy of 97%. Through modeling, development, and testing of these Machine Learning algorithms, the it was shown that is indeed feasible to estimate environmental conditions from the Perception Layer of the three-layer IoT architecture with high precision.

### **The doctoral candidate's contribution**

The doctoral candidate's contributions in this research paper encompass various aspects of the study, including data analysis, feature engineering, model selection, performance comparison, and computational complexity analysis. These contributions shed light on the feasibility of utilizing RSSI data from LoRaWAN devices for accurate detection of parking space occupancy while addressing potential privacy concerns. The candidate conducted extensive data analysis, which revealed the impact of parking status changes on RSSI and SNR values, as well as the correlation between gateway distance and the change in parking occupancy. Furthermore, they identified the data set's skewness, indicating that parking spaces were less frequently occupied than free. To mitigate this data imbalance, the Synthetic Minority Over-sampling Technique was applied. The candidate enhanced the data set by incorporating time variables to capture temporal dependence and seasonality effects. Hour, day, month, and day

of the week were included as features, providing deeper insights into the historical occupancy patterns of parking spaces. The candidate employed two Machine Learning models, Random Forest and Neural Network, to estimate parking lot occupancy based on RSSI data. Separate models were built and evaluated for each sensor. The Random Forest model outperformed the Neural Network model, achieving an accuracy of 98% in estimating parking space occupancy. The candidate conducted a comprehensive statistical comparison of model performance. They evaluated the classification accuracy of both Random Forest and Neural Network models after balancing the data set and including time variables, as well as after excluding the time variable. The candidate also performed a comparison of computational complexity between the two models. They measured the actual run-time of each model for five sensors, providing insights into their efficiency and resource requirements. The statistical comparison between the Random Forest and Neural Network models holds significant scientific importance as it enables the identification of the optimal Machine Learning approach for precise estimation of parking space occupancy.

### **3.4 Paper 4: Tag Estimation Method for ALOHA RFID System Based on Machine Learning Classifiers**

#### **Abstract:**

In the last two decades, Radio Frequency Identification (RFID) technology has attained prominent performance improvement and has been recognized as one of the key enablers of the Internet of Things (IoT) concepts. In parallel, extensive employment of Machine Learning (ML) algorithms in diverse IoT areas has led to numerous advantages that increase successful utilization in different scenarios. The work presented in this paper provides a use-case feasibility analysis of the implementation of ML algorithms for the estimation of ALOHA-based frame size in the RFID Gen2 system. Findings presented in this research indicate that the examined ML algorithms can be deployed on modern state-of-the-art resource-constrained microcontrollers enhancing system throughput. In addition, such utilization can cope with latency since the execution time is sufficient to meet protocol needs.

#### **Overview of the scientific contributions**

This study presents the scientific contributions derived from the research conducted on the Network Layer of the three-layer IoT architecture within the context of the dissertation. The contributions align with the hypothesis that leveraging Machine Learning models to estimate frame size and tag count can enhance the throughput of RFID Gen2 systems that employ the ALOHA protocol. The key contributions are as follows: The research delves into the feasibility and effectiveness of utilizing ML models, namely the Neural Network and Ran-

dom Forest, on the Network Layer of the three-layer IoT architecture. Specifically, the ML models are employed to estimate the frame size and tag count in ALOHA-based RFID Gen2 systems. This exploration contributes to the advancement of ML techniques in enhancing the overall throughput of RFID systems. The research aimed to demonstrate the robust performance and generalizability of the ML models across different frame sizes within diverse data sets. Through meticulous data set design and evaluation derived from Monte Carlo simulations, the study showcased the effectiveness of the ML models in capturing the nuances of different data sets. The findings emphasize the models' capability to deliver consistent and reliable results, regardless of the frame size variations. This investigation significantly contributes to understanding the ML techniques' adaptability and reliability in real-world IoT environments. A comparative analysis is conducted to evaluate the performance of ML models in estimating frame size and tag count. The ML models are compared against a state-of-the-art algorithms, namely the Improved Linearized Combinatorial Model. The analysis reveals that the ML models outperform the traditional method and achieve higher accuracy rates. The higher throughput achieved by the Neural Network model compared to the ILCM model in the given context indicates its capability to achieve a greater number of successful slots per frame, resulting in enhanced system efficiency. Furthermore, this research investigates the implementation of ML models on resource-constrained devices commonly used in IoT applications, while also exploring the benefits of smart quantization techniques. By leveraging techniques such as 16-bit float-point and 8-bit integer approximations, the research demonstrates significant reductions in memory requirements and improved execution times. The findings highlight that through quantization, ML models can be executed efficiently on memory-restricted microcontrollers, making them more accessible for practical implementation.

### **The doctoral candidate's contribution**

The doctoral candidate conducted Monte Carlo simulations and implemented the code to generate the data set used in the research. The meticulous design and implementation of the data set aimed to closely emulate real-life scenarios and enable an adequate comparison with state-of-the-art algorithms. This approach for Monte Carlo simulations of the tag distribution in the slots was informed by previous research studies in the field. The candidate conducted an evaluation of Machine Learning models, including Neural Network and Random Forest, to estimate tag count in ALOHA-based RFID Gen2 systems. The evaluation demonstrated the strong performance and adaptability of the ML models across various data sets. Additionally, a comparative analysis was performed, comparing the ML models to the ILCM. The latter showcased the ML models' strong performance and adaptability across diverse data sets, particularly when dealing with larger frame sizes and more tag distribution options, in terms of accuracy rates. Based on the evaluation of classifier performance and comparison to the ILCM model, the candidate selected Neural Network architectures

for further utilization, showcasing enhanced performance in terms of throughput when compared to the ILCM model. The Neural Network model demonstrated improved efficiency in accurately estimating the number of interrogating tags, aligning with the study's objective to enhance system throughput. Further analysis revealed differences in computational burden, with the NN model outperforming the ILCM model in terms of successful slots per frame. This analysis provided valuable insights into the computational costs and efficiency of the Neural Network model. By leveraging Machine Learning models to estimate frame size and tag count, the doctoral candidate has demonstrated the feasibility of enhancing the throughput of ALOHA-based RFID systems that employ the ALOHA protocol on the Network Layer of the three-layer IoT architecture. These contributions significantly align with the hypothesis and advance the understanding and application of Machine Learning in improving the efficiency and throughput of RFID systems.

### **3.5 Paper 5: Tangible Interfaces in Early Years' Education: A Systematic Review**

#### **Abstract:**

This paper presents a systematic review of the literature on Tangible User Interfaces (TUIs) and interactions in young children's education by identifying 155 studies published between 2001 and 2019. The review was based on a set of clear research questions addressing application domains, forms of tangible objects, TUI design and assessment. The results indicate that (i) the form of tangible object is closely related to the application domain, (ii) the manipulatives are the most dominant form of tangible object, (iii) the majority of studies addressed all three stages of TUI development (design, implementation and evaluation) and declared a small sample of young children as a major shortcoming, and (iv) additional empirical research is required to collect evidence that TUIs are truly beneficial for children's acquisition of knowledge. This review also identifies gaps in the current work, thus providing suggestions for future research in TUIs application in educational context expected to be beneficial for researchers, curriculum designers and practitioners in early years' education. To the authors' knowledge, this is the first systematic review specific to TUIs' studies in early years' education and is an asset to the scientific community.

#### **Overview of the scientific contributions**

This paper presents a comprehensive scientific exploration of Tangible User Interfaces in the context of emerging technologies and their applications in children's education. The review focused on research questions related to application domains, forms of tangible objects, TUI design, and assessment. The findings of the review revealed several key contributions.



Firstly, the study defined the term "form of tangible object" and identified three distinctive embodiments through which the interaction happens: manipulatives, tabletops, and tablets. This definition provides a comprehensive framework for understanding and categorizing the various types of tangible objects used in TUIs. Manipulatives were found to be the most dominant form of tangible objects used in TUIs. Additionally, the majority of the reviewed studies addressed all three stages of TUI development, namely design, implementation, and evaluation. However, a common limitation identified was the small sample size of young children involved in the studies, highlighting the need for larger and more diverse participant groups in future research. The review also identified fourteen application domains for TUIs in children's education, ranging from Art and Literacy to Science and Problem Solving. This demonstrates the wide range of educational contexts in which TUIs have been applied successfully. The benefits of tangible interaction, such as increased social interaction and collaboration among children, were consistently highlighted across the studies. In addition, the review identifies areas of improvement in current research, including the use of small sample sizes and limited generalizability. It also offers insights into future research directions in this field. One such direction is exploring the potential of combining TUIs with IoT technologies, which shows promise in enabling innovative educational applications and enhancing learning outcomes for children.

### **The doctoral candidate's contribution**

The doctoral candidate has made significant contributions to the field of Tangible User Interfaces in early years' education through their meticulous and comprehensive review and analyses of the researched literature.

One of the notable contributions of the candidate's work is the formulation of a comprehensive framework for understanding and categorizing the "form of tangible object" in TUIs. By identifying three distinct embodiments - manipulatives, tabletops, and tablets - the candidate has provided a clear and structured approach for researchers and practitioners to analyze and develop tangible objects in educational settings. Additionally, the candidate's review highlighted the prevalence of manipulatives as the most dominant form of tangible objects in TUIs, underscoring their significance in facilitating hands-on experiences and physical interaction, which are crucial for young children's engagement and learning. Furthermore, the candidate established a clear relationship between the forms of tangible objects and the application domains in TUIs, providing valuable insights into how different forms of tangible objects are suited for specific educational contexts. Additionally, the candidate explored the relation between children's age and application domain, shedding light on how the appropriateness and effectiveness of TUIs may vary based on the developmental stage of the children. Moreover, the candidate actively addressed the gaps and limitations present in the researched publications. They highlighted the need for larger sample sizes and more diverse participant groups in order to enhance the generalizability of findings and ensure

the validity of research outcomes. By identifying these gaps, the candidate has paved the way for future studies to address these limitations and further advance the field of TUIs in early years' education. Finally, the candidate explored the potential of integrating TUIs with Internet of Things technologies, envisioning innovative educational applications that capitalize on the synergistic benefits of both fields. This forward-thinking perspective opens up prospective possibilities for creating interactive and personalized learning experiences for young children.

### **3.6 Paper 6: Towards a Machine Learning Smart Toy Design for Early Childhood Geometry Education: Usability and Performance**

#### **Abstract:**

This study presents the design and evaluation of a plush smart toy prototype for teaching geometry shapes to young children. The hardware design involves the integration of sensors, microcontrollers, an LCD screen, and a machine learning algorithm to enable gesture recognition by the toy. The machine learning algorithm detects whether the child's gesture outline matches the shape displayed on the LCD screen. A pilot study was conducted with 14 preschool children to assess the usability and performance of the smart toy. The results indicate that the smart toy is easy to use, engages children in learning, and has the potential to be an effective educational tool for preschool children. The findings suggest that smart toys with machine learning algorithms can be used to enhance young children's learning experiences in a fun and engaging way. This study highlights the importance of designing user-friendly toys that support children's learning and underscores the potential of machine learning algorithms in developing effective educational toys.

#### **Overview of the scientific contributions**

This research integrates IoT sensing technology and Machine Learning algorithms to achieve precise detection and interpretation of intricate human gestures for interaction. By leveraging sensor data output, the study provides empirical evidence of the effectiveness of ML algorithms in enabling gesture-initiated feedback. These findings support the hypothesis that accurate detection and interpretation of complex human gestures for interaction can be achieved on the Application Layer of the three-layer IoT architecture. Additionally, the paper presents the design and evaluation of a prototype plush Smart Toy specifically tailored for teaching geometry shapes to young children. The utilization of ML algorithms for gesture recognition is explored, and the findings of a pilot user study involving preschool-aged children interacting with the prototype toy during an experimental session are reported. In

terms of results, the paper highlights the successful application of a 1D Convolutional Neural Network as the final Machine Learning model for gesture recognition. The CNN model achieves high accuracy levels ranging from 93.8% up to 98.3% when tested on a data set comprising gestures performed by adult users. This outcome underscores the potential of ML in accurately classifying gestures and enhancing shape identification in early childhood geometry education. The paper highlights the importance of pilot testing for evaluating the usability of the Smart Toy, focusing on aspects such as usability, engagement levels, and motor interactions. Specific details about the research methods employed are provided, and observations of motor aspects, engagement levels, and overall behavior are recorded, offering valuable insights into the usability of the Smart Toy for Early Childhood Geometry Education. The exploratory study yielded significant findings regarding the benefits and strengths of the Smart Toy prototype. The results indicate that the prototype is user-friendly, easily manageable, and effectively engages children in the learning process. This suggests its potential as a valuable educational tool for preschool-aged children. The study also provides insights into the different modes of interaction between children and the toy, the distinctions in gestures made by children compared to adults, and the impact of age on user performance in gesture-based interfaces. These findings enhance the understanding of the factors influencing user performance and highlight the need to consider potential confounding variables during data analysis. In conclusion, the research findings emphasize the importance of involving children in the design process and highlights the need for comprehensive data sets of children's gestures to enhance the accuracy of ML models in gesture classification.

### **The doctoral candidate's contribution**

In this research, the doctoral candidate has made significant contributions to the design, implementation, and evaluation of a plush Smart Toy prototype aimed at teaching geometry shapes to young children. The candidate played a pivotal role in the inception of the toy's interaction concept, which utilizes hand movements, as well as the selection of the application domain of geometry based on relevant literature research. The candidate's involvement also extended to the analysis of collected sensor data and the testing of Machine Learning algorithms for gesture recognition using different data sets. The results of the candidate's research demonstrate the effectiveness of this approach, revealing that the applied Machine Learning algorithms can achieve reliable gesture recognition, supporting the feasibility of achieving accurate interaction through the utilization of sensor data in the proposed IoT architecture. This aligns with the hypothesis that highly accurate detection and interpretation of complex human gestures for interaction can be achieved on the Application Layer of the three-layer IoT architecture by employing Machine Learning algorithms based on the sensor data output.

Through collaborative efforts, the candidate shaped the methodology for the usability study, focusing on aspects such as usability, engagement levels, and motor interactions. This

involved conducting statistical analyses of objective and subjective aspects of children's interactions with the smart toy.

Additionally, the candidate contributed to the experimental design, including the selection of materials and methods, as well as the experiment procedure. The candidate conducted an extensive analysis of the collected data from the usability study, providing valuable insights into user performance and emphasizing the need to consider potential confounding variables during data analysis. The research findings shed light on factors influencing user performance and highlight the importance of involving children in the design process. Furthermore, the candidate's research revealed a correlation between the perceived difficulty of a shape and the real-time required to draw it, which can be valuable in developing instructional materials and activities that objectively measure shape difficulty. Overall, the candidate's contributions in this research have significantly advanced the understanding of Machine Learning-enabled Smart Toy design for early childhood geometry education. The candidate's involvement in shaping the interaction concept, conducting analyses, and evaluating user performance has enhanced the scientific knowledge in this field and has practical implications for designing effective educational toys for young children.

## 4 CONCLUSION

As we move further into the 21st century, the Internet of Things (IoT) continues to revolutionize various aspects of our lives, shaping new paradigms with its ubiquitous, connected technology. The IoT landscape, characterized by its complex interplay of "things," data, people, and processes, has immense potential that we are continually learning to harness. The breakthrough advancements in this field span across several layers of IoT architecture, each addressing unique sectors and challenges. Not only have these propelled the growth and efficiency of interconnected devices and systems, but they have also paved the way for progressive leaps in industries like healthcare, logistics, transportation, and education. With that regard, this dissertation extensively explores the potential of Machine Learning (ML) techniques in augmenting the functionality and performance of Internet of Things services within the three layers of the IoT architecture: Perception, Network, and Application layers. Throughout the research, special attention has been given to considering the usability context in the development and integration of ML algorithms. By leveraging ML techniques, this dissertation offers valuable scientific contributions to enhance the performance of Internet of Things Stack Services. The integration of ML algorithms into IoT systems leads to significant advancements in data handling and processing, thereby improving the overall performance of IoT services in various domains.

In the Perception layer, the research provides vital insights into how ML algorithms can better handle and process large volumes of data. Practical examples such as smart parking systems and soil humidity sensing in the context of precision farming demonstrate the potential of ML to transform raw IoT data into valuable information, leading to effective city resources management and improved agricultural practices, respectively. The exploration of soil humidity sensing through signal strength approach introduced a novel mechanism that circumvents the traditional, costly, and energy-consuming sensors. Machine learning models empowered with high accuracy signal strength data have proven highly effective at estimating the soil moisture conditions. This innovation not only assures optimal water usage for smart irrigation systems but also resolves the issue of maintenance in remote locations that often arises due to battery replacement needs by offering cost-effective, sustainable, and dependable solutions. On the other hand, the incorporation of machine learning in smart parking has been equally transformative. The investigation of occupancy sensing through software leveraged Received Signal Strength Indicator (RSSI) data from the LoRaWAN de-

VICES to estimate the occupancy status of parking slots with an impressive accuracy rate. Traditional methods which depended heavily on sensor technologies, such as infrared or magnetic sensors, were cast aside for a more cost-effective and reliable beacon-based mechanism for detecting vehicle presence. Both instances underline the effective adaptation of machine learning in enhancing the perception capabilities of devices, which results in managing resources more efficiently, and highlights the potential of improving IoT through machine learning. Moreover, the part of research also underscores the importance of addressing privacy leakage concerns in IoT technologies highlighting the need for robust security measures to ensure the safe handling of sensitive data within IoT systems that incorporate LoRaWAN devices.

The research conducted in the Network layer of this dissertation focuses on the integration of Machine Learning techniques within IoT systems, with a specific emphasis on the utilization of RFID technology. One of the key challenges faced by RFID technology in large-scale infrastructures, such as commercial warehouses, is the efficient reading of a vast number of RFID tags. By harnessing the power of ML models, such as Neural Networks, the dissertation explores how RFID systems can be optimized for tag number estimation, resulting in enhanced performance. These models can effectively estimate the optimal frame size and tag count in RFID systems that use the ALOHA protocol. The use of Machine Learning models has elevated the effectiveness of these systems compared to traditional estimation methods, providing robust performance across many frame sizes and a wide range of data sets. A key aspect of this integration involves quantization of the ML models. Quantization significantly reduces the memory requirements of the models, making them more practical and efficient to use on memory-restricted microcontrollers, which are integral components of IoT and embedded devices. The findings from this research validate the feasibility of utilizing ML models to enhance the throughput of RFID-based IoT systems, opening doors to the development of more resource-efficient and effective systems in commercial operations and beyond.

Within the Application layer, this research places a strong emphasis on the development of a Smart Toy prototype that leverages Machine Learning algorithms for gesture recognition and user interaction. By considering the usability context, the dissertation showcases how ML techniques enhance the educational potential of the smart toy, presenting innovative applications of the Internet of Things in early childhood education. This pioneering toy integrates ML algorithms and IoT sensing technology, representing a fusion of advanced technologies in the context of early childhood education. The Smart Toy, a notable achievement of this research, employs hand movements and gesture recognition to teach young children about geometry shapes. This functionality is made possible by a 1D Convolutional Neural Network (CNN), that compares the child's hand-drawn shape with the one displayed on the toy's LCD screen. This unique interactive experience not only captures children's interest but also enhances their learning process. The results from a pilot user study involving

preschool-aged children further confirm the viability and effectiveness of the Smart Toy in educational settings. The Smart Toy was found to be user-friendly, engaging, and successful in teaching geometry shapes to the children. The successful integration of ML and IoT in creating interactive and personalized learning experiences signifies their immense potential in the field of education.

The implications of this research for the field of Internet of Things are manifold. Firstly, it underscores the potential for enhanced efficiency in data management through the application of Machine Learning algorithms. By converting raw data into actionable insights, IoT devices can operate more effectively and respond proactively to changing environmental conditions. moreover, ML models can analyze large amounts of data generated by IoT devices, extracting significant information and correlations. This efficient big data analysis provides deeper insights into data, mines hidden correlations, and aids precise predictions based on past observations. Secondly, the study reveals how ML can optimize the performance of IoT sensors, improving accuracy, response time, and energy consumption. This contributes to the reliability and effectiveness of IoT networks and devices. Thirdly, the integration of ML techniques enables advanced automated decision-making processes in IoT systems, leveraging past experiences and predictive analytics to make data-driven decisions. This significantly enhances service delivery and operational efficiency. Furthermore, the research highlights the potential for cost reduction and energy conservation in IoT systems. For example, in the context of smart parking, ML-based condition estimation can replace expensive sensors, reducing overall costs while maintaining efficient functionality. Lastly, the study emphasizes the importance of addressing privacy and security concerns in IoT, emphasizing the need for robust measures to protect sensitive data and ensure the integrity of IoT systems. In conclusion, this research opens up new possibilities for improving data management, sensor performance, decision-making, cost-effectiveness, and security in the realm of IoT, advancing the field and shaping its future development.

Moving forward, future research in this area should continue to explore innovative ML algorithms and techniques that further enhance the usability of IoT systems. Additionally, the development of standardized frameworks and guidelines that integrate usability and security considerations will be crucial for the widespread adoption and success of ML-enabled IoT applications. By continuing to bridge the gap between ML and IoT, we can unlock the full potential of these technologies and create a more connected, efficient, and user-centric IoT ecosystem. Overall, this dissertation concludes that the integration of ML techniques within the IoT architecture can significantly enhance the functionality, performance, and usability of IoT services. By leveraging ML algorithms in the Perception, Network, and Application layers, IoT systems can achieve improved data perception, efficient communication, and enhanced user experiences. The findings of this research contribute to the advancement of ML-enabled IoT systems, highlighting the importance of considering the usability context in the design and development process.





## BIBLIOGRAPHY

- [1] M. Zorzi, A. Gluhak, S. Lange and A. Bassi, From today's INTRANet of things to a future INTERNet of things: a wireless- and mobility-related view, *IEEE Wireless Communications*, 17, 6, 44–51, 2010.
- [2] F. Firouzi, B. Farahani, M. Weinberger, G. DePace and F. S. Aliee, *IoT Fundamentals: Definitions, Architectures, Challenges, and Promises*, 3–50, Springer International Publishing, Cham, 2020.
- [3] A. Rayes and S. Salam, *Internet of Things from hype to reality*, Springer, 2019.
- [4] L. Atzori, A. Iera and G. Morabito, The Internet of Things: A survey, *Computer Networks*, 54, 15, 2787–2805, 2010.
- [5] G. Barile, A. Leoni, L. Pantoli and V. Stornelli, Real-time autonomous system for structural and environmental monitoring of dynamic events, *Electronics*, 7, 12, 420, 2018.
- [6] M. Wu, T.-J. Lu, F.-Y. Ling, J. Sun and H.-Y. Du, Research on the architecture of Internet of Things, *2010 3rd international conference on advanced computer theory and engineering (ICACTE)*, 5, V5–484, IEEE, 2010.
- [7] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari and M. Ayyash, Internet of things: A survey on enabling technologies, protocols, and applications, *IEEE communications surveys & tutorials*, 17, 4, 2347–2376, 2015.
- [8] M. Lombardi, F. Pascale and D. Santaniello, Internet of things: A general overview between architectures, protocols and applications, *Information*, 12, 2, 87, 2021.
- [9] J. Jamali, B. Bahrami, A. Heidari, P. Allahverdizadeh and F. Norouzi, IoT Architecture, *Towards the Internet of Things*, 11–33, Springer, Cham, 2020.
- [10] P. Sethi and S. R. Sarangi, Internet of Things: Architectures, Protocols, and Applications, *Journal of Electrical and Computer Engineering*, 2017, 9324035, 2017.
- [11] L. Atzori, A. Iera, G. Morabito and M. Nitti, The Social Internet of Things (SIoT) – When social networks meet the Internet of Things: Concept, architecture and network characterization, *Computer networks*, 56, 16, 3594–3608, 2012.
- [12] A. Fahim, M. A. Hasan and M. Chowdhury, Smart parking systems: Comprehensive review based on various aspects, *Heliyon*, 7, 5, e07050, 2021.
- [13] F. Al-Turjman and P. Malekloo, Smart parking in iot-enabled cities: A survey, *2019 IEEE 11th International Conference on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS)*, 1335–1340, IEEE, 2019.

- [14] M. Khalid, K. Wang, N. Aslam, Y. Cao, N. Ahmad and M. K. Khan, From smart parking towards autonomous valet parking: A survey, challenges and future works, *Journal of Network and Computer Applications*, 175, 102935, 2021.
- [15] C. Biyik, Z. Allam, G. Pieri, D. Moroni, M. O’Fraifer, E. O’Connell, S. Olariu and M. Khalid, Smart parking systems: Reviewing the literature, architecture and ways forward, *Smart Cities*, 4, 2, 623–642, 2021.
- [16] K. Mekki, E. Bajic, F. Chaxel and F. Meyer, Overview of cellular LPWAN technologies for IoT deployment: Sigfox, LoraWan, and NB-IoT, *2018 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, 197–202, IEEE, 2018.
- [17] M. Abomhara and G. M. Kjøien, Cyber Security and the Internet of Things: Vulnerabilities, Threats, Intruders and Attacks, *J. Cyber Secur. Mobil.*, 4, 65–88, 2015.
- [18] P. Šolić, A. Leoni, R. Colella, T. Perković, L. Catarinucci and V. Stornelli, IoT-Ready Energy-Autonomous Parking Sensor Device, *IEEE Internet of Things Journal*, 8, 6, 4830–4840, 2021.
- [19] O. Vermesan and J. Bacquet, *Next Generation Internet of Things: Distributed Intelligence at the Edge and Human Machine-to-Machine Cooperation*, 2018.
- [20] I. Hashem, V. Chang, N. Anuar, K. Adewole, I. Yaqoob, A. Gani, E. Ahmed and H. Chiroma, The role of Big Data in Smart City, *Int. J. Inf. Manag.*, 36, 6, 748–758, 2016.
- [21] G. M. Lee and N. Crespi, Shaping future service environments with the cloud and Internet of Things: networking challenges and service evolution, *Leveraging Applications of Formal Methods, Verification, and Validation*, 6415, –, Springer Berlin Heidelberg, 2010.
- [22] S. C. K. Tekouabou, E. A. A. Alaoui, W. Cherif and H. Silkan, Improving parking availability prediction in smart cities with IoT and ensemble-based model, *Journal of King Saud University - Computer and Information Sciences*, 34, 3, 687–697, 2022.
- [23] D. Sarkar, R. Bali and T. Sharma, *Practical Machine Learning with Python: A Problem-Solver’s Guide to Building Real-World Intelligent Systems*, Apress, 2017.
- [24] X. X. Zhu, D. Tuia, L. Mou, G.-S. Xia, L. Zhang, F. Xu and F. Fraundorfer, Deep learning in remote sensing: A comprehensive review and list of resources, *IEEE Geoscience and Remote Sensing Magazine*, 5, 4, 8–36, 2017.
- [25] S. Bhattacharya, S. Somayaji, T. R. Gadekallu and P. K. Maddikunta, A review on deep learning for future smart cities, *Internet Technology Letters*, 1–6, 2020.
- [26] C. Sobin, A survey on architecture, protocols and challenges in IoT, *Wireless Personal Communications*, 112, 3, 1383–1429, 2020.
- [27] S. Nižetić, P. Šolić, D. López-de-Ipiña González-de Artaza and L. Patrono, Internet of Things (IoT): Opportunities, issues and challenges towards a smart and sustainable future, *Journal of Cleaner Production*, 274, 122877, 2020.

- [28] Short Paper: IoT: Challenges, projects, architectures, author=Gazis, V. and others, *2015 18th International Conference on Intelligence in Next Generation Networks*, 145–147, IEEE, 2015.
- [29] L. Patrono, L. Atzori, P. Šolić, S. Nižetić and D. L.-d.-I. Gonz’alez-de Artaza, Challenges to be addressed to realize ‘Internet of Things solutions for smart environments, *Future Generation Computer Systems*, 2019.
- [30] A. Taivalsaari and T. Mikkonen, A Taxonomy of IoT Client Architectures, *IEEE Software*, 2018.
- [31] S. Ullo and G. Sinha, Advances in Smart Environment Monitoring Systems Using IoT and Sensors, *Sensors*, 20, 11, 3113, 2020.
- [32] J. M. Talavera, L. E. Tob’on, J. A. G’omez, M. A. Culman, J. M. Aranda, D. T. Parra, L. A. Quiroz, A. Hoyos and L. E. Garreta, Review of IoT applications in agro-industrial and environmental fields, *Computers and Electronics in Agriculture*, 142, Part A, 283–297, 2017.
- [33] P. Mani Sai Jyothi and D. Nandan, Utilization of the Internet of Things in Agriculture: Possibilities and Challenges, *Soft Computing: Theories and Applications*, 1154, Springer, 2020.
- [34] M. Ayaz, M. Ammad-Uddin, Z. Sharif, A. Mansour and E.-H. M. Aggoune, Internet-of-Things (IoT)-Based Smart Agriculture: Toward Making the Fields Talk, *IEEE Access*, 7, 129551–129583, 2019.
- [35] T. Talaviya, D. Shah, N. Patel, H. Yagnik and M. Shah, Implementation of artificial intelligence in agriculture for optimisation of irrigation and application of pesticides and herbicides, *Artificial Intelligence in Agriculture*, 4, 53–63, 2020.
- [36] A. N. Harun, M. R. M. Kassim, I. Mat and S. S. Ramli, Precision irrigation using wireless sensor network, *2015 International Conference on Smart Sensors and Application (ICSSA)*, 71–75, 2015.
- [37] R. Sui and J. Baggard, Wireless sensor network for monitoring soil moisture and weather conditions, *Applied Engineering in Agriculture*, 31, 193–200, 2015.
- [38] L. D. Rodić, T. Županović, T. Perković, P. Šolić and J. J. Rodrigues, Machine learning and soil humidity sensing: Signal strength approach, *ACM Transactions on Internet Technology (TOIT)*, 22, 2, 1–21, 2021.
- [39] Evaluation of California weather-based “smart” irrigation controller programs, <https://p2infohouse.org/ref/53/52030.pdf>, 2020, accessed: 24 February 2020.
- [40] S. Perković, S. Damjanović, P. Šolić, L. Patrono and J. J. Rodrigues, Meeting Challenges in IoT: Sensing, Energy Efficiency and the Implementation, *Fourth International Congress on Information and Communication Technology in concurrent with ICT Excellence Awards (ICICT 2019)*, 2019.

- [41] S. Nizetić, N. Djilali, A. Papadopoulos and J. R. J., Smart technologies for promotion of energy efficiency, utilization of sustainable resources and waste management, *Journal of Cleaner Production*, 231, 565–591, 2019.
- [42] A. Singh, A. Kumar, A. Kumar and V. Dwivedi, Radio frequency global positioning system for real-time vehicle parking, *2016 International Conference on Signal Processing and Communication (ICSC)*, 479–483, 2016.
- [43] H. Paidi, H. Fleyeh, J. Håkansson and R. G. Nyberg, Smart parking sensors, technologies and applications for open parking lots: a review, *IET Intelligent Transport Systems*, 12, 6, 735–741, October 2018.
- [44] IBM, IBM Survey, <https://www03.ibm.com/press/us/en/pressrelease/35515.wss>, accessed 20 September 2021.
- [45] A. Koster, O. Oliveira, V. Volpato, F. Delvequio and F. Koch, Recognition and recommendation of parking places, *Advances in Artificial Intelligence – IBERAMIA 2014*, 675–685, Springer International Publishing, 2014.
- [46] C. Shoup, The high cost of free parking, *Journal of Planning Education and Research*, 17, 1, 3–20, 1997.
- [47] L. Cui, Z. Zhang, N. Gao, Z. Meng and Z. Li, Radio Frequency Identification and Sensing Techniques and Their Applications—A Review of the State-of-the-Art, *Sensors*, 19, 18, 4012, 2019.
- [48] Y. Santos and E. Canedo, On the Design and Implementation of an IoT based Architecture for Reading Ultra High Frequency Tags, *Information*, 10, 2, 41, 2019.
- [49] M. Škiljo, P. Šolić, Z. Blažević, L. Patrono and J. Rodrigues, Electromagnetic characterization of SNR variation in passive Gen2 RFID system, *2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN)*, 172–175, IEEE, 2017.
- [50] P. Šolić, J. Maras, J. Radić and Z. Blažević, Comparing Theoretical and Experimental Results in Gen2 RFID Throughput, *IEEE Transactions on Automation Science and Engineering*, 14, 1, 349–357, 2017.
- [51] E. Inc., Class1 Generation 2 UHF Air Interface Protocol Standard, Tech. rep., 2015.
- [52] P. Šolić, J. Radić and N. Rožić, Energy Efficient Tag Estimation Method for ALOHA-Based RFID Systems, *IEEE Sensors Journal*, 14, 10, 3637–3647, 2014.
- [53] D. Dobkin, *The RF in RFID*, Elsevier, 2008.
- [54] Z. Zhu, M.-H. Yu and P. Riezebos, A research framework of smart education, *Smart Learning Environments*, 3, 1, 1–14, 2016.
- [55] D. Stojanović, Z. Bogdanović, L. Petrović, S. Mitrović and A. Labus, Empowering learning process in secondary education using pervasive technologies, *Interactive Learning Environments*, 0, 0, 1–14, 2020.
- [56] R. Sami, S. Abbas and Z. Ibrahim, A survey: the role of the internet of things in the development of education, *Indonesian Journal of Electrical Engineering and Computer Science*, 19, 215, 07 2020.

- [57] P. Kuppusamy, SMART EDUCATION ARCHITECTURE USING THE INTERNET OF THINGS (IOT) TECHNOLOGY, *The International Journal of Management Education*, 9, 46–70, 04 2019.
- [58] L. McRae, K. Ellis and M. Kent, Internet of things (IoT): education and technology, *Relatsh. between Educ. Technol. students with Disabil. Leanne, Res*, 1–37, 2018.
- [59] A. Manches, P. Duncan, L. Plowman et al., Three questions about the Internet of things and children, *TechTrends*, 59, 3, 76–83, Mar 2015.
- [60] K. Al-Htaybat, L. von Alberti-Alhtaybat and Z. Alhatabat, Educating digital natives for the future: accounting educators' evaluation of the accounting curriculum, *Accounting Education*, 27, 1–25, 02 2018.
- [61] D. Čoko, L. D. Rodić, T. Perković and P. Šolić, Geometry from Thin Air: Theremin as a Playful Learning Device, *2021 16th International Conference on Telecommunications (ConTEL)*, 89–96, 2021.
- [62] J. S. He, S. Ji and P. O. Bobbie, Internet of Things (IoT)-Based Learning Framework to Facilitate STEM Undergraduate Education, *Proceedings of the SouthEast Conference, ACM SE '17*, 88–94, Association for Computing Machinery, New York, NY, USA, 2017.
- [63] K. Kimmo, Finland: Country report on ict in education, Report, 2017.
- [64] E. Slunjski, The national curricula for early childhood and preschool education, Report, Ministry of Science and Education, Croatia, 2015.
- [65] L. Đujić Rodić and A. Granić, Tangible User Interfaces for Enhancement of Young Children's Mathematical Problem Solving and Reasoning: A Preliminary Review of Relevant Literature, *CECIIS 2018: 29th Central European Conference on Information and Intelligent Systems: Proceedings*, 77–84, Faculty of Organization and Informatics, University of Zagreb, 2018.
- [66] K. Cagiltay, N. Kara and C. C. Aydin, *Smart Toy Based Learning*, 703–711, Springer New York, New York, NY, 2014.
- [67] V. Komis, C. Karachristos, D. Mourta, K. Sgoura, A. Misirli and A. Jaillet, Smart Toys in Early Childhood and Primary Education: A Systematic Review of Technological and Educational Affordances, *Applied Sciences*, 11, 18, 2021.
- [68] P. Frei, V. Su, B. Mikhak and H. Ishii, Curlybot: Designing a New Class of Computational Toys, *CHI '00*, 129–136, Association for Computing Machinery, New York, NY, USA, 2000.
- [69] X. He, T. Li, O. Turel, Y. Kuang, H. Zhao and Q. He, The Impact of STEM Education on Mathematical Development in Children Aged 5-6 Years, *International Journal of Educational Research*, 109, 101795, 2021.
- [70] J. S. Bazargani, A. Sadeghi-Niaraki, F. Rahimi, T. Abuhmed and S.-M. Choi, An IoT-Based Approach for Learning Geometric Shapes in Early Childhood, *IEEE Access*, 10, 130632–130641, 2022.

- [71] H. Tripathy, S. Mishra and K. Dash, Significance of IoT in Education Domain, *International Journal of Computer Science and Mobile Computing*, 10, 59–83, January 2021.
- [72] N. Poursafar, M. E. E. Alahi and S. Mukhopadhyay, Long-range wireless technologies for IoT applications: A review, *2017 Eleventh International Conference on Sensing Technology (ICST)*, 1–6, 2017.
- [73] A. Lavric, A. I. Petrariu and V. Popa, Long Range SigFox Communication Protocol Scalability Analysis Under Large-Scale, High-Density Conditions, *IEEE Access*, 7, 35816–35825, 2019.
- [74] S. Barrachina-Muñoz, T. Adame, A. Bel and B. Bellalta, Towards Energy Efficient LPWANs through Learning-based Multi-hop Routing, *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)*, 644–649, 2019.
- [75] L. G. Kolobe, C. Lebekwe and B. Sigweni, Systematic literature survey: applications of LoRa communication, *International Journal of Electrical and Computer Engineering (IJECE)*, 10, 3176, 06 2020.
- [76] M. Saari, A. M. bin Baharudin, P. Sillberg, S. Hyrynsalmi and W. Yan, LoRa — A survey of recent research trends, *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 0872–0877, 2018.
- [77] L. Oliveira, J. J. P. C. Rodrigues, S. A. Kozlov, R. A. L. Rabêlo and V. H. C. d. Albuquerque, MAC Layer Protocols for Internet of Things: A Survey, *Future Internet*, 11, 1, 2019.
- [78] K. Mekki, E. Bajic, F. Chaxel and F. Meyer, Overview of Cellular LPWAN Technologies for IoT Deployment: Sigfox, LoRaWAN, and NB-IoT, *2018 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, 197–202, 2018.
- [79] J. Ding, M. Nemati, C. Ranaweera and J. Choi, IoT Connectivity Technologies and Applications: A Survey, *IEEE Access*, 8, 67646–67673, 2020.
- [80] E. Zanaj, G. Caso, L. De Nardis, A. Mohammadpour, Alay and M.-G. Di Benedetto, Energy Efficiency in Short and Wide-Area IoT Technologies—A Survey, *Technologies*, 9, 1, 2021.
- [81] W. Ayoub, A. E. Samhat, F. Nouvel, M. Mroue and J.-C. Prévotet, Internet of Mobile Things: Overview of LoRaWAN, DASH7, and NB-IoT in LPWANs Standards and Supported Mobility, *IEEE Communications Surveys Tutorials*, 21, 2, 1561–1581, 2019.
- [82] A. Khalifeh, K. A. Aldahdouh, K. A. Darabkh and W. Al-Sit, A Survey of 5G Emerging Wireless Technologies Featuring LoRaWAN, Sigfox, NB-IoT and LTE-M, *2019 International Conference on Wireless Communications Signal Processing and Networking (WiSPNET)*, 561–566, 2019.
- [83] J. S. Lee and L. E. Miller, *CDMA systems engineering handbook*, Artech House, Inc., 1998.

- [84] Y. Singhal, A. Jain, S. Batra, Y. Varshney and M. Rathi, Review of Bagging and Boosting Classification Performance on Unbalanced Binary Classification, *2018 IEEE 8th International Advance Computing Conference (IACC)*, 338–343, 2018.
- [85] Yanminsun, A. Wong and M. S. Kamel, Classification of imbalanced data: a review, *International Journal of Pattern Recognition and Artificial Intelligence*, 23, 11 2011.
- [86] G. Amato, F. Carrara, F. Falchi, C. Gennaro, C. Meghini and C. Vairo, Deep Learning for Decentralized Parking Lot Occupancy Detection, *Expert Systems with Applications*, 72, 10 2016.
- [87] L. Rabiner and B. Juang, An introduction to hidden Markov models, *IEEE ASSP Magazine*, 3, 1, 4–16, 1986.
- [88] J. Provoost, L. Wismans, S. Van der Drift, A. Kamilaris and M. Van Keulen, Short term prediction of parking area states using real time data and machine learning techniques, *arXiv preprint:1911.13178*, 2019.
- [89] A. C. Müller and S. Guido, *Introduction to Machine Learning with Python*, O’Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472, 2017.
- [90] H. Landaluce, L. Arjona, A. Perallos, F. Falcone, I. Angulo and F. Muralter, A Review of IoT Sensing Applications and Challenges Using RFID and Wireless Sensor Networks, *Sensors*, 20, 9, 2020.
- [91] D. D. Dobkin, *The RF in RFID*, Elsevier, Burlington, MA, USA, 2008.
- [92] M. Škiljo, P. Šolić, Z. Blažević, L. Patrono and J. J. P. C. Rodrigues, Electromagnetic characterization of SNR variation in passive Gen2 RFID system, *2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN)*, 172–175, 2017.
- [93] P. Šolić, J. Maras, J. Radić and Z. Blažević, Comparing Theoretical and Experimental Results in Gen2 RFID Throughput, *IEEE Transactions on Automation Science and Engineering*, 14, 1, 349–357, 2017.
- [94] EPCglobalInc., Class1 generation 2 uhf air interface protocol standard, Tech. rep.
- [95] P. Šolić, J. Radić and N. Rožić, Energy Efficient Tag Estimation Method for ALOHA-Based RFID Systems, *IEEE Sensors Journal*, 14, 10, 3637–3647, 2014.
- [96] C. Law, K. Lee and K.-Y. Siu, Efficient memoryless protocol for tag identification (extended abstract), Association for Computing Machinery, New York, NY, USA, 2000.
- [97] J. Capetanakis, Tree algorithms for packet broadcast channels, *IEEE Transactions on Information Theory*, 25, 5, 505–515, 1979.
- [98] L. D. Rodić, I. Stančić, K. Zovko and P. Šolić, Machine Learning as Tag Estimation Method for ALOHA-based RFID system, *2021 6th International Conference on Smart and Sustainable Technologies (SpliTech)*, 1–6, 2021.
- [99] F. Schoute, Dynamic frame length aloha, *IEEE Transactions on Communications*, 31, 4, 565–568, 1983.

- [100] P. Šolić, J. Maras, J. Radić and Z. Blažević, Comparing Theoretical and Experimental Results in Gen2 RFID Throughput, *IEEE Transactions on Automation Science and Engineering*, 14, 1, 349–357, 2017.
- [101] M. Škiljo, P. Šolić, Z. Blažević, L. D. Rodić and T. Perković, Uhf rfid: Retail store performance, *IEEE Journal of Radio Frequency Identification*, 1–1, 2021.
- [102] W. Chen, An Accurate Tag Estimate Method for Improving the Performance of an RFID Anticollision Algorithm Based on Dynamic Frame Length ALOHA, *IEEE Transactions on Automation Science and Engineering*, 6, 1, 9–15, 2009.
- [103] A. Cutler, D. Cutler and J. Stevens, *Random Forests*, 45, 157–176, 01 2011.
- [104] A. Paul, D. P. Mukherjee, P. Das, A. Gangopadhyay, A. R. Chintla and S. Kundu, Improved Random Forest for Classification, *IEEE Transactions on Image Processing*, 27, 8, 4012–4024, 2018.
- [105] V. Y. Kulkarni and P. K. Sinha, Pruning of Random Forest classifiers: A survey and future directions, *2012 International Conference on Data Science Engineering (ICDSE)*, 64–68, 2012.
- [106] Z. C. Lipton, A critical review of recurrent neural networks for sequence learning, *CoRR*, 2015.
- [107] A. Y. Ng and M. I. Jordan, On Discriminative vs. Generative Classifiers: A comparison of logistic regression and naive Bayes, T. G. Dietterich, S. Becker and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14 [Neural Information Processing Systems: Natural and Synthetic, NIPS 2001, December 3-8, 2001, Vancouver, British Columbia, Canada]*, 841–848, MIT Press, 2001.
- [108] R. Joo, S. Bertrand, J. Tam and R. Fablet, Hidden Markov Models: The Best Models for Forager Movements?, *PLOS ONE*, 8, 8, 1–12, 08 2013.
- [109] G. P. Zhang, Neural networks for classification: a survey, *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 30, 4, 451–462, 2000.
- [110] P. Roßbach, Neural Networks vs. Random Forests—Does it always have to be Deep Learning, *Germany: Frankfurt School of Finance and Management*, 2018.
- [111] J. Amar, "the monte carlo method in science and engineering, 9–19, 03 2006.
- [112] D. P. Kroese, T. Brereton, T. Taimre and Z. I. Botev, Why the Monte Carlo Method is so Important Today, 6, 6, 386–392, Nov. 2014.
- [113] J. Radić, P. Šolić and M. Škiljo, Anticollision algorithm for radio frequency identification system with low memory requirements: Na, *Transactions on Emerging Telecommunications Technologies*, 31, e3969, 04 2020.
- [114] J. Su, Z. Sheng, V. C. M. Leung and Y. Chen, Energy efficient tag identification algorithms for rfid: Survey, motivation and new design, *IEEE Wireless Communications*, 26, 3, 118–124, 2019.



- [115] A. Elsts and R. McConville, Are Microcontrollers Ready for Deep Learning-Based Human Activity Recognition?, *Electronics*, 10, 21, 2021.
- [116] P. Šolić, M. Šarić and M. Stella, RFID reader-tag communication throughput analysis using Gen2 Q-algorithm frame adaptation scheme, *Int. J. Circuits, Systems and Signal Proc.*, 8, 2014.
- [117] Z. P. Sin, P. H. Ng and H. V. Leong, Stuffed toy as an appealing tangible interface for children, *2021 IEEE International Symposium on Multimedia (ISM)*, 75–78, IEEE, 2021.
- [118] Y. Sugiura, C. Lee, M. Ogata, A. Withana, Y. Makino, D. Sakamoto, M. Inami and T. Igarashi, Pinoky: a ring that animates your plush toys, *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 725–734, 2012.
- [119] P. Ihamäki and K. Heljakka, Smart, skilled and connected in the 21st century: Educational promises of the internet of toys (iotoys), *Proceedings of the 2018 Hawaii university international conferences, arts, humanities, social sciences & education, Prince Waikiki Hotel, Honolulu, Hawaii*, 1–19, 2018.
- [120] A. N. Antle and A. F. Wise, Getting down to details: Using theories of cognition and learning to inform tangible user interface design, *Interacting with Computers*, 25, 1, 2013.
- [121] S. Suggate, H. Stoeger and E. Pufke, Relations between playing activities and fine motor development, *Early Child Development and Care*, 187, 8, 1297–1310, 2017.
- [122] A. Bautista, W.-M. Roth and J. S. Thom, Knowing, insight learning, and the integrity of kinetic movement, *Interchange*, 42, 363–388, 2011.
- [123] I. Elia, Observing the use of gestures in young children’s geometric thinking, *Contemporary research and perspectives on early childhood mathematics education*, 159–182, 2018.
- [124] J. S. Thom, (re)(con) figuring space: Three children’s geometric reasonings, *Contemporary research and perspectives on early childhood mathematics education*, 131–158, Springer, 2018.
- [125] B. Verdine, R. Golinkoff, K. Hirsh-Pasek and N. Newcombe, I. spatial skills, their development, and their links to mathematics, *Monographs of the Society for Research in Child Development*, 82, 7–30, 03 2017.
- [126] D.-S. Tran, N.-H. Ho, H.-J. Yang, E.-T. Baek, S.-H. Kim and G. Lee, Real-time hand gesture spotting and recognition using rgb-d camera and 3d convolutional neural network, *Applied Sciences*, 10, 2, 2020.
- [127] D. Avola, L. Cinque, A. Fagioli, G. L. Foresti, A. Fragomeni and D. Pannone, 3d hand pose and shape estimation from rgb images for keypoint-based hand gesture recognition, *Pattern Recognition*, 129, 108762, 2022.
- [128] D. Čoko, I. Stančić, L. Dujčić Rodić and D. Čošić, Theraprox: Capacitive proximity sensing, *Electronics*, 11, 3, 2022.

- [129] C. Cortes and V. Vapnik, Support-vector networks, *Machine learning*, 20, 273–297, 1995.
- [130] L. Breiman, Random forests, *Machine learning*, 45, 5–32, 2001.
- [131] Y. LeCun, Y. Bengio and G. Hinton, Deep learning, *Nature*, 521, 7553, 436–444, 2015.
- [132] J. Bergstra and Y. Bengio, Random search for hyper-parameter optimization., *Journal of machine learning research*, 13, 2, 2012.
- [133] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg et al., Scikit-learn: Machine learning in python, *The Journal of machine Learning research*, 12, 2825–2830, 2011.
- [134] F. Chollet, *Deep learning with Python*, Simon and Schuster, 2021.
- [135] M. Özcan, F. Aliew and H. Görgün, Accurate and precise distance estimation for noisy ir sensor readings contaminated by outliers, *Measurement*, 156, 107633, 2020.
- [136] F. N. Fritsch and R. E. Carlson, Monotone piecewise cubic interpolation, *SIAM Journal on Numerical Analysis*, 17, 2, 238–246, 1980.
- [137] A. Druin, Cooperative inquiry: Developing new technologies for children with children, *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '99, 592–599, Association for Computing Machinery, New York, NY, USA, 1999.
- [138] A. Druin, *Mobile technology for children: Designing for interaction and learning*, Morgan Kaufmann, 2009.
- [139] H. L. O'Brien and E. G. Toms, The development and evaluation of a survey to measure user engagement, *Journal of the American Society for Information Science and Technology*, 61, 1, 50–69, 2010.
- [140] J. C. Read, Validating the fun toolkit: an instrument for measuring children's opinions of technology, *Cognition, Technology & Work*, 10, 119–128, 2008.
- [141] P.-N. Chou, W.-F. Chen, C.-Y. Wu and R. P. Carey, Utilizing 3d open source software to facilitate student learning of fundamental engineering knowledge: A quasi-experimental study, *Int. J. Eng. Educ*, 33, 1, 382–388, 2017.
- [142] Z. Gecu-Parmaksiz and O. Delialioglu, Augmented reality-based virtual manipulatives versus physical manipulatives for teaching geometric shapes to preschool children, *British Journal of Educational Technology*, 50, 6, 3376–3390, 2019.
- [143] O. S. Kesicioglu and M. Mart, The preschool teachers' opinion on teaching geometry., *Southeast Asia Early Childhood*, 11, 2, 21–36, 2022.
- [144] E. V. Laski and R. S. Siegler, Learning from number board games: You learn what you encode., *Developmental psychology*, 50, 3, 853, 2014.
- [145] D. H. Clements and J. Sarama, Early childhood mathematics intervention, *Science*, 333, 6045, 968–970, 2011.

- [146] F. Garcia-Sanjuan, S. Jurdi, J. Jaen and V. Nacher, Evaluating a tactile and a tangible multi-tablet gamified quiz system for collaborative learning in primary education, *Computers & Education*, 123, 65–84, 2018.
- [147] R. M. Yilmaz, Educational magic toys developed with augmented reality technology for early childhood education, *Computers in human behavior*, 54, 240–248, 2016.
- [148] E. Baauw and P. Markopoulous, A comparison of think-aloud and post-task interview for usability testing with children, *Proceedings of the 2004 conference on Interaction design and children: building a community*, 115–116, 2004.
- [149] M. A. Khanum and M. C. Trivedi, Take care: a study on usability evaluation methods for children, *arXiv preprint arXiv:1212.0647*, 2012.
- [150] P. Mertala, Young children’s perceptions of ubiquitous computing and the internet of things, *British Journal of Educational Technology*, 51, 1, 84–102, 2020.
- [151] C. D. Tippett and T. M. Milford, Findings from a pre-kindergarten classroom: Making the case for stem in early childhood education, *International Journal of Science and Mathematics Education*, 15, 67–86, 2017.
- [152] J. Read, S. MacFarlane and C. Casey, Endurability, engagement and expectations: Measuring children’s fun, *Interaction Design and Children*, 01 2009.
- [153] M. Roussou and A. Katifori, Flow, staging, wayfinding, personalization: Evaluating user experience with mobile museum narratives, *Multimodal Technologies and Interaction*, 2, 2, 2018.
- [154] J. Höysniemi, P. Hämäläinen and L. Turkki, Using peer tutoring in evaluating the usability of a physically interactive computer game with children, *Interacting with Computers*, 15(2), 203–225, 04 2003.
- [155] C. Rico-Olarte, D. López, S. Narváez, C. Farinango and P. Pharow, Haphop-physio: a computer game to support cognitive therapies in children, *Psychology Research and Behavior Management*, 10, 209–217, 07 2017.
- [156] P. Markopoulos, J. C. Read, S. MacFarlane and J. Höysniemi, editors, *Evaluating Children’s Interactive Products*, Interactive Technologies, 343–354, Morgan Kaufmann, Burlington, 2008.
- [157] T. M. Mitchell, *Machine Learning*, McGraw-Hill, 1997.
- [158] A. M. Turing, Computing machinery and intelligence, *Mind*, 59, 236, 433–460, 1950.
- [159] S. Angra and S. Ahuja, Machine learning and its applications: A review, *2017 International Conference on Big Data Analytics and Computational Intelligence (ICBDAC)*, 57–60, 2017.
- [160] D. Sarkar, R. Bali and T. Sharma, *Practical Machine Learning with Python: A Problem-Solver’s Guide to Building Real-World Intelligent Systems*, Apress, USA, 1st edn., 2017.
- [161] F. Zantalis, G. Koulouras, S. Karabetsos and D. Kandris, A review of machine learning and iot in smart transportation, *Future Internet*, 11, 4, 94, 2019.

- [162] I. Goodfellow, Y. Bengio and A. Courville, *Deep Learning*, MIT Press, 2016.
- [163] S. Garca, J. Luengo and F. Herrera, *Data Preprocessing in Data Mining*, Springer Publishing Company, Incorporated, 2014.
- [164] M. Mohammed, M. Khan and E. Bashier, *Machine Learning: Algorithms and Applications*, Taylor & Francis, 2016.
- [165] B. Ravindran, Chapter 23 - relativized hierarchical decomposition of markov decision processes, *Decision Making*, 202 of *Progress in Brain Research*, 465 – 488, Elsevier, 2013.
- [166] M. Mohammadi, A. Al-Fuqaha, S. Sorour and M. Guizani, Deep learning for iot big data and streaming analytics: A survey, *IEEE Communications Surveys Tutorials*, 20, 4, 2923–2960, 2018.
- [167] H. Li, K. Ota and M. Dong, Learning iot in edge: Deep learning for the internet of things with edge computing, *IEEE Network*, 32, 1, 96–101, 2018.
- [168] L. Zhang, L. Zhang and B. Du, Deep learning for remote sensing data: A technical tutorial on the state of the art, *IEEE Geoscience and Remote Sensing Magazine*, 4, 2, 22–40, 2016.
- [169] L. Cui, S. Yang, F. Chen, Z. Ming, N. Lu and J. Qin, A survey on application of machine learning for internet of things, *Int. J. Mach. Learn. Cybern.*, 9, 8, 1399–1417, 2018.
- [170] M. A. Khan and J. Kim, Toward developing efficient conv-ae-based intrusion detection system using heterogeneous dataset, *Electronics*, 9, 11, 2020.
- [171] S. C. K. Tekouabou, E. A. A. Alaoui, W. Cherif and H. Silkan, Improving parking availability prediction in smart cities with iot and ensemble-based model, *Journal of King Saud University - Computer and Information Sciences*, 2020.
- [172] V. N. Vapnik, *Statistical Learning Theory*, Wiley-Interscience, 1998.
- [173] L. Breiman, Random forests, *Machine Learning*, 45, 1, 5–32, 2001.
- [174] P. Probst, M. N. Wright and A.-L. Boulesteix, Hyperparameters and tuning strategies for random forest, *WIREs Data Mining and Knowledge Discovery*, 9, 3, e1301, 2019.
- [175] C. Strobl, A.-L. Boulesteix, A. Zeileis and T. Hothorn, Bias in random forest variable importance measures: Illustrations, sources and a solution, *BMC Bioinformatics*, 8, 1, 25, 2007.
- [176] P. A. A. Resende and A. C. Drummond, A survey of random forest based methods for intrusion detection systems, *ACM Comput. Surv.*, 51, 3, May 2018.
- [177] C. Lindner, P. A. Bromiley, M. C. Ionita and T. F. Cootes, Robust and accurate shape model matching using random forest regression-voting, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37, 9, 1862–1874, 2015.

- [178] M. Mahdianpari, B. Salehi, F. Mohammadimanesh, B. Brisco, S. Homayouni, E. Gill, E. R. DeLancey and L. Bourgeau-Chavez, Big data for a big country: The first generation of canadian wetland inventory map at a spatial resolution of 10-m using sentinel-1 and sentinel-2 data on the google earth engine cloud computing platform, *Canadian Journal of Remote Sensing*, 46, 1, 15–33, 2020.
- [179] N. Dogru and A. Subasi, Traffic accident detection using random forest classifier, *2018 15th Learning and Technology Conference (L T)*, 40–45, 2018.
- [180] T. Su and S. Zhang, Object-based crop classification in hetao plain using random forest, *Earth Science Informatics*, 2020.
- [181] R. Doshi, N. Apthorpe and N. Feamster, Machine learning ddos detection for consumer internet of things devices, *2018 IEEE Security and Privacy Workshops (SPW)*, 29–35, 2018.
- [182] T. M. Oshiro, P. S. Perez and J. A. Baranauskas, How many trees in a random forest?, P. Perner, editor, *Machine Learning and Data Mining in Pattern Recognition*, 154–168, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.
- [183] M. Belgiu and L. Drăguț, Random forest in remote sensing: A review of applications and future directions, *ISPRS Journal of Photogrammetry and Remote Sensing*, 114, 24–31, 2016.
- [184] G. A. Fink, *Markov Models for Pattern Recognition - From Theory to Applications*, Advances in Computer Vision and Pattern Recognition, Springer, 2014.
- [185] M. K. Mustafa, T. Allen and K. Appiah, A comparative review of dynamic neural networks and hidden markov model methods for mobile on-device speech recognition, *Neural Comput. Appl.*, 31, S-2, 891–899, 2019.
- [186] B.-J. Yoon, Hidden markov models and their applications in biological sequence analysis, *Current Genomics*, 10, 6, 402–415, 2009.
- [187] N. Raman and S. J. Maybank, Activity recognition using a supervised non-parametric hierarchical HMM, *Neurocomputing*, 199, 163–177, 2016.
- [188] W. Liang, Y. Zhang, J. Tan and Y. Li, A novel approach to ecg classification based upon two-layered hmms in body sensor networks, *Sensors*, 14, 4, 5994–6011, 2014.
- [189] D. A. Coast, R. M. Stern, G. G. Cano and S. A. Briller, An approach to cardiac arrhythmia analysis using hidden markov models, *IEEE Transactions on Biomedical Engineering*, 37, 9, 826–836, 1990.
- [190] G. L. Kouemou, History and theoretical basics of hidden markov models, P. Dymarski, editor, *Hidden Markov Models*, chap. 1, IntechOpen, Rijeka, 2011.
- [191] A. Gron, *Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*, O’Reilly Media, Inc., 1st edn., 2017.
- [192] L. R. Rabiner, A tutorial on hidden markov models and selected applications in speech recognition, *Proceedings of the IEEE*, 77, 2, 257–286, 1989.

- [193] W. Zucchini and I. Macdonald, *Hidden Markov Models for Time Series: An Introduction Using R*, 04 2009.
- [194] X. Xi, Further applications of higher-order markov chains and developments in regime-switching models, 2012.
- [195] L. E. Peterson, K-nearest neighbor, *Scholarpedia*, 4, 2, 1883, 2009, revision #137311.
- [196] H. A. Abu Alfeilat, A. B. Hassanat, O. Lasassmeh, A. S. Tarawneh, M. B. Alhasanat, H. S. Eyal Salman and V. S. Prasath, Effects of distance measure choice on k-nearest neighbor classifier performance: a review, *Big data*, 7, 4, 221–248, 2019.
- [197] J. Brownlee, *Master Machine Learning Algorithms: Discover How They Work and Implement Them From Scratch*, Machine Learning Mastery, v1.1 edn., 2016.
- [198] N. Ali, D. Neagu and P. Trundle, Evaluation of k-nearest neighbour classifier performance for heterogeneous data sets, *SN Applied Sciences*, 1, 12, 1–15, 2019.
- [199] P. C. Sen, M. Hajra and M. Ghosh, Supervised classification algorithms in machine learning: A survey and review, J. K. Mandal and D. Bhattacharya, editors, *Emerging Technology in Modelling and Graphics*, 99–111, Springer Singapore, Singapore, 2020.
- [200] G. Beliakov and G. Li, Improving the speed and stability of the k-nearest neighbors method, *Pattern Recogn. Lett.*, 33, 10, 1296–1301, 2012.
- [201] X. Wu and V. Kumar, editors, *The Top Ten Algorithms in Data Mining*, Chapman and Hall/CRC, 2009.
- [202] T. S. Dipanjan Sarkar, Raghav Bali, *Practical Machine Learning with Python*, Apress, 2018.
- [203] O. I. Abiodun, A. Jantan, A. E. Omolara, K. V. Dada, N. A. Mohamed and H. Arshad, State-of-the-art in artificial neural network applications: A survey, *Heliyon*, 4, 11, e00938, 2018.
- [204] M. H. B. O. D. J. Martin T Hagan, Howard B Demuth, *Neural Network Design (2nd Edition)*, Martin Hagan, 2014.
- [205] Y. LeCun, Y. Bengio and G. Hinton, Deep learning, *Nature*, 521, 436–44, 05 2015.
- [206] S. Hayou, A. Doucet and J. Rousseau, On the impact of the activation function on deep neural networks training, K. Chaudhuri and R. Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, 97 of *Proceedings of Machine Learning Research*, 2672–2680, PMLR, 09–15 Jun 2019.
- [207] G. Bontempi, S. B. Taieb and Y. Le Borgne, Machine learning strategies for time series forecasting, *Business Intelligence - Second European Summer School, eBISS 2012, Brussels, Belgium, July 15-21, 2012, Tutorial Lectures*, 138 of *Lecture Notes in Business Information Processing*, 62–77, Springer, 2012.
- [208] Z. Chen, Y. Liu and S. Liu, Mechanical state prediction based on lstm neural network, *2017 36th Chinese Control Conference (CCC)*, 3876–3881, 2017.

- [209] A. Elhassouny and F. Smarandache, Trends in deep convolutional neural Networks architectures: A review, *2019 International conference of computer science and renewable energies (ICCSRE)*, 1–8, IEEE, 2019.
- [210] Y. LeCun, L. Bottou, Y. Bengio and P. Haffner, Gradient-based learning applied to document recognition, *Proceedings of the IEEE*, 86, 11, 2278–2324, 1998.
- [211] S. Albelwi and A. Mahmood, A framework for designing the architectures of deep convolutional neural networks, *Entropy*, 19, 6, 242, 2017.
- [212] A. Dhillon and G. K. Verma, Convolutional neural network: a review of models, methodologies and applications to object detection, *Progress in Artificial Intelligence*, 9, 2, 85–112, 2020.
- [213] J. Koushik, Understanding convolutional neural networks, *arXiv preprint arXiv:1605.09081*, 2016.
- [214] Z. Li, F. Liu, W. Yang, S. Peng and J. Zhou, A survey of convolutional neural networks: analysis, applications, and prospects, *IEEE transactions on neural networks and learning systems*, 2021.
- [215] K. Simonyan and A. Zisserman, Very deep convolutional networks for large-scale image recognition, *arXiv preprint arXiv:1409.1556*, 2014.
- [216] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke and A. Rabinovich, Going deeper with convolutions, *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1–9, 2015.
- [217] K. He, X. Zhang, S. Ren and J. Sun, Deep residual learning for image recognition, *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778, 2016.
- [218] Y. Cai, W. Zheng, X. Zhang, L. Zhangzhong and X. Xue, Research on soil moisture prediction model based on deep learning, *PloS one*, 14, 4, e0214508, 2019.
- [219] A. Sharma, A. Jain, P. Gupta and V. Chowdary, Machine learning applications for precision agriculture: A comprehensive review, *IEEE Access*, 9, 4843–4873, 2020.
- [220] X. Zhang, R. Li, M. Jiao, Q. Zhang, Y. Wang and J. Li, Development of soil moisture monitor and forecast system, *Transactions of the Chinese Society of Agricultural Engineering*, 32, 18, 140–146, 2016.
- [221] J. Martínez-Fernández, A. González-Zamora, N. Sánchez, A. Gumuzzio and C. Herrero-Jiménez, Satellite soil moisture for agricultural drought monitoring: Assessment of the SMOS derived Soil Water Deficit Index, *Remote Sensing of Environment*, 177, 277–286, 2016.
- [222] A. D. Chukalla, M. S. Krol and A. Y. Hoekstra, Green and blue water footprint reduction in irrigated agriculture: effect of irrigation techniques, irrigation strategies and mulching, *Hydrology and earth system sciences*, 19, 12, 4877–4891, 2015.
- [223] M. Feki, G. Ravazzani, A. Ceppi, G. Milleo and M. Mancini, Impact of infiltration process modeling on soil water content simulations for irrigation management, *Water*, 10, 7, 850, 2018.

- [224] M. Ragnoli, G. Barile, A. Leoni, G. Ferri and V. Stornelli, An autonomous low-power lora-based flood-monitoring system, *J. Low Power Electron. Appl.*, 10, 15, 2020.
- [225] C. Brewster, I. Roussaki, N. Kalatzis, K. Doolin and K. Ellis, Iot in agriculture: Designing a europe-wide large-scale pilot, *IEEE Communications Magazine*, 55, 9, 26–33, 2017.
- [226] T. Perković, S. Damjanović, P. Šolić, L. Patrono and J. Rodrigues, Meeting challenges in iot: Sensing, energy efficiency, and the implementation, *Fourth International Congress on Information and Communication Technology*, 419–430, Springer, 2020.
- [227] A. Sajjad, K. Ajay and S. Haroon, Estimating soil moisture using remote sensing data: A machine learning approach, *Advances in Water Resources*, 33, 69–80, 01 2010.
- [228] Y. Tian, Y. xu and G. Wang, Agricultural drought prediction using climate indices based on support vector regression in xiangjiang river basin, *The Science of the total environment*, 622-623, 710–720, 12 2017.
- [229] J. Rhee and J. Im, Meteorological drought forecasting for ungauged areas based on machine learning: Using long-range climate forecast and remote sensing data, *Agricultural and Forest Meteorology*, 237–238, 105–122, 05 2017.
- [230] M. Gill, T. Asefa, M. Kembrowski and M. McKee, Soil moisture prediction using support vector machines, *JAWRA Journal of the American Water Resources Association*, 42, 1033 – 1046, 06 2007.
- [231] X.-D. Song, G. Zhang, F. Liu, D. Li, Y. Zhao and J. Yang, Modeling spatio-temporal distribution of soil moisture by deep learning-based cellular automata model, *Journal of Arid Land*, 8, 734–748, 05 2016.
- [232] K. G. Liakos, P. Busato, D. Moshou, S. Pearson and D. Bochtis, Machine learning in agriculture: A review, *Sensors*, 18, 8, 2018.
- [233] R. Prasad, R. Kumar and D. Singh, A radial basis function approach to retrieve soil moisture and crop variables from x-brand scatterometer observations, *Progress in Electromagnetics Research B*, 12, 201–217, 01 2009.
- [234] L. Pasolli, C. Notarnicola and L. Bruzzone, Estimating soil moisture with the support vector regression technique, *IEEE Geosci. Remote Sensing Lett.*, 8, 6, 1080–1084, 2011.
- [235] J. Zhang, Y. Zhu, X. Zhang, M. Ye and J. Yang, Developing a long short-term memory (lstm) based model for predicting water table depth in agricultural areas, *Journal of Hydrology*, 561, 04 2018.
- [236] T. Lakhankar, H. Ghedira, M. Temimi, M. Sengupta, R. Khanbilvardi and R. A. Blake, Non-parametric methods for soil moisture retrieval from satellite remote sensing data, *Remote Sensing*, 1, 1, 3–21, 2009.
- [237] H. Wennerström, F. Hermans, O. Rensfelt, C. Rohner and L.- Norden, A long-term study of correlations between meteorological conditions and 802.15.4 link performance, *SECON*, 221–229, 2013.



- [238] J. Luomala and I. Hakala, Effects of temperature and humidity on radio signal strength in outdoor wireless sensor networks, *2015 Federated Conference on Computer Science and Information Systems (FedCSIS)*, 1247–1255, 2015.
- [239] R. V. Aroca, A. C. Hernandez, D. V. Magalhães, M. Becker, C. M. P. Vaz and A. G. Calbo, Calibration of passive UHF RFID tags using neural networks to measure soil moisture, *J. Sensors*, 2018, 3436503:1–3436503:12, 2018.
- [240] A. Hasan, R. Bhattacharyya and S. E. Sarma, Towards pervasive soil moisture sensing using RFID tag antenna-based sensors, *IEEE International Conference on RFID Technology and Applications, RFID-TA 2015, Tokyo, Japan, September 16-18, 2015*, 165–170, IEEE, 2015.
- [241] G. Anagnostopoulos and K. Alexandros, A Reproducible Comparison of RSSI Fingerprinting Localization Methods Using LoRaWAN (datasets), Sep. 2019.
- [242] P. Solic, R. Colella, L. Catarinucci, T. Perkovic and L. Patrono, Proof of Presence: Novel Vehicle Detection System, *IEEE Wireless Communications*, 2019.
- [243] F. Liedmann, C. Holewa and C. Wietfeld, The radio field as a sensor - A segmentation based soil moisture sensing approach, *2018 IEEE Sensors Applications Symposium, SAS 2018, Seoul, South Korea, March 12-14, 2018*, 1–6, IEEE, 2018.
- [244] F. Liedmann and C. Wietfeld, SoMoS - a multidimensional radio field based soil moisture sensing system, *IEEE Sensors*, Oct 2017.
- [245] J. Barriga, J. Sulca, J. León, A. Ulloa, D. Portero, R. Andrade and S. Yoo, Smart parking: A literature review from the technological perspective, *Appl. Sci.*, 9, 21, 4569, 2019.
- [246] A. Khanna and R. Anand, Iot based smart parking system, *2016 International Conference on Internet of Things and Applications (IOTA)*, 266–270, 2016.
- [247] R. Grodi, D. B. Rawat and F. Rios-Gutierrez, Smart parking: Parking occupancy monitoring and visualization system for smart cities, *SoutheastCon 2016*, 1–5, 2016.
- [248] W. Cai, D. Zhang and Y. Pan, Implementation of smart parking guidance system based on parking lots sensors networks, *2015 IEEE 16th International Conference on Communication Technology (ICCT)*, 419–424, 2015.
- [249] M. Hiesmair and K. A. Hummel, Empowering road vehicles to learn parking situations based on optical sensor measurements, *Proceedings of the Seventh International Conference on the Internet of Things, IOT 2017, Linz, Austria, October 22-25, 2017*, 39:1–39:2, ACM, 2017.
- [250] P. Tatulea, F. Calin, R. Brad, L. Brâncovean and M. Greavu, An image feature-based method for parking lot occupancy, *Future Internet*, 11, 8, 169, 2019.
- [251] E. E. Tsiropoulou, J. Baras, S. Papavassiliou and S. Sinha, Rfid-based smart parking management system, *Cyber-Physical Systems*, 3, 1–20, 08 2017.
- [252] X. Bao, Y. Zhan, C. Xu, K. Hu, C. Zheng and Y. Wang, A novel dual microwave doppler radar based vehicle detection sensor for parking lot occupancy detection, 46, 14, 1, 2017.

- [253] R. Prophet, M. Hoffmann, M. Vossiek, G. Li and C. Sturm, Parking space detection from a radar based target list, *2017 IEEE MTT-S International Conference on Microwaves for Intelligent Mobility (ICMIM)*, 91–94, 2017.
- [254] L. Lou, Q. Li, Z. Zhang, R. Yang and W. He, An iot-driven vehicle detection method based on multisource data fusion technology for smart parking management system, *IEEE Internet of Things Journal*, 7, 11, 11020–11029, 2020.
- [255] R. A. Gheorghiu, V. Iordache and V. A. Stan, Urban traffic detectors – comparison between inductive loop and magnetic sensors, *2021 13th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, 1–4, 2021.
- [256] Z. Zhang, X. Li, H. Yuan and F. Yu, A street parking system using wireless sensor networks, *International Journal of Distributed Sensor Networks*, 9, 2013.
- [257] K. Na, Y. Kim and H. Cha, Acoustic sensor network-based parking lot surveillance system, *European Conference on Wireless Sensor Networks*, 247–262, Springer, 2009.
- [258] J. Shi, L. Jin, J. Li and Z. Fang, A smart parking system based on nb-iot and third-party payment platform, *2017 17th International Symposium on Communications and Information Technologies (ISCIT)*, 1–5, 2017.
- [259] C. Lee, Y. Han, S. Jeon, D. Seo and I. Jung, Smart parking system using ultrasonic sensor and bluetooth communication in internet of things, *KIISE Transactions on Computing Practices*, 22, 268–277, 06 2016.
- [260] A. Mackey, P. Spachos and K. N. Plataniotis, Smart parking system based on bluetooth low energy beacons with particle filtering, *IEEE Systems Journal*, 14, 3, 3371–3382, 2020.
- [261] A. Camero, J. Toutouh, D. H. Stolfi and E. Alba, Evolutionary deep learning for car park occupancy prediction in smart cities, *Learning and Intelligent Optimization - 12th International Conference, LION 12, Kalamata, Greece, June 10-15, 2018, Revised Selected Papers*, 11353 of *Lecture Notes in Computer Science*, 386–401, Springer, 2018.
- [262] Y. Zheng, S. Rajasegarar and C. Leckie, Parking availability prediction for sensor-enabled car parks in smart cities, *Tenth IEEE International Conference on Intelligent Sensors, Sensor Networks and Information Processing, ISSNIP 2015, Singapore, April 7-9, 2015*, 1–6, IEEE, 2015.
- [263] A. Klappenecker, H. Lee and J. L. Welch, Finding available parking spaces made easy, *Ad Hoc Networks*, 12, 243–249, 2014.
- [264] W. Zhai, Design of narrowband-iot oriented wireless sensor network in urban smart parking, *International Journal of Online Engineering (iJOE)*, 13, 116, 12 2017.
- [265] T. Ebuchi and H. Yamamoto, Vehicle/pedestrian localization system using multiple radio beacons and machine learning for smart parking, *2019 International Conference on Artificial Intelligence in Information and Communication (ICAIIIC)*, 086–091, 2019.

- [266] P. Seymer, D. Wijesekera and C.-D. Kan, Secure outdoor smart parking using dual mode bluetooth mesh networks, *2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring)*, 1–7, 2019.
- [267] H. Bura, N. Lin, N. Kumar, S. Malekar, S. Nagaraj and K. Liu, An edge based smart parking solution using camera networks and deep learning, *2018 IEEE International Conference on Cognitive Computing (ICCC)*, 17–24, 2018.
- [268] M. Jones, A. Khan, P. Kulkarni, P. E. Carnelli and M. Sooriyabandara, Parkus 2.0: Automated cruise detection for parking availability inference, *Proceedings of the 14th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services, Melbourne, Australia, November 7-10, 2017*, 242–251, ACM, 2017.
- [269] Y. Ji, D. Tang, P. Blythe, W. Guo and W. Wang, Short-term forecasting of available parking space using wavelet neural network model, *IET Intelligent Transport Systems*, 9, 202–209(7), March 2015.
- [270] G. Ali, T. Ali, M. Irfan, U. Draz, M. Sohail, A. Glowacz, M. Sulowicz, R. Mielnik, Z. B. Faheem and C. Martis, Iot based smart parking system using deep long short memory network, *Electronics*, 9, 10, 1696, Oct 2020.
- [271] J. Xiao, Y. Lou and J. Frisby, How likely am i to find parking? – a practical model-based framework for predicting parking availability, *Transp Res Part B Methodol*, 112, 9, 19—39, 2018.
- [272] T. S. Luleseged and G. D. M. Serugendo, Cooperative multiagent system for parking availability prediction based on time varying dynamic markov chains, *Journal of Advanced Transportation*, 2017, 1760842, 2017.
- [273] E. I. Vlahogianni, K. L. Kepaptsoglou, V. Tsetos and M. G. Karlaftis, A real-time parking prediction system for smart cities, *J. Intell. Transp. Syst.*, 20, 2, 192–204, 2016.
- [274] M. Farag, M. Din and H. Elshenbary, Deep learning versus traditional methods for parking lots occupancy classification, *Indonesian Journal of Electrical Engineering and Computer Science*, 19, 964, 08 2020.
- [275] S. Yang, W. Ma, X. Pi and S. Qian, A deep learning approach to real-time parking occupancy prediction in transportation networks incorporating multiple spatio-temporal data sources, *Transportation Research Part C: Emerging Technologies*, 107, 248 – 265, 2019.
- [276] A. B. Baktir and B. Bolat, Determining the occupancy of vehicle parking areas by deep learning, *2020 International Conference on Electrical, Communication, and Computer Engineering (ICECCE)*, 1–4, 2020.
- [277] A. Zacepins, V. Komasilovs and A. Kviesis, Implementation of smart parking solution by image analysis, *VEHITS*, 2018.
- [278] J. Barker and S. u. Rehman, Investigating the use of machine learning for smart parking applications, *2019 11th International Conference on Knowledge and Systems Engineering (KSE)*, 1–5, 2019.

- [279] J. Krieg, G. Jakllari, H. Toma and A. Beylot, Unlocking the smartphone's senses for smart city parking, *2016 IEEE International Conference on Communications (ICC)*, 1–7, 2016.
- [280] S. U. Raj, M. Veera Manikanta, P. S. Sai Harsitha and M. Judith Leo, Vacant parking lot detection system using random forest classification, *2019 3rd International Conference on Computing Methodologies and Communication (ICCMC)*, 454–458, 2019.
- [281] F. M. Awan, Y. Saleem, R. Minerva and N. Crespi, A comparative analysis of machine/deep learning models for parking space availability prediction, *Sensors*, 20, 1, 2020.
- [282] F. Al-Turjman, Mobile Couriers' selection for the Smart-grid in Smart-cities' Pervasive Sensing, *Future Generation Comp. Syst.*, 82, 327–341, 2018.
- [283] T. Lin, H. Rivano and F. Le Mouël, A survey of smart parking solutions, *IEEE Transactions on Intelligent Transportation Systems*, 18, 12, 3229–3253, 2017.
- [284] T. Perković, P. Šolić, H. Zargariasl, D. Čoko and J. J. Rodrigues, Smart parking sensors: State of the art and performance evaluation, *Journal of Cleaner Production*, 121181, 2020.
- [285] F. K. Shaikh, S. Zeadally and E. Exposito, Enabling technologies for green internet of things, *IEEE Systems Journal*, 11, 2, 983–994, 2015.
- [286] R. Arshad, S. Zahoor, M. A. Shah, A. Wahid and H. Yu, Green iot: An investigation on energy saving practices for 2020 and beyond, *Ieee Access*, 5, 15667–15681, 2017.
- [287] C. Zhu, V. C. Leung, L. Shu and E. C.-H. Ngai, Green internet of things for smart world, *IEEE access*, 3, 2151–2162, 2015.
- [288] I. Bose and S. Yan, The green potential of RFID projects: A case-based analysis, *It Professional*, 13, 1, 41–47, 2011.
- [289] H. Vogt, Efficient object identification with passive rfid tags, *International Conference on Pervasive Computing*, 98–113, Springer, 2002.
- [290] P. Šolić, J. Radić and N. Rozić, Linearized combinatorial model for optimal frame selection in gen2 rfid system, *2012 IEEE International Conference on RFID (RFID)*, 89–94, 2012.
- [291] E. Vahedi, V. W. S. Wong, I. F. Blake and R. K. Ward, Probabilistic analysis and correction of chen's tag estimate method, *IEEE Transactions on Automation Science and Engineering*, 8, 3, 659–663, 2011.
- [292] L. Arjona, H. Landaluce, A. Perallos and E. Onieva, Scalable rfid tag estimator with enhanced accuracy and low estimation time, *IEEE Signal Processing Letters*, 24, 7, 982–986, 2017.
- [293] M. Delgado, J. Vales-Alonso and F. Gonzalez-Castao, Analysis of dfsa anti-collision protocols in passive rfid environments, 2610 – 2617, 12 2009.

- [294] J. Vales-Alonso, V. Bueno-Delgado, E. Egea-Lopez, F. J. Gonzalez-Castano and J. Alcaraz, Multiframe maximum-likelihood tag estimation for rfid anticollision protocols, *IEEE Transactions on Industrial Informatics*, 7, 3, 487–496, 2011.
- [295] S. Wang, C. Aggarwal and H. Liu, *Using a Random Forest to Inspire a Neural Network and Improving on It*, 1–9, 06 2017.
- [296] I. E. d. B. Filho, I. Silva and C. M. D. Viegas, An effective extension of anti-collision protocol for rfid in the industrial internet of things (iiot), *Sensors*, 18, 12, 2018.
- [297] J. Su, Z. Sheng, A. X. Liu, Z. Fu and Y. Chen, A time and energy saving-based frame adjustment strategy (TES-FAS) tag identification algorithm for UHF RFID systems, *IEEE Transactions on Wireless Communications*, 19, 5, 2974–2986, 2020.
- [298] L. Arjona, H. Landaluce, A. Perallos and E. Onieva, Dynamic frame update policy for UHF RFID sensor tag collisions, *Sensors*, 20, 9, 2696, 2020.
- [299] Z. Huang, R. Xu, C. Chu, Z. Li, Y. Qiu, J. Li, Y. Ma and G. Wen, A novel cross layer anti-collision algorithm for slotted ALOHA-based UHF RFID systems, *IEEE Access*, 7, 36207–36217, 2019.
- [300] J. Su, Z. Sheng, V. C. Leung and Y. Chen, Energy efficient tag identification algorithms for RFID: survey, motivation and new design, *IEEE Wireless Communications*, 26, 3, 118–124, 2019.
- [301] V. M. Axline, *Play therapy*, Ballantine Books, 1974.
- [302] P. O. Peretti and T. M. Sydney, Parental toy choice stereotyping and its effects on child toy preference and sex-role typing, *Social Behavior and Personality: an international journal*, 12, 2, 213–216, 1984.
- [303] M. Akdeniz and F. Özdiñç, Maya: An artificial intelligence based smart toy for pre-school children, *International Journal of Child-Computer Interaction*, 29, 100347, 2021.
- [304] K. Heljakka and P. Ihämäki, Preschoolers learning with the internet of toys: From toy-based edutainment to transmedia literacy, *Seminar. net*, 14, 85–102, 2018.
- [305] N. Kara and K. Cagiltay, Smart toys for preschool children: A design and development research, *Electronic Commerce Research and Applications*, 39, 100909, 2020.
- [306] K. Ryokai and J. Cassell, Computer support for children’s collaborative fantasy play and storytelling, 1999.
- [307] J. W. Glos and J. Cassell, Rosebud: Technological toys for storytelling, *CHI’97 Extended Abstracts on Human Factors in Computing Systems*, 359–360, 1997.
- [308] N. Kara, C. C. Aydin and K. Cagiltay, Design and development of a smart storytelling toy, *Interactive Learning Environments*, 22, 3, 288–297, 2014.
- [309] D. Merrill, J. Kalanithi and P. Maes, Siftables: towards sensor network user interfaces, *Proceedings of the 1st international conference on Tangible and embedded interaction*, 75–78, 2007.

- [310] S. Price and Y. Rogers, Let's get physical: The learning benefits of interacting in digitally augmented physical spaces, *Computers & Education*, 43, 1-2, 137–151, 2004.
- [311] R. Luckin, D. Connolly, L. Plowman and S. Airey, Children's interactions with interactive toy technology, *Journal of Computer Assisted Learning*, 19, 2, 165–176, 2003.
- [312] M. Resnick, Technologies for lifelong kindergarten, *Educational Technology Research and Development*, 46, 4, 43–55, 1998.
- [313] P. Frei, V. Su, B. Mikhak and H. Ishii, Curlybot: designing a new class of computational toys, *Proceedings of the SIGCHI conference on Human factors in computing systems*, 129–136, 2000.
- [314] L. D. Rodić and A. Granić, Tangible interfaces in early years' education: a systematic review, *Personal and Ubiquitous Computing*, 1–39, 2021.
- [315] S. Björk and J. Holopainen, Games and design patterns, *The game design reader: A rules of play anthology*, 410–437, 2005.
- [316] R. J. Jacob, A. Girouard, L. M. Hirshfield, M. S. Horn, O. Shaer, E. T. Solovey and J. Zigelbaum, Reality-based interaction: a framework for post-wimp interfaces, *Proceedings of the SIGCHI conference on Human factors in computing systems*, 201–210, 2008.
- [317] A. Druin, J. Stewart, D. Proft, B. Bederson and J. Hollan, Kidpad: a design collaboration between children, technologists, and educators, *Proceedings of the ACM SIGCHI conference on human factors in computing systems*, 463–470, 1997.
- [318] M. Liang, Y. Li, T. Weber and H. Hussmann, Tangible Interaction for Children's Creative Learning: A Review, C&C '21, Association for Computing Machinery, New York, NY, USA, 2021.
- [319] S. Schroth, H. Tang, A. Carr-chellman and M. AlQahtani, An exploratory study of osmo tangram and tangram manipulative in an elementary mathematics classroom, *Journal of Educational Technology Development and Exchange (JETDE)*, 11, 1, 1, 2019.
- [320] B. Wohl, B. Porter and S. Clinch, Teaching computer science to 5-7 year-olds: An initial study with scratch, cubelets and unplugged computing, *Proceedings of the workshop in primary and secondary computing education*, 55–60, 2015.
- [321] K. Heljakka, P. Ihamäki, P. Tuomi and P. Saarikoski, Gamified coding: Toy robots and playful learning in early education, *2019 International Conference on Computational Science and Computational Intelligence (CSCI)*, 800–805, 2019.
- [322] M. Kusmin, M. Saar and M. Laanpere, Smart schoolhouse — designing iot study kits for project-based learning in stem subjects, *2018 IEEE Global Engineering Education Conference (EDUCON)*, 1514–1517, 2018.
- [323] S. Papert, *Children, computers, and powerful ideas*, Harvester, 1980.
- [324] J. Piaget, Cognitive development in children development and learning, *Journal of Research in Science Teaching*, 2, 176–186, 1964.

- [325] M. Montessori, *The Montessori Method by Maria Montessori (1870-1952). Translated by Anne Everett George (1882).*, New York: Frederick A. Stokes Company, 1912.
- [326] G. Keren and M. Fridin, Kindergarten social assistive robot (kindsar) for children's geometric thinking and metacognitive development in preschool education: A pilot study, *Computers in Human Behavior*, 35, 400–412, 2014.
- [327] M. U. Bers, The tangiblek robotics program: Applied computational thinking for young children., *Early Childhood Research & Practice*, 12, 2, n2, 2010.
- [328] E. R. Kazakoff, A. Sullivan and M. U. Bers, The effect of a classroom-based intensive robotics and programming workshop on sequencing ability in early childhood, *Early Childhood Education Journal*, 41, 245–255, 2013.
- [329] O. Zuckerman, Designing digital objects for learning: lessons from Froebel and Montessori, *International Journal of Arts and Technology*, 3, 1, 124–135, 2010.
- [330] S.-Y. Lin, S.-Y. Chien, C.-L. Hsiao, C.-H. Hsia and K.-M. Chao, Enhancing computational thinking capability of preschool children by game-based smart toys, *Electronic Commerce Research and Applications*, 44, 101011, 2020.
- [331] A. N. Antle, Exploring how children use their hands to think: an embodied interactional analysis, *Behaviour & Information Technology*, 32, 9, 938–954, 2013.
- [332] J. Yu and R. Roque, A review of computational toys and kits for young children, *International Journal of Child-Computer Interaction*, 21, 17–36, 2019.
- [333] A. Sullivan, M. Elkin and M. U. Bers, Kibo robot demo: engaging young children in programming and engineering, *Proceedings of the 14th international conference on interaction design and children*, 418–421, 2015.
- [334] F. Delprino, C. Piva, G. Tommasi, M. Gelsomini, N. Izzo and M. Matera, Abbot: a smart toy motivating children to become outdoor explorers, *Proceedings of the 2018 International Conference on Advanced Visual Interfaces*, 1–9, 2018.
- [335] P. Putjorn, P. Siriaraya, C. S. Ang and F. Deravi, Designing a ubiquitous sensor-based platform to facilitate learning for young children in thailand, *Proceedings of the 19th International Conference on Human-Computer Interaction with Mobile Devices and Services*, MobileHCI '17, Association for Computing Machinery, New York, NY, USA, 2017.
- [336] L. Zhang, F. Sun, D. Wang and G. Dai, A Tangible Interaction Tool for Children and its application in Storytelling, *Journal of Computer-Aided Design Computer Graphics*, 29, 3, 557–564, 2017.
- [337] S. Meadthaisong and T. Meadthaisong, Smart farming using internet of thing(iot) in agriculture by tangible programming for children, *2020 17th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, 611–614, 2020.

- [338] K. Heljakka, P. Ihamäki, P. Tuomi and P. Saarikoski, Gamified coding: Toy robots and playful learning in early education, *2019 International Conference on Computational Science and Computational Intelligence (CSCI)*, 800–805, 2019.
- [339] K. Heljakka and P. Ihamäki, Ready, steady, move! coding toys, preschoolers, and mobile playful learning, *Learning and Collaboration Technologies. Ubiquitous and Virtual Environments for Learning and Collaboration: 6th International Conference, LCT 2019, Held as Part of the 21st HCI International Conference, HCII 2019, Orlando, FL, USA, July 26–31, 2019, Proceedings, Part II 21*, 68–79, Springer, 2019.
- [340] S.-Y. Lin, S.-Y. Chien, C.-L. Hsiao, C.-H. Hsia and K.-M. Chao, Enhancing computational thinking capability of preschool children by game-based smart toys, *Electronic Commerce Research and Applications*, 44, 101011, 2020.
- [341] O. de Paula Albuquerque, M. Fantinato, J. Kelner and A. P. de Albuquerque, Privacy in smart toys: Risks and proposed solutions, *Electronic Commerce Research and Applications*, 39, 100922, 2020.
- [342] L. Pontes, G. Coutinho, P. C. Hung and B. Yankson, Security in smart toys: a systematic review of literature, *Distributed, Ambient and Pervasive Interactions: 7th International Conference, DAPI 2019, Held as Part of the 21st HCI International Conference, HCII 2019, Orlando, FL, USA, July 26–31, 2019, Proceedings 21*, 28–38, Springer, 2019.
- [343] S. Maričić and J. Stamatović, The effect of preschool mathematics education in development of geometry concepts in children, *Eurasia Journal of Mathematics, Science and Technology Education*, 2018.
- [344] R. Novita, M. Putra, E. Rosayanti and F. Fitriati, Design learning in mathematics education: Engaging early childhood students in geometrical activities to enhance geometry and spatial reasoning, *Journal of Physics: Conference Series*, 1088, 012016, IOP Publishing, 2018.
- [345] J. S. Thom and L. M. McGarvey, The act and artifact of drawing (s): observing geometric thinking with, in, and through children’s drawings, *ZDM*, 47, 465–481, 2015.



## **APPENDIX A**

## Article

# Towards a Machine Learning Smart Toy Design for Early Childhood Geometry Education: Usability and Performance

Lea Dujic Rodić <sup>1,\*</sup>, Ivo Stančić <sup>1</sup>, Duje Čoko <sup>1</sup>, Toni Perković <sup>1</sup> and Andrina Granić <sup>2</sup>

<sup>1</sup> Faculty of Electrical Engineering, Mechanical Engineering and Naval Architecture, University of Split, 32 Ruđera Boškovića, 21000 Split, Croatia; istancic@fesb.hr (I.S.); dcoko@fesb.hr (D.Č.); toperkovic@fesb.hr (T.P.)

<sup>2</sup> Faculty of Science, University of Split, Ruđera Boškovića 33, 21000 Split, Croatia; andrina.granic@pmfst.hr

\* Correspondence: dujic@fesb.hr

**Abstract:** This study presents the design and evaluation of a plush smart toy prototype for teaching geometry shapes to young children. The hardware design involves the integration of sensors, microcontrollers, an LCD screen, and a machine learning algorithm to enable gesture recognition by the toy. The machine learning algorithm detects whether the child's gesture outline matches the shape displayed on the LCD screen. A pilot study was conducted with 14 preschool children to assess the usability and performance of the smart toy. The results indicate that the smart toy is easy to use, engages children in learning, and has the potential to be an effective educational tool for preschool children. The findings suggest that smart toys with machine learning algorithms can be used to enhance young children's learning experiences in a fun and engaging way. This study highlights the importance of designing user-friendly toys that support children's learning and underscores the potential of machine learning algorithms in developing effective educational toys.

**Keywords:** IoT; smart toy; machine learning; early childhood education; geometry; usability; human–computer interaction



**Citation:** Dujic Rodić, L.; Stančić, I.; Čoko, D.; Perković, T.; Granić, A. Towards a Machine Learning Smart Toy Design for Early Childhood Geometry Education: Usability and Performance. *Electronics* **2023**, *12*, 1951. <https://doi.org/10.3390/electronics12081951>

Academic Editors: Mohammad Jafari and Rania Hodhod

Received: 23 March 2023

Revised: 11 April 2023

Accepted: 18 April 2023

Published: 21 April 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

The Internet of Things (IoT) has emerged as a revolutionary technology that connects various devices and systems to a network, allowing them to communicate and exchange data, thus revolutionizing the way we interact with the world around us. The proliferation of the IoT has ushered in a new era of smart and interconnected systems capable of improving efficiency, automating processes, and improving quality of life. This technology has found uses in a variety of industries, including healthcare, agriculture, transportation, smart cities, and energy [1]. In recent years, the integration of IoT in education has been a growing trend, offering innovative solutions for teaching and learning [2]. IoT technology has the potential to create interactive and immersive learning experiences that can improve student engagement, motivation, and learning outcomes due to the low-cost functionalities of smart devices [3]. These devices can collect and analyze data to improve educational quality and help educators make informed decisions [4]. As a consequence, they promote creativity, critical thinking, communication, and collaboration, leading to the development of higher-order thinking skills among learners [5]. Furthermore, the IoT can help bridge the digital divide by providing students with equal access to education regardless of their location or socioeconomic status [6].

Children, in particular, are benefiting from the incorporation of the IoT in education, since their daily activities primarily focus on the manipulation of physical materials such as toys [7]. Various IoT integration methods for child users have been investigated in this regard. For example, a study presented in [8] sought to improve the vocabulary learning of foreign language children by using multimodal cues in a task-based learning system composed of an educational robot and a 3D book powered by the IoT. According

to the findings of the user study, the researchers believed that the use of multimodal cues can improve vocabulary learning for children learning a foreign language. The research presented in [9] focuses on the design and development of an IoT device that teaches children about smart agriculture and the programming of smart farm systems.

Modern-day children are commonly referred to as digital natives, as they have grown up with current technology being ubiquitous and seamlessly integrated into their daily lives [10]. They are known for their natural and intuitive ability to interact with technology and use digital devices effectively. This proficiency has revolutionized the way they learn, resulting in new methods and modalities of knowledge acquisition [11]. One major area that has been impacted by the rise of digital natives is science, technology, engineering, and mathematics (STEM) education [12]. With the growing importance of technology in almost every aspect of our lives, including IoT applications, the demand for skilled professionals in the STEM field has increased significantly. In response, countries around the world, such as in the European Union, are placing a renewed focus on STEM education and revising their school curricula to make it more engaging and relevant for young learners [13,14]. Therefore, to facilitate meaningful and deeper learning in these areas, future IoT educational applications should be specifically designed to promote the development of abstract mathematical concepts [11,15].

In this regard, both scientific research and commercial applications have focused on toys with IoT features such as software and sensors, commonly referred to as smart toys [16]. These toys are characterized by their ability to facilitate two-way interactions between children and toys, using both tangible objects and electronic components. Smart toys offer a unique play experience that differs from traditional toys by providing an interactive environment that promotes general child development [17]. Moreover, as such, they have the potential to aid in the development of thinking and problem-solving skills, particularly in relation to abstract mathematical concepts such as geometry [18]. Although geometry is an essential subject in mathematics, many students struggle to visualize its concepts, which can impede their ability to learn and apply geometric principles effectively in the future [15].

Recent studies emphasize that there are currently limited empirical studies on STEM education in young children [19]. According to a rather novel study, there is little research on how children interact with IoT-based geometry learning systems and how these systems can be effectively integrated into educational settings [20]. In general, additional research is required to evaluate the effectiveness of smart toys in facilitating the learning process [17], while the authors in [21] suggest that the incorporation of such technology has the potential to revolutionize education.

This study introduces a novel approach to early childhood education by designing and evaluating the first prototype of a smart learning toy for preschool geometry education. Section 2 provides a review of the current state of the art in smart toys for STEM education, highlighting their benefits, limitations, and gaps that the present study aims to address. Section 3 outlines the materials and methods used for prototype design, including hardware components and machine learning algorithms for gesture-initiated feedback. The hardware components include sensors to detect movement and position, a microcontroller for data processing, and a speaker for feedback delivery. Machine learning algorithms were tested and utilized to recognize complex gestures that form a particular geometric shape. Section 4 reports the findings of a pilot user study that involved preschool-aged children interacting with the prototype toy in an experiment session. The study aimed to assess the usability, level of engagement, and motor aspects of interactions with the IoT smart toy designed to promote geometry learning among preschool children. Sections 5 and 6 provide a comprehensive discussion and conclusions, respectively, on the design, usability, and performance of the smart learning toy for preschool geometry education. The study's findings contribute valuable insights to the field of education technology, demonstrating the potential of IoT-based learning systems to improve early childhood education.

## 2. State of the Art

The application of technology to toys and its impact on children's interaction with them has become an area of increasing focus for the scientific community due to the vital role that toys play in the development of children [17]. Smart toys, which incorporate digital features such as software or sensors, provide a more interactive environment than traditional toys, fostering the development of cognitive, social, and behavioral skills in children [16,22]. According to toy manufacturers and marketers, the possibilities of using smart and connected toys in education appear to offer rich, interactive, innovative, and mobile learning experiences for preschool children [23]. As such, smart toys have emerged as a promising tool for STEM education in preschool children [24]. For successful STEM education, research has emphasized the importance of improving mathematical skills, programming skills, and problem-solving skills. The design and implementation of technology for learning cannot take place without taking into account the psychological aspects of a child's development that affect their ability to learn and interact with technology, on the one hand, and the pedagogical practices that improve those abilities, on the other [12,15,25]. STEM education for children is based on the principle of introducing them to programming through a high-level language, which was pioneered by Seymour Papert [26] with his development of Logo Turtles. This approach is based on Piaget's theory of cognitive constructivism [27]. In recent decades, educational technology research has been influenced by Piaget's theory of cognitive development and Montessori's educational approach, which emphasize the importance of hands-on learning and the manipulation of objects in the development of logical-mathematical knowledge [27,28]. Studies have shown that physical manipulation plays a critical role in the development of thinking skills, enabling the transition between physical and virtual representations and simplifying abstract concepts for young children [29]. Interactive features such as sound, animation, and movement-initiated feedback can also provide rich contextual information to enhance learning and motivate children to complete tasks successfully [11].

A study presented in [17] provides a review of smart toys from the last 30 years, focusing on toys for children in early childhood and primary school by analyzing and categorizing smart toys based on their technological and educational capabilities. One of the major findings of the study emphasizes that, in recent years, smart toys have focused on special sciences (programming) and some skills of the 21st century (STEM and computational thinking). On the contrary, in the first 20 years, a greater emphasis was placed on cross-disciplinary skills such as collaboration, emotional thinking, symbolic thinking, storytelling, and problem solving. We have adopted the smart toy categorizations from this research. Another novel research study presented in [30] aimed to review 30 computational toys and kits designed for children aged 7 years and under, including physical, virtual, and hybrid kits. Qualitative analysis examined the kits' design, support for exploring computational concepts and practices, participation in projects and activities, and exploration of other domains of knowledge. The study presents design suggestions and opportunities to expand the exploration of computational concepts, new modes of expression, and expanded support for children from underrepresented groups in computing. The findings reveal commonalities between existing kits and suggest ways for designers and researchers to improve the possibilities for children to create, explore, and play with computing.

Smart toys are now being scientifically researched in the STEM context for preschool education. For example, the KIBO Robot Demo is an educational robot designed to teach young children (ages 4 to 8) programming and engineering concepts [31]. The children can program the robot using wooden blocks with barcodes, learning basic programming concepts such as sequencing, loops, and conditional statements. The system has been tested in a variety of settings and has been shown to effectively engage children in programming and engineering. Research presented in [32] focused on the development of a smart toy called ABBOT, designed to motivate children to become outdoor explorers. ABBOT is equipped with sensors that allow it to collect environmental data such as temperature, humidity, and light levels. The device is also designed to encourage children to participate

in outdoor activities and learn about their environment by providing feedback and rewards. The research study presented in [33] described the anthropomorphic design, development, and testing of a prototype called OBSY, which is an observation learning system aimed at facilitating the learning of scientific concepts for primary school children in Thailand. The system consists of a ubiquitous sensor-based device that resembles an octopus with a mobile web application hosted on the device. Sensors attached to the OBSY device collect environmental data, which is then interpreted using the web application accessed through tablet computers. The system was developed through a user-centered design approach and aims to promote learning science in an engaging and interactive way. The study presented in [34] described the design and interactive behavior of a tangible augmented reality toy kit that teaches preschool children about color mixing, mathematics, and geometric 2D–3D shapes. The game allows children to interact with both physical and on-screen objects using touch-screen and AR interactions. The researchers conclude that the game has the potential to improve the learning experience for young children and to promote interest in STEM fields. Through tangible programming, the study presented in [9] investigated the use of IoT technology in the smart farming education of children. It involved creating a tangible programming kit that simulates a smart farming system using sensors and Internet of Things devices. User testing revealed that the kit was effective in promoting engagement and learning in young children and has the potential to improve learning in the fields of agriculture and technology.

Research in [24,35] investigated coding with two commercial smart toy robots, Dash and Botley, as part of playful learning in the context of Finnish early education. The results of our study show how Finnish preschoolers aged 5–6 years approached, conducted, and played coding with the two toy robots. The study's main conclusion was that preschoolers used toy robots with coding affordances primarily in developing gamified play around them by designing tracks for the toys, programming the toys to solve obstacle paths, and competing in player-generated dexterity, speed, and physically mobile play contests.

A rather recent study presented in [36] examined the effects of didactic approaches in guiding early childhood children in learning computational logic and programming concepts. To develop the students' cognitive skills, a teaching methodology was developed that utilizes a commercial smart mBot Arduino robot. mBot is a beginner-friendly educational robot that makes programming and learning robots simple and enjoyable. mBot also helps develop logical thinking and design skills. The study concluded that the developed method enhances learning processes and computational thinking abilities.

In recent years, due to the development of smart toys that are enhanced with the Internet of Things (IoT) and can connect to the Internet, there has been a growing body of research on cyber security and privacy risks of smart toys. The studies presented in [37,38] focused on reviewing major smart toy-related children's privacy risks and the major mitigations to such risks.

Despite requests from the scientific community to investigate how to best incorporate new technology into the formal and informal learning contexts of young children, the design and development periods of new smart toys have not been adequately emphasized, as highlighted by the research in [39]. Therefore, these authors applied a design and development research method to create guidelines for designing and using smart toys for preschool children. The research examined a smart toy developed in a pilot study, held focus group meetings with early childhood teachers, created two prototypes, and tested them with preschool children, teachers, and scholars. The study divided the design guidelines into three categories: content, visual design, and interaction.

Based on the literature, the use of smart toys in preschool education represents a promising approach to fostering STEM skills in young children, and, in that regard, learning geometry at an early age is crucial for the development of spatial reasoning skills. Studies indicate that it is critical to introduce geometry in the preschool period, when the first critical geometrical observations are made [40,41]. In that regard, recent studies on gestures emphasize the body's significance in spatial and geometric reasoning, highlighting the

importance of kinetic movement in the genesis and development of abstract geometrical cognition in early years [42–45].

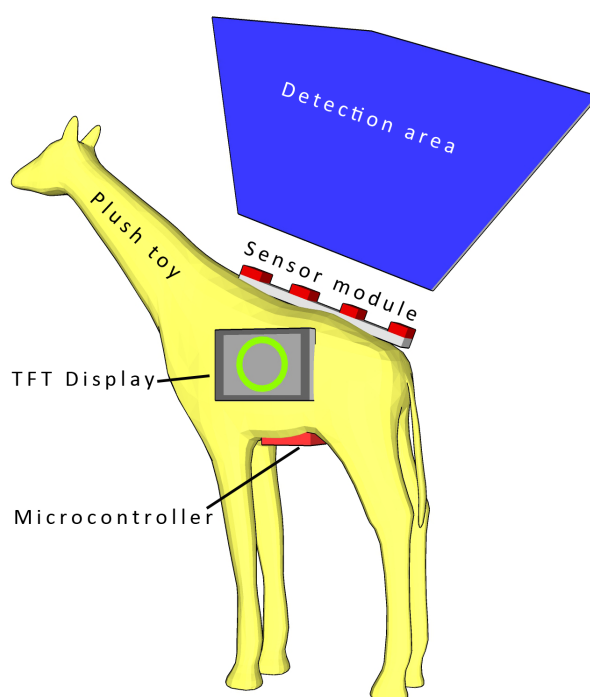
Aligned with the aforementioned rationale, the present study was designed to explicate the design, development, and evaluation of a plush smart toy prototype aimed at facilitating the teaching of geometric shapes to young children. Through the pilot study user evaluation, we intend to investigate the nature of the children’s engagement with the smart toy prototype, to test its feasibility, and to gather some initial data on the toy’s effectiveness. Thus, we will be able to gain valuable insights into the experiences of children as they use the toy and identify potential areas for design improvement.

### 3. Materials and Methods

#### 3.1. Toy and Interaction Design

In order to design a smart toy for learning geometry, we chose to integrate IoT sensing technology in conjunction with appropriate machine learning algorithms into a commercially available plush giraffe toy. This allowed us to take advantage of the softness, familiarity, and flexibility of the design of the plush toy while also providing a dynamic and engaging learning experience for children. The flexibility of the design of the plush toy allows for the seamless integration of IoT technology. Sensors and other electronic components can easily be embedded within the toy while still maintaining the overall aesthetic and feel of the toy. Plush toys are considered soft and safe for children to play with, making them an ideal platform for designing interactive toys. Second, plush toys are often familiar to children, providing a comforting and appealing object to interact with [46,47]. This familiarity can help children form an emotional connection with their smart toy, making the experience more personalized and enjoyable. Research has indicated that animal (plush included) and robot toys are generally regarded as gender neutral, which provided us with an intriguing opportunity to determine whether or not children of different genders prefer one toy over another [48]. Plush toys have been applied and investigated throughout the years of research, as in [39,49–51].

The main hardware components of the smart toy are presented in Figure 1. The specific functions of the components are elaborated in the rest of this section.



**Figure 1.** Main smart toy hardware components.

The proposed interactions with the toy occur in the following manner: Digital representations of geometrical shapes are displayed on the LCD screen, followed by an audible signal. The child is required to map the shape presented on the screen by drawing the shape above the sensor module detection area using hand movements. This allows the child to map an abstract geometric shape from its digital representation on screen into its embodied representation. The machine learning algorithm incorporated in the smart toy detects whether the gesture outline drawn by the child matches the shape presented on the LCD screen. This provides immediate feedback to the child, allowing them to understand if they have correctly identified and drawn the shape. This approach reduces the child's cognitive effort and promotes effortless interaction with the system [52]. The use of hand gestures to interact with the device improves its usability, particularly for young children, and can contribute to the development of fine motor skills. Fine motor skills are increasingly recognized as an important aspect of early childhood development and have been linked to better learning capabilities and overall cognitive development [53]. Current studies on gestures emphasize the role of kinetic movement in the origin and development of abstract geometrical cognition in childhood [42,44,45]. The audio and visual feedback provided by the toy also enriches the learning experience, making it more engaging and enjoyable for the child. Furthermore, this type of activity can also promote the development of spatial skills, which are critical to success in STEM fields such as mathematics and science [54].

### 3.2. Hardware

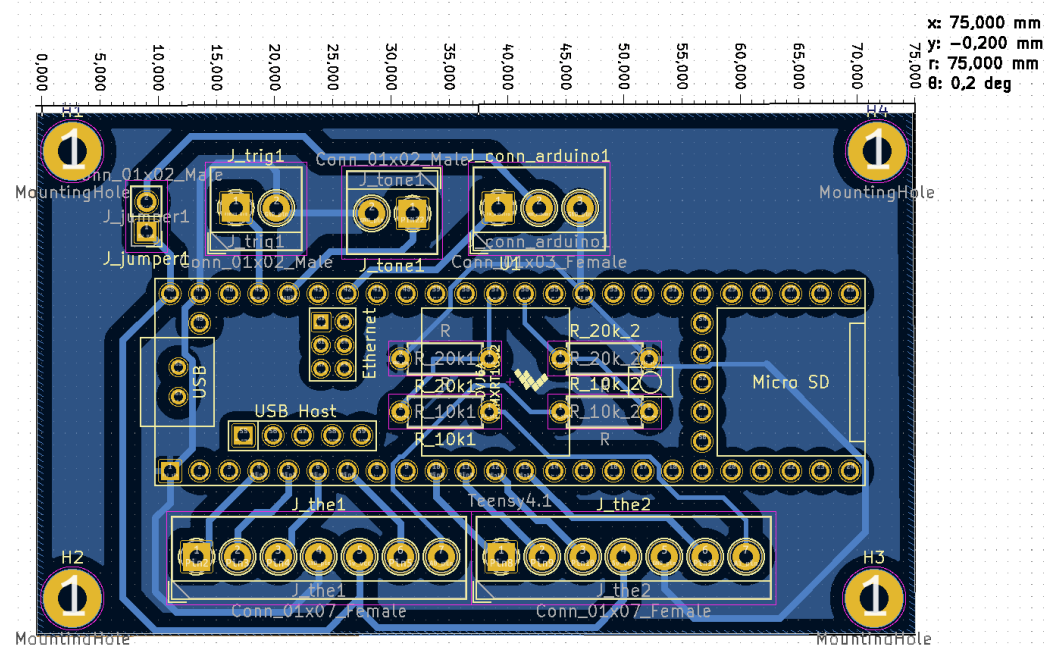
The main functionality of our proposed system is simple in design (as originally planned) and is based on a microcontroller with additional modules attached. We considered several commercially available microcontroller boards for use as the core of the proposed system, where the minimum requirement was the ability to log data onto a microSD card and perform real-time acquisition from the utilized sensors. As most of the considered sensors outputted simple analog signals and did not require any other on-the-fly communication with the microcontroller, any microcontroller board with at least four multiplexed 10-bit A/D inputs would be sufficient for the data acquisition task. In a scenario where four analog sensors are used with 50 readings per second, 400 Bytes would be required for every second of measurement. This led us to a simple calculation that 10 s of acquisition required at least 4 KB of RAM. ATMEGA328P-based microcontroller boards are widely available (used mostly in the Arduino family of microcontroller boards), but only 2 KB of RAM made it nearly impossible to perform real-time acquisition for a prolonged period. As additional data preprocessing was considered (and was finally implemented), together with the possibility of ML inference on the microcontroller itself, a more capable 32-bit microcontroller board was required.

The Teensy 3.6 microcontroller board features an ARM Cortex-M4 MK66FX1M0VMD18 core with 1024 KB Flash and 256 K RAM and clocked at 180 MHz (overclockable at 240 MHz). This computer board is not the fastest microcontroller board available on the market today (even within the Teensy family of microcontroller boards), but it was capable of performing all the planned tasks during the development stage of the proposed system. A compatible pinout with the fastest Teensy 4.1 Development Board (ARM Cortex-M7 at 600 MHz, 7936 K Flash, and 1024 K RAM) made a seamless upgrade possible, if more complex, and resource-hungry ML were to be implemented during future development. Teensy microcontrollers are compatible with the SdFat library (<https://github.com/greiman/SdFat> (accessed on 11 April 2023)) that allows extremely fast file writing, reading, and handling. A SanDisk Class 10 MicroSD card was used for logging the measured data, but any other class 10 microSD card would be sufficient for the task.

As feedback to the user, we implemented both audio and visual components. A piezoelectric speaker (buzzer) could provide limited and short monophonic melodies used for indicating the start and end of the measurement or the error state within the system. Visual feedback was provided using a Newhaven 4.3 inch TFT display with an integrated FTDI FT800 TFT Controller. The display featured a 480 × 272 px resolution, could dis-

play up to 262 K colors, and interfaced with the microcontroller using SPI, which can be clocked up to a 30 MHz clock rate. The advanced library provided by Newhaven ([https://github.com/NewhavenDisplay/FTDL\\_FT800/](https://github.com/NewhavenDisplay/FTDL_FT800/) (accessed on 11 April 2023)) enabled easy integration into the system and the effortless creation of simple geometrical objects to be displayed together with progress bar objects. The extremely fast SPI interfacing did not interfere with the measurement process at all as the TFT content was refreshed only once per second. Additionally, the pushbutton was connected via a long cable to the interrupt-enabled GPIO pin and used as a trigger for the measurement start.

A small PCB breakout board with size  $75 \times 44$  mm was designed using Kicad 6.0 (<https://www.kicad.org/> (accessed on 11 April 2023)) and manufactured using an LPKF ProtoMat S64 CNC machine (<https://www.lpkf.com/en/industries-technologies/research-in-house-pcb-prototyping/products/lpkf-protomat-s64> (accessed on 11 April 2023)), as shown in Figure 2. The PCB secures the microcontroller board and TFT display in place, and provides pins for the easier connection of the sensor module and remote switch.

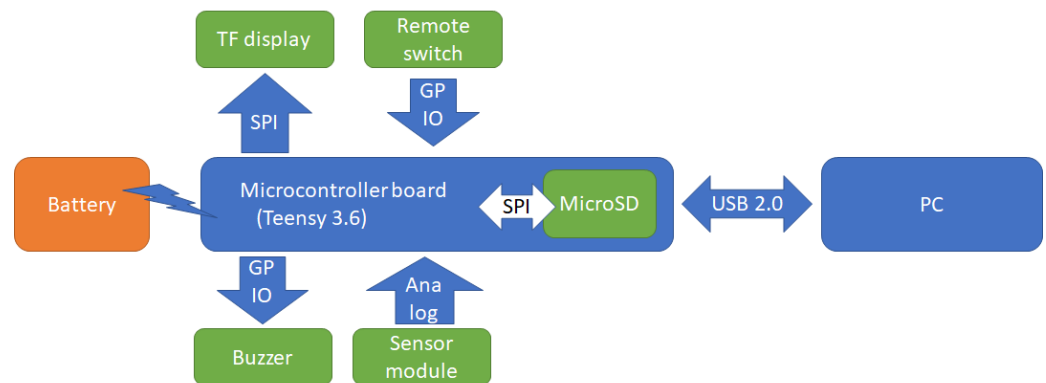


**Figure 2.** Screenshot of the PCB design of an in-house developed breakout board for Teensy 3.6, created by KiCad schematic editor

During the development and testing stage of the system, the microcontroller board was directly connected to the PC using a 480 Mbit/s USB 2.0 interface. This configuration allowed insight into all raw sensor data, more flexibility when testing different ML models, and deeper information on the performance of each ML classification algorithm tested. Model training and inference were performed on a dedicated laptop computer. To be more precise, the machine features an Intel(R) Core(TM) i7-7700HQ@2.80 GHz processor, 16 GB of RAM, and NVIDIA GeForce GTX 1050 Ti CUDA capable graphics card and ran a 64-bit Windows 10 operating system. For more efficient computing with GPU, the NVIDIA CUDA deep neural network library (cuDNN) was applied. All PC-based code was written for Python 3.8 with Tensorflow 2.2.0 on top.

A schematic of all electronic components and interfaces between devices is presented in Figure 3. As the USB 2.0 interface used to connect the examiner PC was relatively long ( $\approx 2$  m), the PC could be placed away from the tested device, thus not obstructing the subject's concentration or interfering with the measurements in any way. In the development stage, the microcontroller board together with all the attached components could be powered via a USB cable or by its own battery source (via a power bank connected directly to the Vin and GND pins).



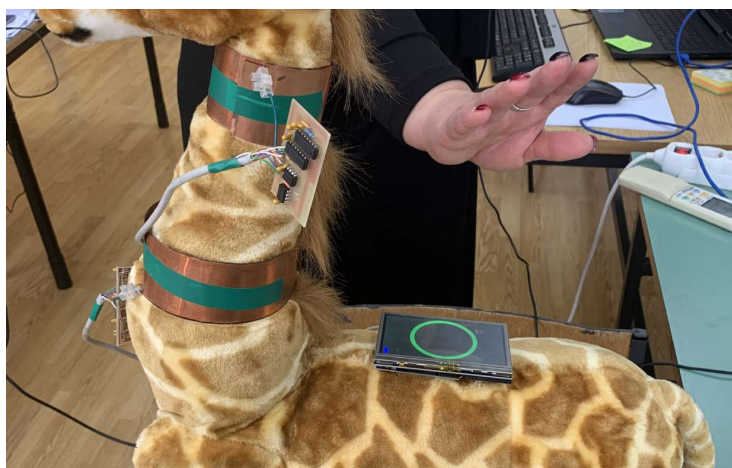


**Figure 3.** Electronic components and interfaces between devices.

### 3.3. Sensing Technology

To locate a hand in space, several possibilities were considered, including visual recognition, capacitive sensors, ultrasonic sensors, TOF sensors, and finally, selected infrared sensors. Our aim, on the hardware side, was to create the simplest possible gesture recognition device that could be completely embedded into a plush toy body and could run data acquisition, ML model inference, and visualization using a single embedded microcontroller. The whole system is based around the microcontroller rather than a single-board computer (such as the Raspberry PI) due to power requirements (longer battery autonomy) and faster boot-up times. Using an RGB (or RGBD) camera as a sensor in real-time was a plausible option that would require a powerful embedded computer to process data in real-time [55,56]. As an alternative, proximity sensors were primarily selected due to their low price, minimal power requirements, and relatively simple 1D output. Ultrasonic distance sensors HC-SR04 (<https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf> (accessed on 11 April 2023)) initially showed themselves to be a valid choice for the proposed system, where their conical-shaped sensing area enabled the detection and distance measurement in large volumes. Unfortunately, the use of several ultrasonic sensors in overlapping sensing areas showed unsatisfactory performance due to interference and achieved a useful acquisition rate lower than 10 Hz. The third type of sensor initially considered was an array of TOF VL53L0X (<https://www.st.com/en/imaging-and-photonics-solutions/vl53l0x.html> (accessed on 11 April 2023)) distance sensors. TOF sensors are superior in tasks involving precise distance measurement and a high refresh rate due to their principle of operation and small sensing area (which is point-sized); the proposed system would require a dense array of sensors to reliably detect and track hand movement. Like the camera in the visual recognition approach, this sensor type was also discarded due to the overall cost and complexity of the system.

An in-house developed capacitive proximity sensor [57] was selected for the initial prototype version of the device. Featuring a  $\approx 10$  cm sensing range and low power consumption, this presented an adequate candidate for gesture sensing. In order to allow for gesture recognition in a two-dimensional plane, a set of two sensors was employed. As presented in Figure 4, two capacitive sensors were mounted on the neck of a plush toy. This arrangement created a kind of virtual canvas, spreading behind the neck and above the back of the plush toy, for users to perform their gestures on. This broadened the number of discernible gestures when compared to a single-sensor scenario.



**Figure 4.** Researcher interacting with the first prototype of the device, featuring capacitive proximity sensors.

One of the most critical parts of the capacitive sensor is its sensing element. The material of which it is made as well as its shape and size extremely determine the sensing range of the device. This is caused by different amounts of ambient capacitance added to the sensing oscillator. This capacitance is compensated for during the calibration procedure by adjusting a digital potentiometer in the reference oscillator. The conductivity of the sensing element greatly influences the charge distribution along its surface. As the exact behavior of the sensor with different sensing elements is impossible to determine, we decided to use an experimental approach by switching materials as well as sizes. In the end we opted for a copper sheet because it allowed for the greatest sensing range. The final size of the sensing element was also experimentally determined in terms of being large enough to provide an adequate sensing range but not introducing an enormous amount of ambient capacitance, which would interfere with the calibration procedure.

The calibration procedure is based on equalizing the frequencies of two oscillators (sensing and reference) while there are no moving objects present within the sensing range of the device. As a result, after a successful calibration, the output voltage from the sensor is at its maximum value. Bringing an object within a sensing range reduces the output voltage in proportion to the distance from the object. If the device operates in a static environment, a single calibration run should be sufficient. By default, the calibration is activated during each power-on or reset sequence. However, if the device needs to be recalibrated for gesture recognition purposes, this can be performed at the user's discretion. Geometrical shapes are displayed on the LCD screen, followed by an audible signal. The user interacts with the toy by making a gesture in the sensing field of the capacitive sensors, thus mapping the presented shape. This interaction produces two time-series vectors (one for each sensor) that are stored on the microSD card.

Another sensor that was considered was an infrared beam sensor, particularly a Sharp GP2Y0A21YK0F Analog Distance Sensor ([https://global.sharp/products/device/lineup/data/pdf/datasheet/gp2y0a21yk\\_e.pdf](https://global.sharp/products/device/lineup/data/pdf/datasheet/gp2y0a21yk_e.pdf) (accessed on 11 April 2023)). This sensor can obtain measurements up to 80 cm. There are a few similar models that are electrically compatible but have different ranges, such as GP2Y0A21YK0F, which works up to a 30 cm distance. All the aforementioned sensors are analog, which means that they yield a signal roughly in the range of 0–5 V, which can be read using the microcontroller's integrated AD converter. The relationship of the measured distance and the output analog signal is not linear, thus recalculations must be performed to obtain the exact distance. Additionally, the IR-type distance sensors have relatively large minimal measurement distances (4 cm for GP2Y0A21YK0F model and 10 cm for GP2Y0A21YK0F model), where readings follow different non-linear relations and thus are unusable. The sensor was composed of an IR LED emitter that projects a light beam and a receiver in the form of a simple 1D camera

that measures the reflected light that is returned from the object. Since the sensor measures the light reflected by the object, it may be affected by the environmental lighting conditions. Another aspect that must be considered to obtain a reliable measure is the internal update period of approximately 40 ms, where the sensor outputs faulty readings during a short period of recalculation. A solution was proposed of increasing the system refresh rate to 50 Hz and data preprocessing, as described in Section 3.5. The 30 cm and 80 cm IR sensors were tested in real scenarios, and the readings were compared to select the optimal solution. The 30 cm version had a shorter minimal distance, thus users' hands can be closer to the sensor module, but a shorter maximum distance also showed in practice that some gestures performed in the larger area over the sensor module can be misinterpreted. On the contrary, the 80 cm sensor version has a longer minimal distance; consequently, the distance to an object closer to a sensor module is misinterpreted. A longer maximum distance allowed the sensor to track gestures performed in larger volumes and was thus selected as optimal (but not perfect) for our system.

#### 3.4. Data Collection

The scientific literature has extensively investigated the use of machine learning models for complex hand gesture recognition, and various approaches have been proposed for conducting preliminary testing. Hand gestures are an important part of nonverbal communication with other humans and are an integral part of interaction with the environment [58]. They are characterized by trajectories of the hand key points in the space and can be recorded by a variety of devices, which can be divided into two types: wearable and non-wearable. Wearable devices use miniature body-borne computational and sensory components, such as various inertial sensors placed on hand key points [59] or glove-like devices that can even track complex finger movements [60]. These types of devices require wearing cumbersome equipment or cables that connect the device to a computer and require preparation before use. Non-wearable devices are commonly vision-based devices [61,62] or employ simple proximity/distance sensors to track the location of the hand in space [63,64]. The main drawbacks of most vision-based systems are their inability to track hand position beyond the camera's field of view, their sensitivity to challenging lighting conditions (in outdoor applications), and their computational complexity. Time-of-flight (TOF) cameras are special types of cameras that measure the distance to a large number of points in space and are commonly employed as input devices to game consoles, where they can track hand movements and detect some specific gestures [65,66]. Both devices are complex and require a computer instead of a microcontroller to read and process measured data. The development of small and simple HCI systems based on proximity and distance sensors using relatively inexpensive components has created new opportunities for novel and cost-effective human-computer interface designs [67]. A similar approach is considered in our research, where the developed sensor module relies on an array of simple and inexpensive distance measurement sensors.

Building accurate and robust models for complex hand gesture recognition is challenging due to the diversity and complexity of hand gestures. Therefore, preliminary testing of machine learning models with collected data is critical to ensuring their reliability and effectiveness.

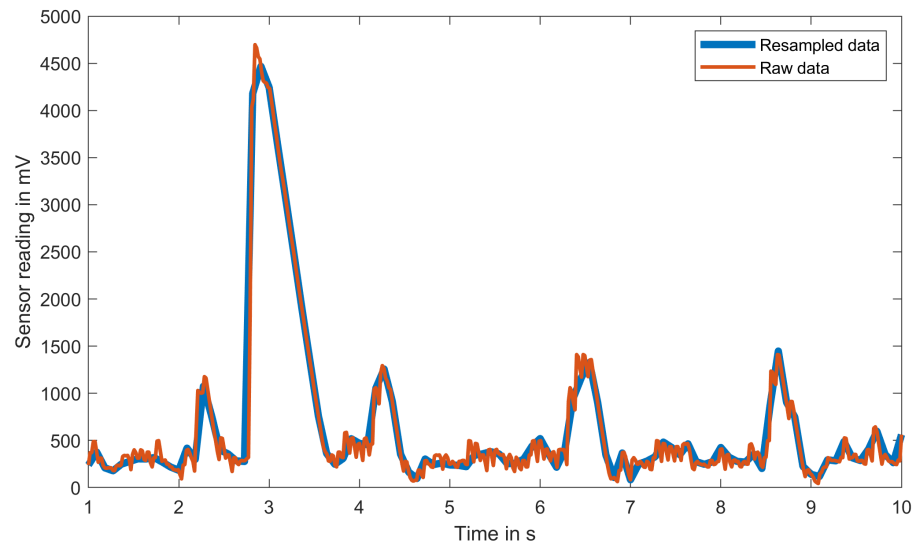
Data were collected from eight adult individuals to serve as data for building a machine learning model. The research employed a non-probability sampling method known as convenience sampling, which entails selecting study participants who are easily accessible and willing to participate. In this case, those were academic staff involved in the research project on a wider scope. All subjects signed an informed consent form in accordance with the Declaration of Helsinki and approved by the Ethics Committee of the Faculty of Electrical Engineering, Mechanical Engineering, and Naval Architecture. Each individual performed gesture movements for around an hour. In general, around 200 gestures (depending on the sensing technology) were gathered per individual and were later processed, depending on the sensing technology.

### 3.5. Data Preprocessing

Raw sensor data were pre-processed before input to ANN to improve the accuracy and efficiency of the machine learning models. Due to the nature of the sensors used, the raw data are noisy and inconsistent, making it difficult to extract meaningful information. Data preprocessing helps to address these issues and prepares the data for analysis through data transformation, data cleaning, and data reduction. Data transformation converts data into a more suitable format by performing linear or non-linear scaling and normalization of numerical values. As a notable example, the IR distance sensor outputs non-linear analog data that could be transformed into a linear distance [68].

By performing non-linear scaling before feeding data to an ML model, the first layers of our machine learning model do not have to find relations between non-linear voltage input and actual linear distance and can focus on resolving hand gestures from transformed linear distance data. The data cleaning technique removes or corrects errors and inconsistencies and predicts missing values. This requirement is again presented on the IR distance sensor, which internally updates readings with a 25 Hz refresh rate while our system is set to a fixed 50 Hz refresh rate. The faster refresh rate was required as the IR distance sensor outputs faulty readings during short periods of internal distance recalculation, and there is the possibility of reading the sensor output during that exact moment. By having more readings than required, simple data filtering can be performed, and outliers are simply removed and replaced with mean neighboring values (using a mean filter). The data reduction technique effectively reduces the size of the dataset while still preserving important information. As reported in the literature, human self-paced movements are within the 3.3 Hz bandwidth (ref), thus the system's 50 Hz sampling rate is excessive for recognizing complex hand gestures. Additionally, the training and inference times of any ML model are significantly reduced by reducing the input size. By our conservative estimation and general experience, a 10 Hz refresh rate was selected as optimal as it balanced the performance and complexity of the ML model. Data reduction was performed by resampling 500 inputs per sensor (for a 10 s measurement time) to 100 inputs using cubic spline interpolation. When data are resampled at a five-fold lower rate, noisy sensor inputs are filtered, and readings are smoothed, as shown in Figure 5. By resampling the data to a 1:5 rate, we effectively achieved low-pass filtering and simplification (reduction) of the ML model. With this approach, we effectively reduced the 50 Hz sensor acquisition rate to a 20 Hz acquisition rate, which is still suitable to recognize complex hand gestures. If a lower acquisition rate were to be used, some faster movements may be tracked with an inadequate number of samples, thus preventing accurate recognition. Additionally, when the original input vector ( $4 \times 500$  samples) is used for ML training with a similar ML model (only the input size was modified), the categorical accuracy of the test is significantly reduced to 0.86 and the model size to around 6.3 MB (1.4 MB for resampled inputs), which may be inadequate for ML implementation on microcontrollers.

An additional pre-processing step was also considered, where only data belonging to the performed gesture are extracted and forwarded to an ML model. This is usually conducted by observing the first and the last samples, where the object is detected by sensors, and extracting all the samples in between. This approach was shown to be unreliable in practice, as the subject may place the hand in the sensed area long before or keep it there long after the required gesture is performed. An example of movement is depicted in Figure 5, where some readings exist throughout the measurement time and do not represent the gesture performed. Thus, an alternative approach was considered, where the complete measurement is forwarded to an ML model with the task of recognizing which gesture was executed at any moment during the allowed measurement time. This was achieved by relying on 1D convolution, which is described in the next section.



**Figure 5.** Comparison of RAW sensor reading and pre-processed data for one sensor during a 10 s measurement time.

### 3.6. Machine Learning: Test and Training

Machine learning has a wide range of applications across various industries and research fields. Some of the most common applications of machine learning today are image and speech recognition, natural language processing, autonomous vehicles and robotics, the Internet of Things, and predictive analytics [69–72]. As technology advances and data become more abundant, the use of machine learning is expected to increase and be implemented in almost every aspect of life. There are many microcontrollers available today that have enough processing power and memory to run machine learning algorithms, with benefits including reduced latency, lower power consumption, and improved privacy and security [73,74].

An artificial neural network (ANN) is a type of machine learning model inspired by the biological structure and functioning of the human brain. It consists of interconnected processing nodes (neurons) that work together to solve a specific problem. Neural networks are typically arranged in a series of interconnected layers, where each layer is made up of a set of neurons that perform a specific mathematical function on the received input. A typical fully connected ANN consists of several types of ANN layers, including input layers, hidden layers, and output layers. Input layers receive input data, (several) hidden layers perform computations where the output of one hidden layer is then passed as input to the next layer, and finally, the output layer produces the final output. Biological neural networks exhibit similar architecture and learning methods for a variety of tasks, where, due to computer power constraints and training time limitations, ANN architectures must be optimized for a specific task. This is usually performed by combining different types of layers into a specific architecture, such as feedforward neural networks, convolutional neural networks, and recurrent neural networks. Our ANN is tasked with classifying hand gestures, and the network input is data acquired from the sensor module. All tested sensor modules are quite similar and perform distance measurements to the subject hand with different properties, as discussed in Section 3.3.

#### Preliminary ML Results

In order to select the most appropriate sensing technology, the collected data from all sensors were first tested by applying a fully connected neural network constructed using the Keras API with TensorFlow as the backend. The training data were pre-processed and normalized. The architecture of the preliminary NN consisted of a sequence of layers that were stacked on top of each other, starting with an input layer, followed by three

hidden layers and an output layer. The first hidden layer contained 1024 neurons, with the activation function used in this layer being a rectified linear unit (ReLU). The second hidden layer had 512 neurons and also used the ReLU activation function. The third hidden layer included a dropout layer, used to prevent over-fitting. The dropout rate was set to 0.2, which means that 20% of the randomly selected neurons in this layer were ignored during each training iteration. The fourth hidden layer had 64 neurons and used the ReLU activation function. The final layer was the output layer, which had three, four, or five neurons (depending on the number of different gestures we tested). It used the softmax activation function. The softmax function is used to output a probability distribution over the 3–5 possible output classes, where the highest probability corresponds to the predicted class label. Due to the fact that this is a multi-class classification problem, for this experiment, the categorical cross-entropy loss function was applied as the loss (cost) function and adaptive moment optimization (Adam) as the optimizer.

The highest accuracy of classification was obtained for three gestures, namely, the circle, square, and pentagon. For each of the sensing technologies, the accuracy of the preliminary NN model is presented in Table 1.

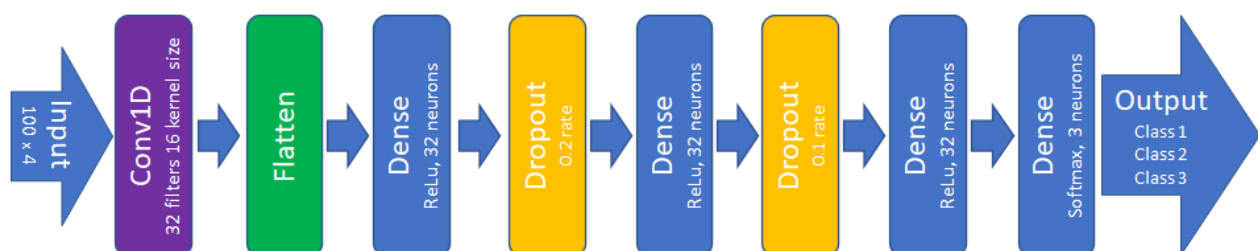
**Table 1.** Classification accuracy of the preliminary neural network model for a particular sensing technology.

Sensing Technology	Neural Network Accuracy
Capacitive	72.05%
IR short range	78.33%
IR long range	95.06%

Based on these results, the IR long-range data were further employed for machine learning utilization in this research.

### 3.7. The Final Architecture of Machine Learning

The final architecture of the NN model displayed in this investigation is constructed of eight layers, as depicted in Figure 6. The first is the input to a 1D convolution layer with 32 filters and 16 kernel sizes. The convolution layer is followed by a flattening, which is then followed by a dense layer with 32 neurons. The dense layer is followed by a dropout layer with a dropout rate of 0.2 and another dense layer with 32 neurons, which is again followed by another dropout layer with a dropout rate of 0.1. The last two layers are the dense layer with 32 neurons and the final output layer. The applied activation functions were ReLU (in dense layers) and softmax (in the output layer). Our proposed architecture is a 1D convolutional neural network, where the convolution layer extracts characteristic features from the signal input and where dense layers try to find relations between the extracted features to classify signals. The dropout rate (probability of setting output from the hidden layer to zero) must be included because of the small training dataset, which prevents the overfitting of the network to a training dataset.



**Figure 6.** ANN architecture.

Since the classification of the hand gestures is a multiclass classification problem, the categorical cross-entropy loss function was applied as the loss function. Another key aspect of the ANN model architecture that was thoroughly examined is the selection of optimizers, learning rates, number of epochs, and batch size. Adam provided the most accurate estimation results on the test dataset, with a 0.0005 learning rate. Optimal training results were obtained with 100 epochs and a 128 batch size.

ML models were originally trained in all shapes, but the performance of the test dataset showed unsatisfactory results with a categorical precision of 87.3%. By removing one gesture from the training and test datasets, the categorical accuracy with the remaining four gestures increased to 89.8%, which was also unsatisfactory. Finally, by keeping only three gestures (namely, circle, square, and pentagon), we achieved better categorical accuracy. After performing several repetitions of the classification, the accuracy ranged from 93.8% to 98.3%, depending on the repetition. The results in the form of a confusion matrix for all three models are presented in Figure 7, and do not show which shape or gesture is to blame for the poor performance of the model with the five gestures in the training dataset.

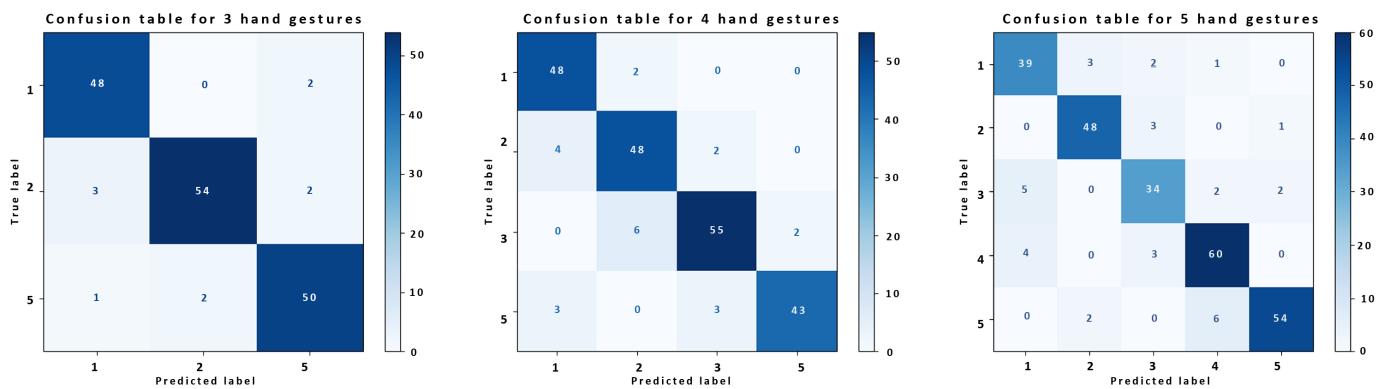
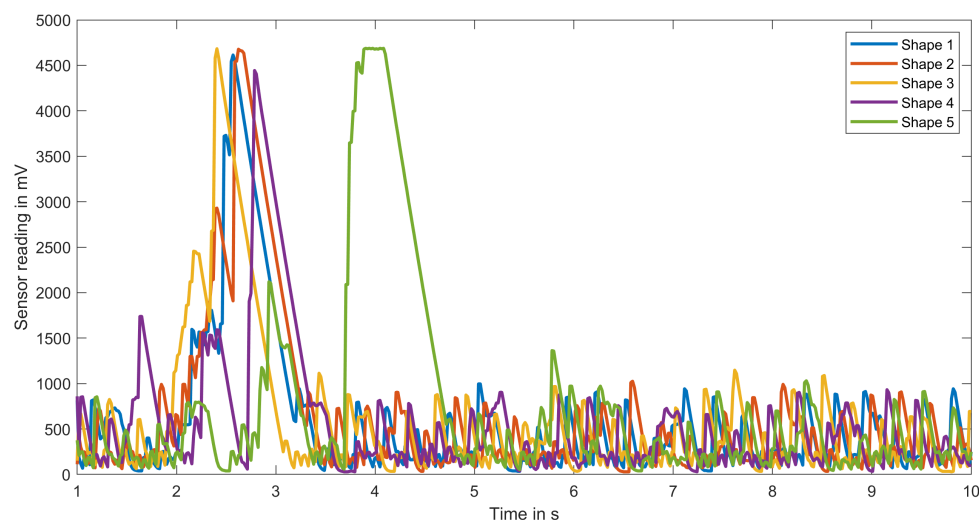


Figure 7. Confusion matrix for models that include three shapes (left), four shapes (middle), and five shapes (right).

We analyzed raw training data from different subjects in search of a solution that could eventually improve performance. An example of the analysis is presented in the form of a plot in Figure 8 for all five gesture-shapes for a single sensor and the same subject. As the system captures raw data in 10 s intervals, useful data (when the user is performing a gesture) take only a few seconds and can be found anywhere inside the original signal. As seen from the sample data presented in Figure 8, useful data take only 2 s intervals per sensor, while the rest of the data are extremely noisy. Relative timings and shapes of slopes between sensors capturing the same gesture are actual features that have to be extracted and used for gesture recognition and classification. By visual inspection of the raw data for several examples (the same person performing the same gesture), some obvious similarities between signals cannot easily be found. Thus, this non-trivial task was delegated to our proposed ML model, which can extract those features and decide which gesture is performed. A more detailed analysis of measured raw data from all four sensors on several subjects in the training set suggested that shapes 3 and 4 (namely, the triangle and rhombus) are similar to shape 1 (circle). We presume that acquiring more training data would improve the performance of a five-shape ML model by allowing it to find more specific features for each shape and consequently build a better model. Due to the aforementioned reasons, we removed shapes 3 and 4 from the training and test datasets.



**Figure 8.** Raw sensor data for five hand gesture shapes recorded with a single sensor.

#### 4. Exploratory Pilot Study

The purpose of this pilot user evaluation was to collect data on children's experiences and perceptions of using IoT technology for educational purposes and to identify potential areas for improvement. The evaluation concentrated on some aspects of usability, levels of engagement, and motor aspects of interactions with the proposed smart toy for early childhood geometry education.

Exploratory pilot studies with children are a crucial step in identifying potential issues in usability testing before conducting a larger usability study. The importance of pilot testing and small sample sizes in child-related research has been highlighted in several studies. To design new technologies for children, Druin [75] used cooperative inquiry with children and found that pilot testing was crucial in refining the design of the technologies. Similarly, in [76], Druin emphasized the importance of pilot testing when designing mobile technology for children and highlighted the need to involve children in the design process. A group size of five–ten participants is a sensible baseline range for usability studies related to problem discovery, as discussed in [77]. Small sample sizes in exploratory pilot studies can also be useful for identifying design flaws or other issues that might not be apparent in larger-scale studies. The small sample size allows for more iterative design processes, which can lead to better user engagement with technology [78]. Additionally, scientific references support the use of small sample sizes in exploratory pilot studies with child participants. For example, in their study of toddlers' use of visual information from video to guide behavior, Schmitt and Anderson [79] used a sample size of 16 children, which allowed for detailed observations of individual children's behaviors and provided rich data for exploring how visual information influences children's actions. Faulkner's research, presented in [80], found that a group size of 10 participants will likely reveal a minimum of 82% of the problems. Nielson in [81] also noted that elaborate usability tests are a waste of resources and that the best results come from testing no more than five users and running as many small tests as possible.

This study was exploratory in nature, and the objective was to test the feasibility and gather preliminary data before conducting a larger and more rigorous study.

The evaluation involved a small group of children aged 4 to 7 years old, who were given the opportunity to interact with the toy and provide feedback on their experiences. The initial idea was to examine the movements of the child's hands while interacting with the prototype toy. The results of the testing provide design guidelines for future interaction realization and movement-initiated feedback. The collected data and findings will be useful



for our future research into the design of the toy as well as the performance of the machine learning model.

In order to conduct the pilot study, the faculty Ethics Committee gave their positive opinion on the experimental procedure, stating that the proposed scientific research would be carried out in accordance with the ethical principle of scientific integrity. All parents signed a consent form before their children participated in the experiment.

#### 4.1. Experiment Design and Procedure

This user evaluation study aimed to assess the usability, levels of engagement, and motor aspects of interactions with an IoT smart toy designed to promote geometry learning among preschool children. The study used a mixed-methods approach, combining quantitative data from pre- and post-test tasks and usability testing with qualitative and quantitative data from video recordings, questionnaires, and interviews with children.

Based on our analysis of the scientific literature, it was found that most of the studies combined a few techniques: an interactive Cyberheroes e-book was evaluated with structured interviews and questionnaires and engaged eight children aged 7 to 9 years [34]; a tangible, interactive learning tool, CyberPLAYce, was assessed with observations, surveys, questionnaires, and audio/video recordings and involved eleven 11- to 12-year-old children [82]; Word Mania, a fun educational game app for children, was evaluated with a Fun Toolkit v3 instrument and enrolled twelve children aged 4 to 9 years [83]; a study looking at STEM in early childhood education involved 14 pre-kindergarten children and used semi-structured interviews, focus groups, and a questionnaire [84]. Hourcade and colleagues [85] recommend using age-appropriate language, providing clear and concise instructions, and using visual aids to support comprehension. Child-friendly data collection methods, such as observation, video recordings, and non-intrusive sensors, can help to minimize disruption and enhance engagement [86].

The study was carried out in a controlled laboratory setting, with one-on-one interaction between the researcher, participants, and the proposed smart toy. The experimental design, along with materials and methods, is further described.

The assessment process was based on a set of criteria that includes several quantitative and qualitative measures, which are expressed in terms of:

- Time-related aspects of interaction (time taken by the user to draw a shape and overall interaction duration);
- Hand gestures used to interact with the toy;
- Perceived ease of use (mapping of the particular shape);
- User mapping accuracy per particular shape;
- Engagement;
- Returnance (as one of the durability dimensions);
- Fun and subjective satisfaction;
- Obtained knowledge.

Several measuring instruments were used to acquire the aforementioned quantitative and qualitative measures:

- Pre-test and post-test tasks: employed to evaluate the level of information acquisition as an indicator of the educational value.
- Attitude questionnaires (Smileyometer and the Again-Again table) [87]: used to measure children's fun and subjective satisfaction.
- Structured interview: used as an instrument to measure children's fun and subjective satisfaction, level of engagement, and their perceived ease of use (mapping of the particular shape).
- Video recording: used as an instrument to measure motor aspects of interaction (hand gestures), time-related aspects of interaction, and engagement.

- Observation checklist: used as an instrument during the assessment process to record notes, document identified problems, and fill in additional information related to task completion accuracy.

Figure 9 represents the overall framework of the experiment.

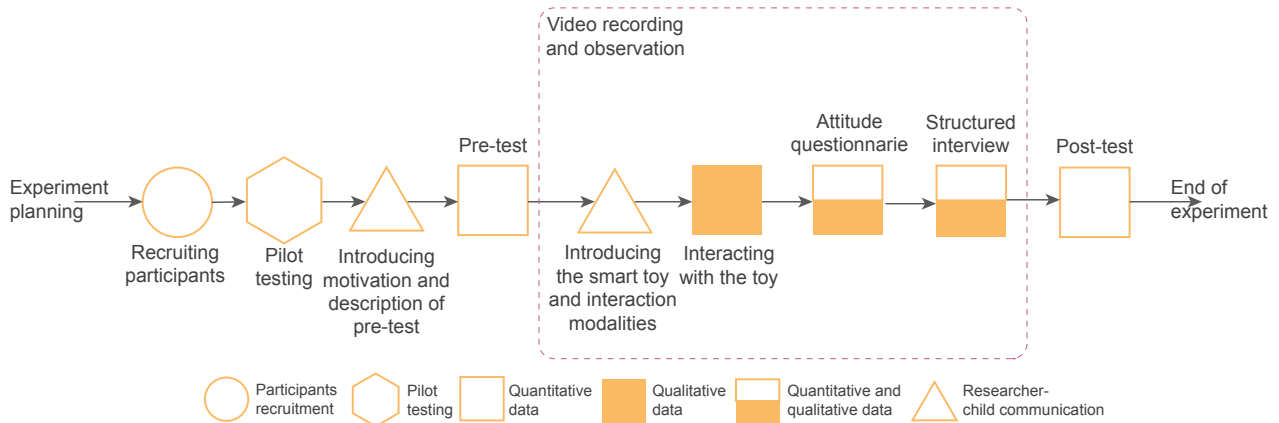


Figure 9. Overall framework of the experiment.

Laboratory equipment utilized for the experiment was as follows:

- Cardboard geometric shapes and boxes;
- Smart toy for geometry learning;
- Computer for data collection;
- Consent forms for parents/guardians.

Figure 10 gives the graphical representation of the laboratory setup and equipment applied in the experiment.



Figure 10. Visualization of laboratory setup.

## Experiment Procedure

The experiment procedure was as follows:

- **Recruitment:** The study employed a convenience sampling strategy, which is a non-probability sampling method. Convenience sampling entails selecting study participants who are readily available and willing to participate. The preschool children were recruited by the university staff, including non-scientific personnel and personal research networks, aiming to ensure that the study sample was as diverse as possible. The study was explained to parents/guardians, who were asked to consent to their child's participation. Overall, fourteen children (seven girls and seven boys) aged from 4 to 7 years old participated in the pilot study. The inclusion criteria included no previous exposure to the smart toy used in the study, as well as no history of developmental or learning disabilities.
- **Pre-test task:** Before interacting with the smart toy, each child was given a pre-test task to assess their current knowledge of basic geometric shapes. Children were given 30 simple cardboard geometric shapes (namely, 10 circles, 10 squares, and 10 pentagons) of different colors and sizes and were asked to put them in the appropriate box for each of the shapes. The evaluation was administered orally by the researcher.
- **Interaction with the smart toy:** Each child had 30 min to play with the toy. The researcher observed the child and documented their level of participation, motor aspects of interaction, interest, and overall behavior while interacting with the smart toy.
- **Data collection:** A video camera was used to record the participants during the experiment. It recorded the duration of the interaction, the accuracy of the completed task, and any errors made by the participants. It also captured the levels of engagement and other aspects of interactions that children had with the toy. Furthermore, the data were also collected by the smart toy in terms of sensor output data obtained from gesture movements.
- **Post-test task:** After interacting with the smart toy, each child completed a post-test task that was the same as the one in the pre-test task. They were again given 30 (new) simple cardboard geometric shapes (namely, 10 circles, 10 squares, and 10 pentagons) of different colors and sizes and were asked to put them in the appropriate box (new) for each of the shapes. The evaluation was administered orally by the researcher. The pre- and post-test tasks were further utilized to examine the effectiveness of the smart toy for geometry learning.
- **Follow-up interview and questionnaire:** The researcher asked close-ended questions about the child's engagement with the smart toy, its ease of use, their learning experience, and their subjective satisfaction while interacting.
- **Data Analysis:** Analyses of the overall collected data included statistical analysis while focusing on several aspects, such as fun and subjective user satisfactions, ease of use, engagement, returnance, and motor aspects of interaction. The pre- and post-test task results were compared to see if the interaction with the smart toy significantly improved geometry knowledge. The results of the questionnaire, interviews, and video recordings were also analyzed in order to gain insights into the child's level of engagement and overall satisfaction with the smart toy.

Experimental materials and methods for pilot testing included several techniques. Firstly, for the pre-test and post-test tasks, simple cardboard geometric shapes were utilized. Scientific studies, such as the ones presented in [88–92], have shown that cardboard cutouts are a valid and reliable tool in user evaluation studies. Simple cardboard geometric shapes can provide tangible representations of geometric concepts, making them appropriate for young children's learning. This method can also be used to establish a baseline to assess the effectiveness of educational toys and games in promoting geometric learning. These shapes offer a simple, low-cost, and effective method of assessing children's geometry skills and knowledge before and after using geometry learning technology. Secondly, a structured interview, as presented in Table 2, was used to gain insight into the children's fun and sub-

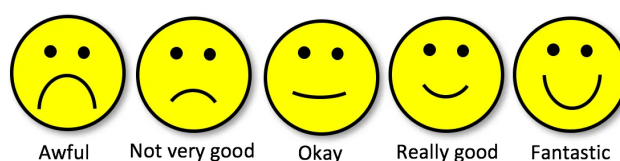
jective satisfactions, as well as their perceived ease of use (mapping of the particular shape) while playing with the smart toy. During usability evaluations, interviews are another way to investigate the user experience, as was shown in many user evaluation studies with children [84,86,93–96]. Post-task interviews allow for the collection of observational and verbalization data quickly and without the need for tape analysis. Post-task interviews have the potential to provide benefits at the expense of slightly longer evaluation sessions with children [86].

**Table 2.** Structured Interview.

Questions	Aspects of Exploration
1. Did you like the game?	Fun, subjective satisfaction
2. Which shape was the easiest for you to draw?	Ease of mapping
3. Which shape was the hardest for you to draw?	Ease of mapping
4. Was the game boring?	Fun, subjective satisfaction
5. Was the game difficult?	Fun, subjective satisfaction
6. Would you like to play this game again?	Fun, subjective satisfaction
7. What else would you like to teach the giraffe?	Engagement

Furthermore, to evaluate the children’s experiences in terms of subjective satisfaction, fun, and returnance, this study used two instruments from the Fun Toolkit: a survey instrument designed to help researchers and developers gather opinions about technology from children [87]. The Fun Toolkit has been used in numerous studies [20,93,97–99] to assess the usability of interactive technology with children, such as educational games, mobile applications, and interactive toys. It has been found to be a highly effective method to gain insight into children’s technological experiences and identify opportunities for improvement in the design of interactive products and services [83,100,101].

Namely, this research employed two instruments from the Fun Toolkit: the Smileyometer and the Again-Again table. The Smileyometer is a simple tool used to measure children’s subjective experiences with technology by asking them to rate their feelings using a visual scale of smiley faces. The tool is based on a 5-point Likert scale (as presented in Figure 11), with responses ranging from 1 (awful) to 5 (excellent) (brilliant) [100].



**Figure 11.** Smileyometer rating scale.

The Smileyometer has been widely adopted and used in research studies to assess and measure satisfaction and fun in children’s experiences with technology [20,97,101,102] because it is simple to use and does not require any writing on the part of children.

In this research, to assess how children felt during the interaction with the toy, they were asked “Can you show me, using these pictures, how you felt while playing this game?”. Then they were given Smileyometer rating scale cards to select the face that best suited their subjective feeling.

Another instrument from the Fun Toolkit used for the child user evaluation in this study was the Again-Again table. The table is used to assess the user experience by asking children if they want to repeat an activity again [87]. It has also been used in research studies [103–106], and has proven to be a reliable survey technique when applied to children [86]. In this research, to measure returnance (as one of the endurance dimensions), we derived the “Again-Again Table” (presented in Table 3) from the original presented in [100]. The table was filled by the researcher asking the research question: “Would you like to draw this shape again?”.

**Table 3.** The Again-Again Table.

	Would You like to Draw This Shape Again?		
	Yes	Maybe	No
Circle			
Square			
Pentagon			

Finally, this study used video recording as a tool to enhance three important aspects of interaction, namely: engagement, time-related aspects of interaction (time taken by the user to draw a shape, and overall interaction duration), and hand gestures used to interact with the toy, which were valuable for this research. Over the years, video recording has been shown to be a useful tool to collect detailed information on children's real-time interactions with technology [107,108]. Studies have shown that it is critical to recognize signs of enjoyment to determine whether the child had a positive or negative experience during the course of the interaction [97] because children often have difficulty articulating their experiences and preferences while using technology. In this context, video recording can be a powerful tool for gathering data and making informed decisions about how to improve technology for child users [109]. Researchers can identify patterns and trends in child behavior and movement by analyzing the recordings, which allows them to identify areas for improvement and optimize the user experience [108].

#### 4.2. Results

Over the course of three consecutive days, 14 children participated in the pilot study. Among them were seven girls and seven boys. Ten preschool children were 6 years old, three were 4 and 5 years old and went to kindergarten, and one was 7 years old and is a first grader.

##### 4.2.1. Objective Aspects of Interaction

The pre-test was designed to assess the children's knowledge of a variety of geometric shapes appropriate for their ages in order to study the change after using the proposed smart toy. The children can touch, feel, and manipulate cardboard cutouts, which provide a tangible and physical representation of geometric shapes. This enables the children to grasp and internalize geometric concepts and relationships. Most children do not have a thorough understanding of all geometric shapes at a young age, so it was important to examine if they can appropriately distinguish and name them. Table 4 shows how the children performed in the pre-test stage.

**Table 4.** The number of correct and incorrect answers given by children in the pre-test stage when identifying geometric shapes.

	Circle	Square	Pentagon
<b>Correct Answers</b>	100%	99.3%	99.3%
<b>Incorrect Answers</b>	0%	0.7%	0.7%

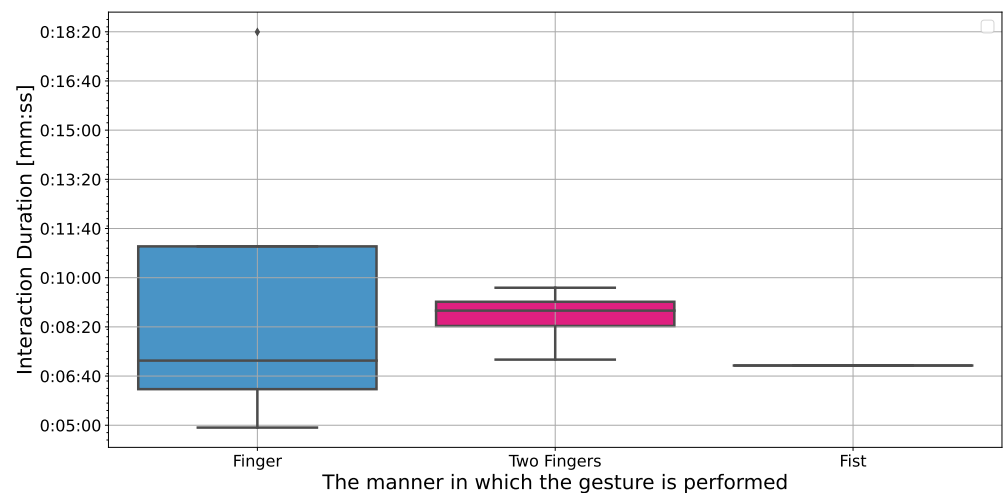
As can be seen, the children were good at distinguishing circles from squares and pentagons. However, due to the fact that cardboard geometric shapes were of different colors and sizes, on two occasions, a square was mistaken for a pentagon and vice versa.

Following the pre-test, the children were taken to a separate area of the laboratory where the smart toy giraffe was placed, as shown in Figure 10. The entire interaction process was recorded on video, and the researcher let the child become acquainted with the toy without intervention or specific instructions. The children were then asked if they wanted to "teach the giraffe" the geometric shapes they had been playing with in the

pre-test. Each child had 30 min to interact with the smart toy. The researcher instructed them to draw (map) the shape from the LCD screen above the giraffe's back with their hands as if they were drawing on a canvas or a board. During interaction with the smart toy, the researcher observed the child and recorded motor aspects of the interaction, their level of participation, interest, and overall behavior. The researcher labeled each gesture made by the child as correct or incorrect. This was later verified by analyzing the video recording. For each shape, the researcher asked the child if they wanted to play a bit more. When the child expressed a desire to stop playing, he or she was interviewed and encouraged to take the post-test.

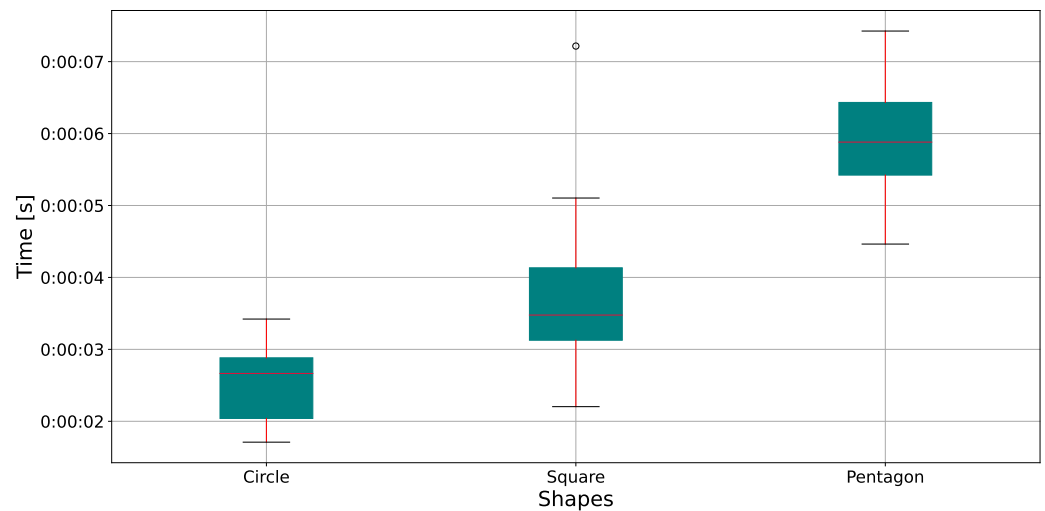
Firstly, four children did not establish the appropriate manner of interaction with the smart toy. Two of them were aged four and five (kindergarten) and eager to touch and cuddle the toy. They showed their emotions by smiling. The other two children were six-year-olds and tried to interact with the toy, however, they did not manage to do so. One of them did not show interest in the toy. This was especially evident in the fact that the child did not touch the giraffe at all. The other tried to perform the gestures but gave up and continued to play with the toy in his own way. This child was interested in the toy and expressed emotions by smiling.

In total, ten children managed to interact with the toy in a suitable way. The primary aspect of the interaction observed was the formation of the gesture. According to the results, five children performed the interaction with a single finger (index finger). Four children interacted with two fingers (thumb and index finger), while one child used the entire fist. Children who used one finger had longer interactions because they performed more gestures, while those who used two fingers or a fist had shorter interactions and performed fewer gestures, as exhibited in Figure 12. No child interacted with the toy for the planned period of 30 min. The majority of interactions lasted from around five to ten minutes. The first-grader engaged with the toy the longest and managed to make a significant number of gestures.



**Figure 12.** Interaction duration related to the established manner of the performed gesture.

The time required to form a specific shape was the second aspect of the observed interaction. Figure 13 shows the distribution of the time required to perform a particular gesture. There is an evident and reasonable increase in complexity correlated with the time required to perform a given gesture, with a circle requiring the least time and a pentagon demanding the most, which was to be expected. In the case of the square shape, there is an outlier caused by one child's playfulness, even though the gesture was correctly performed.



**Figure 13.** Gesture time per particular shape.

To identify any potential confounding variables in our limited data sample size, we conducted a search for variables that were correlated with both the independent variable and the dependent variable. Through our investigation, we discovered that age was highly positively correlated with the number of user gestures, the number of correct user gestures per particular shape, and the number of correct user gestures. Specifically, Pearson's correlation coefficient for age and the number of user gestures was 0.77, while for the number of correct gestures for the circle, square, and pentagon shapes, it was 0.74, 0.7, and 0.83, respectively. Additionally, Pearson's correlation coefficient for age and the number of overall correct user gestures was 0.77, indicating that age may be a confounding variable that needs to be controlled for analysis. We therefore calculated partial correlation coefficients between the number of correct user gestures for particular shapes and the number of performed gestures, while controlling for the effect of age. We found strong positive correlations between the number of correct user gestures and the number of performed gestures for the circle, square, and pentagon shapes, even after controlling for the effect of age. Specifically, the partial correlation coefficients were 0.942 ( $p$ -value = 0.0001), 0.84 ( $p$ -value = 0.004), and 0.899 ( $p$ -value = 0.001) for the circle, square, and pentagon shapes, respectively. The statistically significant relationship between the number of correct user gestures and performed gestures even after controlling for age suggests that age may not be a significant factor in predicting user performance for these shapes. This result may have implications for the future design of gesture-based interfaces, for instance, for older children.

The final part of the assessment of the motor aspects of interaction was the accuracy of the child's gesture mapping. This provides a subjective measure of the ease of mapping while interacting with the toy, which is an important aspect of user experience design. A gesture is considered correct if drawn on a virtual canvas above the sensors in the following way:

- A circle is drawn in 360 degrees, without overwriting the previous trajectory;
- The starting vertex for a square and pentagon is the same as the ending one, without repetition of previous edges.

This was evaluated in real time by the researcher during the experiment and validated by examining the video footage. The results presented in Figure 14 show a somewhat different and unexpected order of complexity among different shapes. That is, a circle has a higher failure rate than a square. This is most likely the result of outlining multiple circles on existing ones. As assumed, the failure rate for a pentagon is the highest.

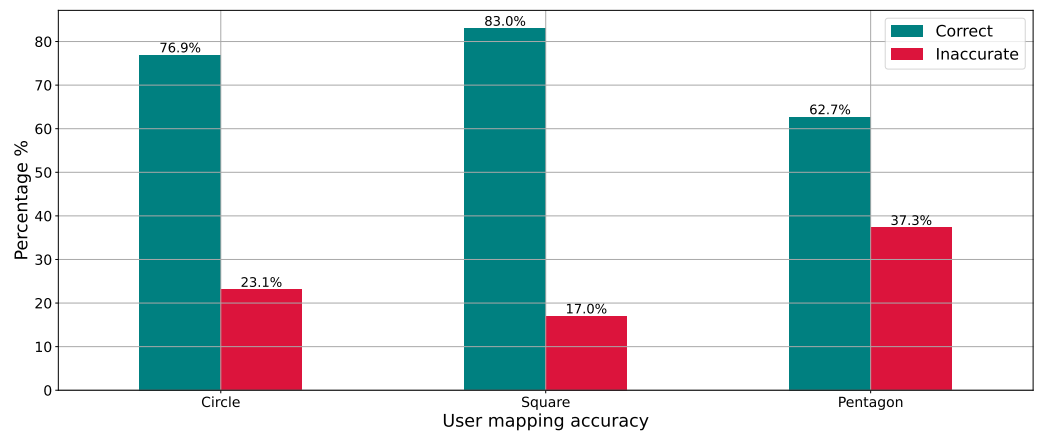


Figure 14. User mapping accuracy per particular shape.

4.2.2. Subjective Aspects of Interaction

These results were then compared with the child’s subjective experience related to the ease of mapping. Based on the answers provided from the interview questions “2. Which shape was the easiest for you to draw?” and “3. Which shape was the hardest for you to draw?”, the following results were obtained and are presented in Figure 15.

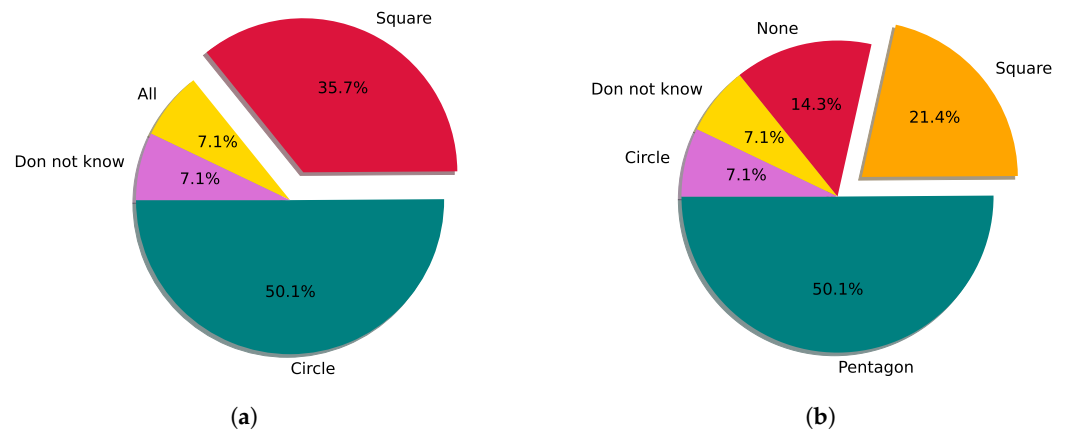


Figure 15. The results of the answers to interview questions 2 and 3. (a) Easiest to draw and (b) hardest to draw.

As can be seen, the children perceived the circle to be the easiest shape to map, as opposed to the pentagon, which they perceived to be the most difficult. This result correlates with the distribution of the time required to perform a specific gesture, with the circle requiring the least time and the pentagon requiring the most. However, these results are in contrast to the objective user mapping accuracy, as the square was the most accurately mapped shape. As was previously mentioned, this is probably due to the fact that a great number of children drew the circle by outlining multiple circles over existing ones.

Furthermore, the relationship between the perceived difficulty of different shapes and the actual time required to draw them was examined. The Mann–Whitney U test was used to compare the time taken to draw the hardest/most time-consuming shape (pentagon) with the time taken to draw the easiest/least time-consuming shape (circle). The null hypothesis, which stated that there would be no significant difference in time taken between the two shapes, was rejected based on the results of the test. The statistic was calculated to be 0.000000 and the *p*-value was found to be 0.00041, indicating a significant difference in the time taken between the two shapes. This suggests that the perceived difficulty of the shapes corresponds to the actual time required to draw them. These findings have implications for the design of educational materials and activities that involve drawing shapes, as they



suggest that the time required to draw a shape can be used as an objective measure of its difficulty.

Regarding the results from the children's subjective impressions of fun and satisfaction, valuable feedback from the children about their subjective experiences with the smart toy was obtained. Table 5 provides information on children's responses to question "Can you show me, using these pictures, how you felt while playing this game?".

**Table 5.** Fun and subjective satisfactions measured with the Smileyometer rating scale.

The Smileyometer Rating Scale Results					
	Awful	Not Very Good	Okay	Really Good	Fantastic
<b>Number of children</b>	0 (0%)	1 (7.1%)	2 (14.3%)	2 (14.3%)	9 (64.3%)

As can be seen, the majority of the children expressed a feeling of "Really good" or "Fantastic" while interacting with the smart toy. These results indicate that the children enjoyed the activity and experienced positive subjective satisfaction. This may also imply that, in future interactions, children are more likely to fully engage in toy play. These implications are supported by the results obtained from children's responses to interview questions "4. Was the game boring?" and "5. Was the game difficult?", presented in Table 6.

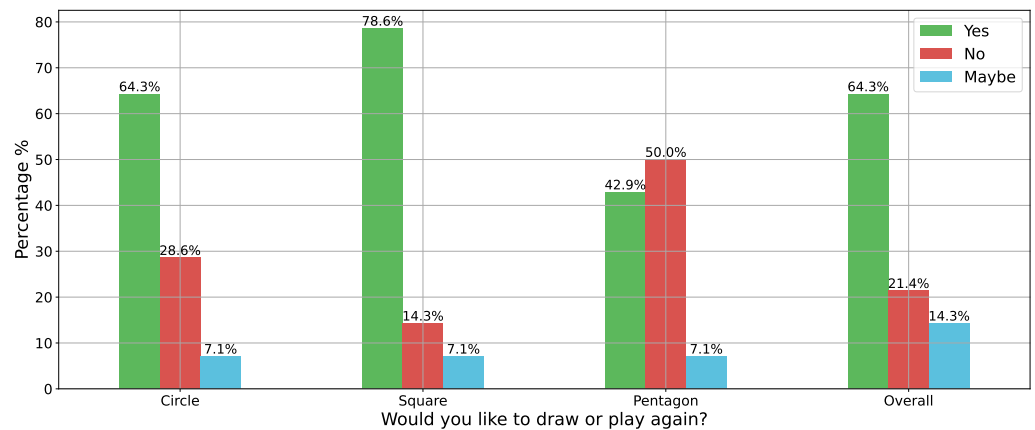
**Table 6.** The results of the answers to interview questions 4 and 5.

	Yes	No
<b>Was the game boring?</b>	3	11
<b>Was the game difficult?</b>	1	13

As can be seen, the children perceived the play with the giraffe to be engaging and easy. Such positive experiences indicate that the toy met expectations, which can be an important factor in promoting children's learning, since they are more likely to continue using the toy. The latter might result overall in greater technology adoption and success.

These implications are in correlation with the results obtained by analyses of video recordings of the children's expressions and behavior during the toy interaction. The majority of the children (12) smiled and were happy while interacting with the toy, one child danced and others bounced excitably. They were also keen on touching, petting, and exploring the toy, while at the same time communicating with the researcher. It was also noticed that some children, four of them, were more concentrated on the task itself rather than on the toy itself. Although they said they felt good interacting with the toy, they did not engage in other types of play with the toy apart from the proposed interaction. They were more interested in the toy's educational features. When asked, "7. What else would you like to teach the giraffe?" the majority of children just smiled and were unsure what to say other than "I don't know". However, some children provided rather interesting answers, such as "I would like to teach her letters", "I would like the draw hearts", and one child answered "I would like to teach her about good behavior."

Finally, the results of the returnability aspect based on the responses from the Again-Again Table 3 are presented in Figure 16.



**Figure 16.** Results from the responses from the Again-Again table (Table 3).

The results indicate that the majority of the children would like to play with the toy again. Furthermore, findings suggest that the children found the square shape to be the most engaging and interesting to play with, as evidenced by their desire to play with it again and their preference for drawing the square. This preference may be related to the objective user mapping aspect, in which the square was the most accurately mapped shape. It is also worth noting that, despite the children’s subjective assessment that the circle was the easiest shape to draw, they preferred drawing the square. This suggests that a child’s interest in the toy was not solely determined by its ease of use. Overall, these findings suggest that future enhancements to the toy’s design should consider not only the ease of use but also the toy’s engagement factor. The objective user mapping aspect can also be considered to increase engagement. As was to be expected, half of the children would not want to draw the pentagon again. It is possible that the children’s lack of interest in drawing the pentagon again is related to their level of motor skill development, as the pentagon has more sides and angles than the other shapes, potentially making it more difficult to draw. They may also feel less confident or interested in attempting to draw the pentagon again or it may be that they found the pentagon more challenging to understand or remember compared to the other shapes. This implication is supported by the researcher’s observations as well as the video analyses, as none of the children were familiar with the shape or knew its name and usually referred to it as the “house shape”.

An immediate post-test followed the interaction with the toy. The results of the test are presented in Table 7. Only the results of the children who interacted with the toy were taken into account. As can be noticed, the accuracy of recognizing and classifying the pentagon seems to decline. This was probably an immediate result of the fatigue of one child who incorrectly classified the pentagon as a square several times, since this child interacted with the toy the longest and performed a great number of gestures. Overall, due to the small sample size, a definitive conclusion about the impact of the toy on the children’s performance in the post-test cannot be drawn. Therefore, in the future, it is important to ensure that sample sizes are adequate to make accurate claims about the impact of the toy on children’s educational performance.

**Table 7.** The number of correct and incorrect answers given by children in the post-test stage for identifying geometric shapes.

	Circle	Square	Pentagon
<b>Correct Answers</b>	99%	99%	95%
<b>Incorrect Answers</b>	1%	1%	5%

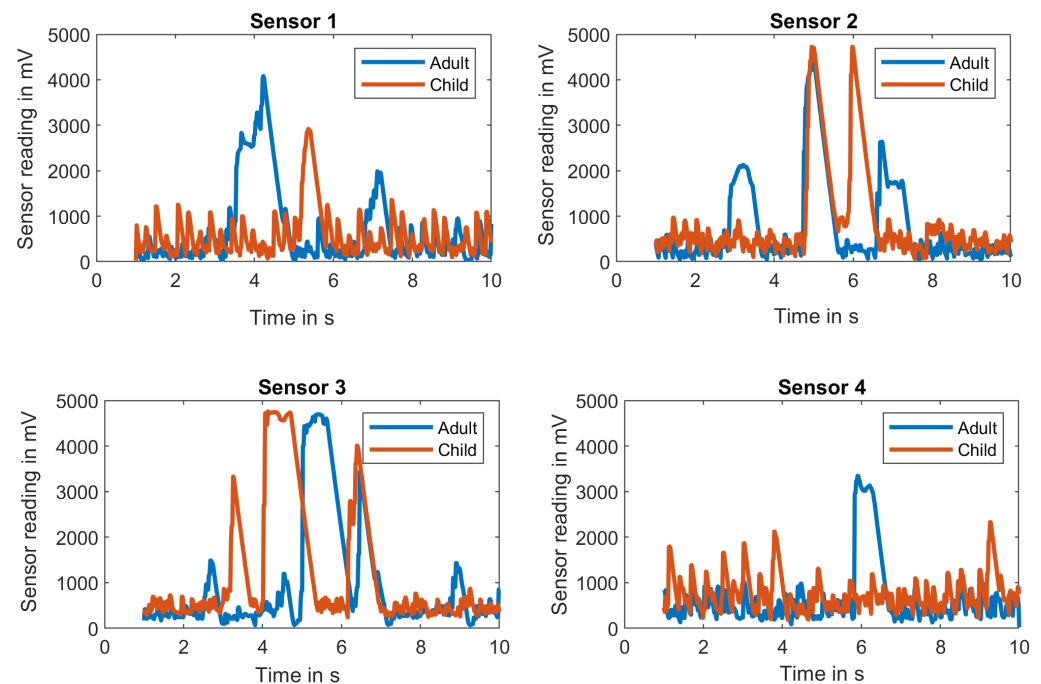
#### 4.2.3. Machine Learning Performance

Finally, the performance of the neural network in experimental scenarios is presented. The children performed an overall number of 111 different gestures, and Table 8 provides insight into the gesture classification accuracy.

**Table 8.** Machine learning gesture classification accuracy results.

	Circle		Square		Pentagon	
	Guessed	Missed	Guessed	Missed	Guessed	Missed
<b>Number of gestures</b>	25	15	19	20	6	26

As can be observed, the classification accuracy is quite low, especially for the pentagon shape. There are several possible reasons for such a bad performance. To begin with, our experimental results have shown that children's gestures differ from adult gestures in terms of frequency and execution. As demonstrated, children performed gestures primarily with their index fingers, while the data used to build the model came from adult users who primarily used their entire fists. Furthermore, of those 111 gestures, half (46, to be exact) came from a single user, the first grader, who made gestures with his index finger, while the other 55 gestures were distributed among the other children, indicating an imbalance in the test set. With that in regard, we later conducted a comparison of raw sensor data from a child and an adult subject while performing gestures for the same geometrical shape, as presented in Figure 17.



**Figure 17.** Comparison of raw sensor data of child and adult subject while performing gestures for a same geometrical shape.

It can be observed that children's gestures greatly differ from adult gestures, both in terms of their frequency and the way they are executed. The adults generally performed gestures with the entire fist, where, as we have seen, the children primarily used their index finger. Therefore, the feature set used to train the machine learning model was unable to accurately capture the variation in children's gestures, leading to a poor classification accuracy.

As this was a pilot study, it provided valuable information on the performance of the smart toy for geometry learning and identified areas for improvement. This information will be used to guide future design iterations, helping to refine the smart toy and improve the accuracy of the machine learning algorithm. In this regard, it will be necessary to collect a large dataset of children's gestures and train the machine learning model specifically on this dataset. This can involve collecting data from a range of ages and developmental stages to ensure that the model can capture the variation in children's gestures. Additionally, it may be necessary to develop new feature sets or modify existing ones to better capture the unique features of children's gestures. Finally, it may be necessary to test the model on a separate validation dataset to ensure that it generalizes well to new examples of children's gestures.

## 5. Discussion

Based on the results of the exploratory pilot study, some of the benefits and strong aspects of this research were the following.

The activity involving drawing shapes was enjoyable and engaging for the majority of the children. The fact that the square was the most correctly mapped shape and that children expressed interest in drawing it again suggests that the activity was effective in promoting learning and skill development related to shape recognition and the mapping of abstract concepts. It should also be noted that, although the children perceived the circle as the easiest shape to draw, they still showed a greater preference for drawing the square. This may indicate that the children found the challenge of drawing the square more rewarding or satisfying than drawing the circle. The toy was also found to be easy to use, which is important to ensure that children can use it independently. The positive results obtained using the Smileyometer rating scale when and during the interview while measuring fun and subjective satisfaction with technology indicate that the children experienced a high level of enjoyment and satisfaction, which can have a significant impact on the success of the smart toy giraffe as an interactive and educational tool. Overall, the results suggest that the activities involving the smart toy were engaging and challenging and can prove to be effective in promoting the early learning of geometry of children in preschool.

One of the main benefits was also the deeper insight into the modalities of interaction that the children had with the toy, as well as the different ways in which the children make the gestures in contrast to the adults.

The results of our study indicate that age is a potential confounding variable that needs to be controlled when analyzing the relationship between user performance and the number of performed gestures for specific shapes. However, even after controlling for age, a statistically significant positive correlation was found between the number of correct user gestures and performed gestures for the circle, square, and pentagon shapes. This suggests that age may not be a significant predictor of user performance for these shapes, which may have implications for the design of gesture-based interfaces, particularly for older children. Moreover, it was found that the perceived difficulty of a shape corresponds to the actual time required to draw it. This finding may be useful for designing educational materials and activities that involve drawing shapes, as it provides an objective measure of shape difficulty. The Mann–Whitney U test was used to compare the time taken to draw the hardest/most time-consuming shape (pentagon) with the time taken to draw the easiest/least time-consuming shape (circle), and a significant difference in the time taken was found between the two shapes. These findings contribute to a better understanding of the factors that influence user performance in gesture-based interfaces and highlight the importance of considering potential confounding variables in data analysis.

However, there were some limitations to this study. Although the results of the machine learning algorithm showed satisfactory results in the adult dataset, the same model performed poorly with child subjects for several reasons. At this stage of the design process, the recruitment of child participants for the data collection phase was challenging due to ethical concerns and limited access to child populations. Additionally,

collecting high-quality and representative data was more difficult due to the younger children's limited attention spans and potential fatigue when performing a larger series of gesture movements. These issues should be addressed in future work to improve the performance of the machine learning model on child data. Furthermore, the setup consisting of four IR distance sensors arranged in an array creates a relatively shallow area (that resembles a plane) where measurement can be executed. When children performed gestures above the sensor area, their movements were often less coordinated and precise than those of adults. This is the result of their phase of development of fine motor skills, making it difficult for them to execute gestures using their entire fist or with the same level of control as adults. As a result, children tend to rely on simpler and more straightforward gestures that are easier to execute, such as pointing with their index finger. Furthermore, children may be more prone to unintentional movements or gestures, which can affect the accuracy of the machine learning algorithm used to detect the outline of the gesture. As our model is trained primarily on adult users, using the same ML model on child users showed a significant reduction in accuracy. By analyzing raw data in Figure 17, we can draw some basic conclusions: adult users activate more sensors while performing this particular gesture as compared to child users. This could be due to the adult user executing a larger gesture, the height of the adult user, and the difference in hand surface that reflect the IR emitted to a sensor. To bring child user accuracy to a level of an adult user, we must obtain more training data that can help us extract specific features in a signal that are found in child users. For future work, the following is considered.

In future work, alternative sensors with larger sensing areas can be considered, such as the VL53L5CX Time of Flight sensor with an  $8 \times 8$  multizone range and  $63^\circ$  diagonal field of view (<https://www.st.com/en/imaging-and-photonics-solutions/vl53l5cx.html> (accessed on 11 April 2023)). Similar results may be achieved by implementing one or more microcontroller-based machine vision cameras such as OpenMV H7 (<https://openmv.io/products/openmv-cam-h7> (accessed on 11 April 2023)) in a multi-vision configuration. As those cameras are basically machine-vision sensors, when properly programmed, they output simple hand location information in predefined coordinate space and the overall complexity of the system can be kept at a reasonable level, requiring minimal setup time or preparation.

Finally, it will be necessary to collect a large dataset of children's gestures and train the model specifically on these data in order to enhance the machine learning model's accuracy when classifying children's gestures. This may involve collecting data from children of various ages and developmental stages to ensure that the model can capture the variation in their gestures. It will also be necessary to test the model on a separate validation dataset to ensure that it can accurately generalize to new examples of children's gestures.

## 6. Conclusions

The literature suggests that utilizing smart toys in preschool education has the potential to foster STEM skills in young children. This study aimed to introduce a prototype of a plush smart toy as an educational tool for teaching young children about geometric shapes, given the potential of using smart toys in preschool education and the importance of studying geometry at a young age for the development of spatial reasoning skills. The plush smart toy design incorporates a range of hardware components, including sensors, microcontrollers, an LCD screen, and a machine learning system, which facilitates gesture recognition. By analyzing the outline of the child's gesture, the machine learning system can determine whether it corresponds to the shape displayed on the LCD screen. Among the three sensing technologies tested, namely capacitive, IR short-range, and IR long-range sensors, the IR long-range technology was found to be the most suitable for the study, based on the machine learning results. Later, a small exploratory pilot study was conducted to analyze the nature of the children's involvement with the smart toy prototype through user evaluation, test the toy's practicality, and acquire some preliminary data on the toy's effectiveness and feasibility. The results of the exploratory study highlighted

several benefits and strong aspects of the smart toy prototype. According to the findings of the study, the smart toy prototype is user-friendly and straightforward to use. The result also indicates that the smart toy engages children in the learning process effectively, making it a potentially valuable educational tool for preschool children. Additionally, the study provided valuable insights into the modalities of interaction between children and the toy, the differences in gestures made by children compared to adults, and the influence of age on user performance in gesture-based interfaces. These findings contribute to a better understanding of the factors affecting user performance and emphasize the importance of considering potential confounding variables in data analysis. Furthermore, our research revealed that the perceived difficulty of a shape corresponds to the real time necessary to draw it. This discovery could be useful in constructing instructional materials and activities that include drawing shapes because it provides an objective measure of shape difficulty.

However, limitations were identified in the machine learning algorithm's ability to recognize children's gestures and the sensor setup's capacity to capture the full range and precision of children's gestures. The lower accuracy rate with children is due to the different ways in which they make gestures compared to the adults who were used in the data collection process, as well as the test set data imbalance. To address these limitations, future work should consider alternative sensors with larger sensing areas, for instance, a time of flight sensor or microcontroller-based machine vision cameras. These technologies can help maintain a reasonable level of system complexity while improving the accuracy and reliability of the gesture recognition toy for children. Finally, future research should focus on collecting a larger dataset of children's gestures and training the machine learning model specifically on these data to enhance its accuracy and generalizability. This may involve gathering data from children of various ages and developmental stages to ensure that the model captures the variation in their gestures and testing the model on a separate validation dataset to confirm its accurate generalization to new examples of children's gestures.

**Author Contributions:** Conceptualization, L.D.R., I.S., D.Č., T.P. and A.G.; methodology, L.D.R., I.S., D.Č. and A.G.; software, I.S.; validation, L.D.R., I.S. and D.Č.; formal analysis, L.D.R., I.S. and T.P.; investigation, L.D.R.; resources, L.D.R.; data curation, L.D.R., I.S., D.Č. and T.P.; writing—original draft preparation, L.D.R., I.S. and D.Č.; writing—review and editing, L.D.R., I.S., T.P. and D.Č.; visualization, I.S. and T.P.; supervision, L.D.R. and A.G. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the Croatian Science Foundation under the project "Internet of Things: Research and Applications", UIP-2017-05-4206.

**Institutional Review Board Statement:** The study was conducted in accordance with the Declaration of Helsinki, and approved by the Ethics Committee of Faculty of Electrical Engineering, Mechanical Engineering and Naval Architecture (protocol code 641-01/22-01/0030, approved 21 October 2022).

**Informed Consent Statement:** Informed consent was obtained from all subjects involved in the study.

**Data Availability Statement:** Access to the data is restricted to researchers who have obtained permission from the Ethics Committee of the Faculty of Electrical Engineering, Mechanical Engineering and Naval Architecture, due to the involvement of minors in the study. Therefore, the data is not available to the general public.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

IoT	Internet of Things
STEM	Science, technology, engineering, and mathematics
LCD	Liquid-crystal display
IR	Infrared
PC	Personal computer

GND	Ground
USB	Universal Serial Bus
GPIO	General purpose input–output
ML	Machine learning
NN	Neural network
TOF	Time of flight
AI	Artificial intelligence

## References

- Li, S.; Xu, L.D.; Zhao, S. The internet of things: A survey. *Inf. Syst. Front.* **2015**, *17*, 243–259. [CrossRef]
- Zhu, Z.T.; Yu, M.H.; Riezebos, P. A research framework of smart education. *Smart Learn. Environ.* **2016**, *3*, 4. [CrossRef]
- Stojanović, D.; Bogdanović, Z.; Petrović, L.; Mitrović, S.; Labus, A. Empowering learning process in secondary education using pervasive technologies. *Interact. Learn. Environ.* **2020**, *31*, 779–792. [CrossRef]
- Sami, R.; Abbas, S.; Ibrahim, Z. A survey: The role of the internet of things in the development of education. *Indones. J. Electr. Eng. Comput. Sci.* **2020**, *19*, 215. [CrossRef]
- Kuppusamy, P. Smart education architecture using the Internet of Things (IOT) technology. *Int. J. Manag. Educ.* **2019**, *9*, 46–70.
- McRae, L.; Ellis, K.; Kent, M. *Internet of Things (IoT): Education and Technology*; Curtin University: Bentley, WA, Australia, 2018; pp. 1–37.
- Manches, A.; Duncan, P.; Plowman, L.; Sabeti, S. Three questions about the Internet of things and children. *TechTrends* **2015**, *59*, 76–83. [CrossRef]
- Lin, V.; Yeh, H.C.; Chen, N.S. Enhancing EFL vocabulary learning with multimodal cues supported by an educational robot and an IoT-Based 3D book. *System* **2022**, *104*, 102691. [CrossRef]
- Meadthaisong, S.; Meadthaisong, T. Smart Farming Using Internet of Thing(IoT) in Agriculture by Tangible Programming for Children. In Proceedings of the 2020 17th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), Phuket, Thailand, 24–27 June 2020; pp. 611–614. [CrossRef]
- Al-Htaybat, K.; von Alberti-Alhtaybat, L.; Alhatabat, Z. Educating digital natives for the future: Accounting educators' evaluation of the accounting curriculum. *Account. Educ.* **2018**, *27*, 333–357. [CrossRef]
- Čoko, D.; Rodić, L.D.; Perković, T.; Šolić, P. Geometry from Thin Air: Theremin as a Playful Learning Device. In Proceedings of the 2021 16th International Conference on Telecommunications (ConTEL), Zagreb, Croatia, 30 June 2021–2 July 2021; pp. 89–96. [CrossRef]
- He, J.S.; Ji, S.; Bobbie, P.O. Internet of Things (IoT)-Based Learning Framework to Facilitate STEM Undergraduate Education. In Proceedings of the SouthEast Conference, ACM SE '17, Kennesaw, GA, USA, 13–15 April 2017; Association for Computing Machinery: New York, NY, USA, 2017; pp. 88–94. [CrossRef]
- Kimmo, K. *Finland: Country Report on ICT in Education*; European Schoolnet (EUN): Brussels, Belgium, 2017.
- The Strategy for Education, Science and Technology. The National Curricula for Early Childhood and Preschool Education. 2014. Available online: <https://mzo.gov.hr/strategy-for-education-science-and-tehnology/385> (accessed on 11 April 2023).
- Dujić Rodić, L.; Granić, A. Tangible User Interfaces for Enhancement of Young Children's Mathematical Problem Solving and Reasoning: A Preliminary Review of Relevant Literature. In Proceedings of the CECIIS 2018: 29th Central European Conference on Information and Intelligent Systems: Proceedings, Varazdin, Croatia, 19–21 September 2018; Faculty of Organization and Informatics, University of Zagreb: Zagreb, Croatia, 2018; pp. 77–84.
- Cağıltay, K.; Kara, N.; Aydin, C.C. Smart Toy Based Learning. In *Handbook of Research on Educational Communications and Technology*; Spector, J.M., Merrill, M.D., Elen, J., Bishop, M.J., Eds.; Springer: New York, NY, USA, 2014; pp. 703–711. [CrossRef]
- Komis, V.; Karachristos, C.; Mourta, D.; Sgoura, K.; Misirli, A.; Jaillet, A. Smart Toys in Early Childhood and Primary Education: A Systematic Review of Technological and Educational Affordances. *Appl. Sci.* **2021**, *11*, 8653. [CrossRef]
- Frei, P.; Su, V.; Mikhak, B.; Ishii, H. *Curlybot: Designing a New Class of Computational Toys, Proceedings of the CHI '00: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, New York, NY, USA, 1–6 April 2000*; Association for Computing Machinery: New York, NY, USA, 2000; pp. 129–136. [CrossRef]
- He, X.; Li, T.; Turel, O.; Kuang, Y.; Zhao, H.; He, Q. The Impact of STEM Education on Mathematical Development in Children Aged 5-6 Years. *Int. J. Educ. Res.* **2021**, *109*, 101795. [CrossRef]
- Bazargani, J.S.; Sadeghi-Niaraki, A.; Rahimi, F.; Abuhmed, T.; Choi, S.M. An IoT-Based Approach for Learning Geometric Shapes in Early Childhood. *IEEE Access* **2022**, *10*, 130632–130641. [CrossRef]
- Tripathy, H.; Mishra, S.; Dash, K. *Significance of IoT in Education Domain*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 59–83. [CrossRef]
- Akdeniz, M.; Özding, F. Maya: An artificial intelligence based smart toy for pre-school children. *Int. J. Child-Comput. Interact.* **2021**, *29*, 100347. [CrossRef]
- Heljakka, K.; Ihamäki, P. Preschoolers learning with the Internet of Toys: From toy-based edutainment to transmedia literacy. *Seminar.net* **2018**, *14*, 85–102. [CrossRef]

24. Heljakka, K.; Ihamäki, P.; Tuomi, P.; Saarikoski, P. Gamified Coding: Toy Robots and Playful Learning in Early Education. In Proceedings of the 2019 International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, USA, 5–7 December 2019; pp. 800–805. [[CrossRef](#)]
25. Kusmin, M.; Saar, M.; Laanpere, M. Smart schoolhouse—Designing IoT study kits for project-based learning in STEM subjects. In Proceedings of the 2018 IEEE Global Engineering Education Conference (EDUCON), Santa Cruz de Tenerife, Spain, 17–20 April 2018; pp. 1514–1517. [[CrossRef](#)]
26. Papert, S. *Children, Computers, and Powerful Ideas*; Harvester: Eugene, OR, USA, 1980.
27. Piaget, J. Cognitive Development in Children Development and Learning. *J. Res. Sci. Teach.* **1964**, *2*, 176–186. [[CrossRef](#)]
28. Montessori, M. *The Montessori Method by Maria Montessori (1870–1952)*; George, A.E.; Frederick, A., Translators; Stokes Company: New York, NY, USA, 1912.
29. Antle, A.N. Exploring how children use their hands to think: An embodied interactional analysis. *Behav. Inf. Technol.* **2013**, *32*, 938–954. [[CrossRef](#)]
30. Yu, J.; Roque, R. A review of computational toys and kits for young children. *Int. J. Child-Comput. Interact.* **2019**, *21*, 17–36. [[CrossRef](#)]
31. Sullivan, A.; Elkin, M.; Bers, M.U. KIBO robot demo: Engaging young children in programming and engineering. In Proceedings of the 14th International Conference on Interaction Design and Children, Medford, MA, USA, 21–25 June 2015; pp. 418–421.
32. Delprino, F.; Piva, C.; Tommasi, G.; Gelsomini, M.; Izzo, N.; Matera, M. ABBOT: A smart toy motivating children to become outdoor explorers. In Proceedings of the 2018 International Conference on Advanced Visual Interfaces, Pescaia, Italy, 29 May–1 June 2018; pp. 1–9.
33. Putjorn, P.; Siriaraya, P.; Ang, C.S.; Deravi, F. Designing a Ubiquitous Sensor-Based Platform to Facilitate Learning for Young Children in Thailand. In Proceedings of the 19th International Conference on Human-Computer Interaction with Mobile Devices and Services, Association for Computing Machinery: New York, NY, USA, 4–7 September 2017. [[CrossRef](#)]
34. Zhang, L.; Sun, F.; Wang, D.; Dai, G. A Tangible Interaction Tool for Children and its application in Storytelling. *J. Comput.-Aided Des. Comput. Graph.* **2017**, *29*, 557–564.
35. Heljakka, K.; Ihamäki, P. Ready, steady, move! Coding toys, preschoolers, and mobile playful learning. In Proceedings of the Learning and Collaboration Technologies. Ubiquitous and Virtual Environments for Learning and Collaboration: 6th International Conference, LCT 2019, Held as Part of the 21st HCI International Conference, HCII 2019, Orlando, FL, USA, 26–31 July 2019; Springer: Berlin, Germany, 2019; pp. 68–79.
36. Lin, S.Y.; Chien, S.Y.; Hsiao, C.L.; Hsia, C.H.; Chao, K.M. Enhancing Computational Thinking Capability of Preschool Children by Game-based Smart Toys. *Electron. Commer. Res. Appl.* **2020**, *44*, 101011. [[CrossRef](#)]
37. de Paula Albuquerque, O.; Fantinato, M.; Kelner, J.; de Albuquerque, A.P. Privacy in smart toys: Risks and proposed solutions. *Electron. Commer. Res. Appl.* **2020**, *39*, 100922. [[CrossRef](#)]
38. Pontes, L.; Coutinho, G.; Hung, P.C.; Yankson, B. Security in smart toys: A systematic review of literature. In Proceedings of the Distributed, Ambient and Pervasive Interactions: 7th International Conference, DAPI 2019, Held as Part of the 21st HCI International Conference, HCII 2019, Orlando, FL, USA, 26–31 July 2019; Springer: Berlin, Germany, 2019; pp. 28–38.
39. Kara, N.; Cagiltay, K. Smart toys for preschool children: A design and development research. *Electron. Commer. Res. Appl.* **2020**, *39*, 100909. [[CrossRef](#)]
40. Maričić, S.; Stamatović, J. The effect of preschool mathematics education in the development of geometry concepts in children. *Eurasia J. Math. Sci. Technol. Educ.* **2017**, *13*, 6175–6187. [[CrossRef](#)]
41. Novita, R.; Putra, M.; Rosayanti, E.; Fitriati, F. Design learning in mathematics education: Engaging early childhood students in geometrical activities to enhance geometry and spatial reasoning. *J. Phys. Conf. Ser.* **2018**, *1088*, 012016. [[CrossRef](#)]
42. Bautista, A.; Roth, W.M.; Thom, J.S. Knowing, insight learning, and the integrity of kinetic movement. *Interchange* **2011**, *42*, 363–388. [[CrossRef](#)]
43. Thom, J.S.; McGarvey, L.M. The act and artifact of drawing(s): Observing geometric thinking with, in, and through children’s drawings. *ZDM* **2015**, *47*, 465–481. [[CrossRef](#)]
44. Elia, I. Observing the use of gestures in young children’s geometric thinking. In *Contemporary Research and Perspectives on Early Childhood Mathematics Education*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 159–182.
45. Thom, J.S. (Re)(con) figuring Space: Three children’s geometric reasonings. In *Contemporary Research and Perspectives on Early Childhood Mathematics Education*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 131–158.
46. Sin, Z.P.; Ng, P.H.; Leong, H.V. Stuffed toy as an appealing tangible interface for children. In Proceedings of the 2021 IEEE International Symposium on Multimedia (ISM), Naples, Italy, 29 November–1 December 2021; IEEE: New York, NY, USA, 2021; pp. 75–78.
47. Sugiura, Y.; Lee, C.; Ogata, M.; Withana, A.; Makino, Y.; Sakamoto, D.; Inami, M.; Igarashi, T. PINOKY: A ring that animates your plush toys. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Austin, TX, USA, 5–10 May 2012; pp. 725–734.
48. Ihamäki, P.; Heljakka, K. Smart, skilled and connected in the 21st century: Educational promises of the Internet of Toys (IoToys). In Proceedings of the 2018 Hawaii University International Conferences, Arts, Humanities, Social Sciences & Education, Prince Waikiki Hotel, Honolulu, HI, USA, 7–9 June 2018; pp. 1–19.



49. Kara, N.; Aydin, C.C.; Cagiltay, K. Design and development of a smart storytelling toy. *Interact. Learn. Environ.* **2014**, *22*, 288–297. [[CrossRef](#)]
50. Fontijn, W.; Mendels, P. StoryToy the interactive storytelling toy. In Proceedings of the Second International Workshop on Gaming Applications in Pervasive Computing Environments at Pervasive, Munich, Germany, 8–13 May 2005.
51. Tseng, T.; Murai, Y.; Freed, N.; Gelosi, D.; Ta, T.D.; Kawahara, Y. PlushPal: Storytelling with interactive plush toys and machine learning. In Proceedings of the Interaction Design and Children, Athens, Greece, 14–30 June 2021; pp. 236–245.
52. Antle, A.N.; Wise, A.F. Getting Down to Details: Using Theories of Cognition and Learning to Inform Tangible User Interface Design. *Interact. Comput.* **2013**, *25*, 1–20. [[CrossRef](#)]
53. Suggate, S.; Stoeger, H.; Pufke, E. Relations between playing activities and fine motor development. *Early Child Dev. Care* **2017**, *187*, 1297–1310. [[CrossRef](#)]
54. Verdine, B.; Golinkoff, R.; Hirsh-Pasek, K.; Newcombe, N. I. spatial skills, their development, and their links to mathematics. *Monogr. Soc. Res. Child Dev.* **2017**, *82*, 7–30. [[CrossRef](#)]
55. Tran, D.S.; Ho, N.H.; Yang, H.J.; Baek, E.T.; Kim, S.H.; Lee, G. Real-Time Hand Gesture Spotting and Recognition Using RGB-D Camera and 3D Convolutional Neural Network. *Appl. Sci.* **2020**, *10*, 722. [[CrossRef](#)]
56. Avola, D.; Cinque, L.; Fagioli, A.; Foresti, G.L.; Fragomeni, A.; Pannone, D. 3D hand pose and shape estimation from RGB images for keypoint-based hand gesture recognition. *Pattern Recognit.* **2022**, *129*, 108762. [[CrossRef](#)]
57. Čoko, D.; Stančić, I.; Dujčić Rodić, L.; Čošić, D. TheraProx: Capacitive Proximity Sensing. *Electronics* **2022**, *11*, 393. [[CrossRef](#)]
58. Oudah, M.; Al-Naji, A.; Chahl, J. Hand Gesture Recognition Based on Computer Vision: A Review of Techniques. *J. Imaging* **2020**, *6*, 73. [[CrossRef](#)]
59. Stančić, I.; Musić, J.; Grujić, T. Gesture recognition system for real-time mobile robot control based on inertial sensors and motion strings. *Eng. Appl. Artif. Intell.* **2017**, *66*, 33–48. [[CrossRef](#)]
60. Lee, M.; Bae, J. Deep Learning Based Real-Time Recognition of Dynamic Finger Gestures Using a Data Glove. *IEEE Access* **2020**, *8*, 219923–219933. [[CrossRef](#)]
61. Sharma, S.; Singh, S. Vision-based hand gesture recognition using deep learning for the interpretation of sign language. *Expert Syst. Appl.* **2021**, *182*, 115657. [[CrossRef](#)]
62. Xia, Z.; Lei, Q.; Yang, Y.; Zhang, H.; He, Y.; Wang, W.; Huang, M. Vision-Based Hand Gesture Recognition for Human-Robot Collaboration: A Survey. In Proceedings of the 2019 5th International Conference on Control, Automation and Robotics (ICCAR), Beijing, China, 19–22 April 2019; pp. 198–205. [[CrossRef](#)]
63. Al, G.A.; Estrela, P.; Martinez-Hernandez, U. Towards an intuitive human-robot interaction based on hand gesture recognition and proximity sensors. In Proceedings of the 2020 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI), Karlsruhe, Germany, 14–16 September 2020; pp. 330–335. [[CrossRef](#)]
64. Virone, M.; Lopes, P.; Rocha, R.P.; de Almeida, A.T.; Tavakoli, M. Dynamic hand gesture recognition using a stretchable multi-layer capacitive array, proximity sensing, and a SVM classifier. In Proceedings of the 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Prague, Czech Republic, 27 September–1 October 2021; pp. 7183–7188. [[CrossRef](#)]
65. Moysiadis, V.; Katikaridis, D.; Benos, L.; Busato, P.; Anagnostis, A.; Kateris, D.; Pearson, S.; Bochtis, D. An Integrated Real-Time Hand Gesture Recognition Framework for Human-Robot Interaction in Agriculture. *Appl. Sci.* **2022**, *12*, 8160. [[CrossRef](#)]
66. Huang, Y.; Yang, J. A multi-scale descriptor for real time RGB-D hand gesture recognition. *Pattern Recognit. Lett.* **2021**, *144*, 97–104. [[CrossRef](#)]
67. Wong, W.K.; Juwono, F.H.; Khoo, B.T.T. Multi-Features Capacitive Hand Gesture Recognition Sensor: A Machine Learning Approach. *IEEE Sens. J.* **2021**, *21*, 8441–8450. [[CrossRef](#)]
68. Özcan, M.; Aliew, F.; Görgün, H. Accurate and precise distance estimation for noisy IR sensor readings contaminated by outliers. *Measurement* **2020**, *156*, 107633. [[CrossRef](#)]
69. Starace, G.; Tiwari, A.; Colangelo, G.; Massaro, A. Advanced Data Systems for Energy Consumption Optimization and Air Quality Control in Smart Public Buildings Using a Versatile Open Source Approach. *Electronics* **2022**, *11*, 3904. [[CrossRef](#)]
70. Le Glaz, A.; Haralambous, Y.; Kim-Dufor, D.H.; Lenca, P.; Billot, R.; Ryan, T.C.; Marsh, J.; DeVylder, J.; Walter, M.; Berrouiguet, S.; et al. Machine Learning and Natural Language Processing in Mental Health: Systematic Review. *J. Med. Internet Res.* **2021**, *23*, e15708. [[CrossRef](#)] [[PubMed](#)]
71. Wang, P.; Fan, E.; Wang, P. Comparative analysis of image classification algorithms based on traditional machine learning and deep learning. *Pattern Recognit. Lett.* **2021**, *141*, 61–67. [[CrossRef](#)]
72. Pedraza, L.F.; Hernández, H.A.; Hernández, C.A. Artificial Neural Network Controller for a Modular Robot Using a Software Defined Radio Communication System. *Electronics* **2020**, *9*, 1626. [[CrossRef](#)]
73. Guimarães, C.J.B.V.; Fernandes, M.A.C. Real-time Neural Networks Implementation Proposal for Microcontrollers. *Electronics* **2020**, *9*, 1597. [[CrossRef](#)]
74. Sakr, F.; Bellotti, F.; Berta, R.; De Gloria, A. Machine Learning on Mainstream Microcontrollers. *Sensors* **2020**, *20*, 2638. [[CrossRef](#)]
75. Druin, A. Cooperative Inquiry: Developing New Technologies for Children with Children. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Pittsburgh, PA, USA, 15–20 May 1999; Association for Computing Machinery: New York, NY, USA, 1999; pp. 592–599. [[CrossRef](#)]
76. Druin, A. *Mobile Technology for Children: Designing for Interaction and Learning*; Morgan Kaufmann: Cambridge, MA, USA, 2009.

77. Macefield, R. How to specify the participant group size for usability studies: A practitioner's guide. *J. Usability Stud.* **2009**, *5*, 34–45.
78. O'Brien, H.L.; Toms, E.G. The development and evaluation of a survey to measure user engagement. *J. Am. Soc. Inf. Sci. Technol.* **2010**, *61*, 50–69. [CrossRef]
79. Schmitt, K.L.; Anderson, D.R. Television and reality: Toddlers' use of visual information from video to guide behavior. *Media Psychol.* **2002**, *4*, 51–76. [CrossRef]
80. Faulkner, L. Beyond the five-user assumption: Benefits of increased sample sizes in usability testing. *Behav. Res. Meth. Instrum. Comput.* **2003**, *35*, 379–383. [CrossRef]
81. Nielsen, J. Nielsen Norman Group. Why You Only Need to Test with 5 Users. 2000. Available online: <https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/> (accessed on 10 April 2023).
82. Soleimani, A.; Green, K.E.; Herro, D.C.; Walker, I.D.; Gardner-McCune, C. CyberPLAYce, A Cyber-Physical-Spatial Storytelling Tool: Results from an Empirical Study with 8–10-Year-Old Storytellers. In *Lecture Notes in Computer Science, Proceedings of the Second International Conference, LCT 2015, Held as Part of HCI International 2015, Los Angeles, CA, USA, 2–7 August 2015*; Zaphiris, P., Ioannou, A., Eds.; Springer: Berlin, Germany, 2015; Volume 9192, pp. 438–446.
83. Hussain, A.; Mkpjojogu, E.; Kamal, F.; Lateef, H. Ascertaining the UX of the Word Mania Mobile App for Children using Fun Toolkit v3. *Int. J. Recent Technol. Eng.* **2019**, *8*, 1–4. [CrossRef]
84. Tippett, C.D.; Milford, T.M. Findings from a pre-kindergarten classroom: Making the case for STEM in early childhood education. *Int. J. Sci. Math. Educ.* **2017**, *15*, 67–86. [CrossRef]
85. Hourcade, J.P. *Child-Computer Interaction*; University of Iowa: Iowa City, IA, USA, 2015.
86. Khanum, M.A.; Trivedi, M.C. Take care: A study on usability evaluation methods for children. *arXiv* **2012**, arXiv:1212.0647.
87. Read, J.C. Validating the Fun Toolkit: An instrument for measuring children's opinions of technology. *Cogn. Technol. Work* **2008**, *10*, 119–128. [CrossRef]
88. Chou, P.N.; Chen, W.F.; Wu, C.Y.; Carey, R.P. Utilizing 3D open source software to facilitate student learning of fundamental engineering knowledge: A quasi-experimental study. *Int. J. Eng. Educ.* **2017**, *33*, 382–388.
89. Gecu-Parmaksiz, Z.; Delialioglu, O. Augmented reality-based virtual manipulatives versus physical manipulatives for teaching geometric shapes to preschool children. *Br. J. Educ. Technol.* **2019**, *50*, 3376–3390. [CrossRef]
90. Kesicioglu, O.S.; Mart, M. The Preschool Teachers' Opinion on Teaching Geometry. *Southeast Asia Early Child. J.* **2022**, *11*, 21–36.
91. Laski, E.V.; Siegler, R.S. Learning from number board games: You learn what you encode. *Dev. Psychol.* **2014**, *50*, 853. [CrossRef]
92. Clements, D.H.; Sarama, J. Early childhood mathematics intervention. *Science* **2011**, *333*, 968–970. [CrossRef]
93. Garcia-Sanjuan, F.; Jurdi, S.; Jaen, J.; Nacher, V. Evaluating a tactile and a tangible multi-tablet gamified quiz system for collaborative learning in primary education. *Comput. Educ.* **2018**, *123*, 65–84. [CrossRef]
94. Yilmaz, R.M. Educational magic toys developed with augmented reality technology for early childhood education. *Comput. Hum. Behav.* **2016**, *54*, 240–248. [CrossRef]
95. Baauw, E.; Markopoulous, P. A comparison of think-aloud and post-task interview for usability testing with children. In *Proceedings of the 2004 Conference on Interaction Design and Children: Building a Community*, Baltimore, MD, USA, 1–3 June 2004; pp. 115–116.
96. Mertala, P. Young children's perceptions of ubiquitous computing and the Internet of Things. *Br. J. Educ. Technol.* **2020**, *51*, 84–102. [CrossRef]
97. Rico-Olarte, C.; López, D.; Narváez, S.; Farinango, C.; Pharow, P. HapHop-Physio: A computer game to support cognitive therapies in children. *Psychol. Res. Behav. Manag.* **2017**, *10*, 209–217. [CrossRef]
98. Radeta, M.; Cesário, V.; Matos, S.; Nisi, V. Gaming Versus Storytelling: Understanding Children's Interactive Experiences in a Museum Setting. In *Lecture Notes in Computer Science*; Springer: Berlin/Heidelberg, Germany, 2017; pp. 163–178. [CrossRef]
99. El Jurdi, S.; Garcia-Sanjuan, F.; Nacher, V.; Jaen, J. Children's Acceptance of a Collaborative Problem Solving Game Based on Physical Versus Digital Learning Spaces. *Interact. Comput.* **2018**, *30*, 187–206. [CrossRef]
100. Read, J.; MacFarlane, S.; Casey, C. Endurability, engagement and expectations: Measuring children's fun. In *Interaction Design and Children*; Shaker Publishing: Eindhoven, The Netherlands, 2009; Volume 2, pp. 1–23.
101. Sim, G.; Horton, M. Investigating children's opinions of games: Fun Toolkit vs. This or That. In *Proceedings of the 11th International Conference on Interaction Design and Children*, Bremen, Germany, 12–15 June 2012; pp. 70–77. [CrossRef]
102. Prakoso, I.; Purnomo, H. Innovative design of the combined rocking horse toy and folding chair for children. *Int. J. Adv. Sci. Eng. Inf. Technol.* **2019**, *9*, 1584–1591. [CrossRef]
103. Zhang-Kennedy, L.; Abdelaziz, Y.; Chiasson, S. Cyberheroes: The design and evaluation of an interactive ebook to educate children about online privacy. *Int. J. Child-Comput. Interact.* **2017**, *13*, 10–18. [CrossRef]
104. Zhang-Kennedy, L.; Chiasson, S. Teaching with an Interactive E-Book to Improve Children's Online Privacy Knowledge. In *Proceedings of the The 15th International Conference on Interaction Design and Children, IDC '16, Manchester, UK, 21–24 June 2016*; Association for Computing Machinery: New York, NY, USA, 2016; pp. 506–511. [CrossRef]
105. Mostowfi, S.; Mamaghani, N.; Khorramar, M. Designing playful learning by using educational board game for children in the age range of 7–12: (A case study: Recycling and waste separation education board game). *Int. J. Environ. Sci. Educ.* **2016**, *11*, 5453–5476.

106. Van Dijk, E.M.; Lingnau, A.; Kockelkorn, H. Measuring enjoyment of an interactive museum experience. In Proceedings of the 14th ACM International Conference on Multimodal Interaction, Santa Monica, CA, USA, 22–26 October 2012; pp. 249–256.
107. Roussou, M.; Katifori, A. Flow, Staging, Wayfinding, Personalization: Evaluating User Experience with Mobile Museum Narratives. *Multimodal Technol. Interact.* **2018**, *2*, 32. [[CrossRef](#)]
108. Höysniemi, J.; Hämäläinen, P.; Turkki, L. Using peer tutoring in evaluating the usability of a physically interactive computer game with children. *Interact. Comput.* **2003**, *15*, 203–225. [[CrossRef](#)]
109. Markopoulos, P.; Read, J.C.; MacFarlane, S.; Höysniemi, J. (Eds.) *Evaluating Children's Interactive Products*; Interactive Technologies, Morgan Kaufmann: Burlington, VT, USA, 2008; pp. 343–354. [[CrossRef](#)]

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.



## **APPENDIX B**

**Paper Title:**

Privacy leakage of LoRaWAN smart parking occupancy sensors

**Abstract:**

Development of smart cities is enabled by its core concepts of smart and sustainable mobility, where Low Power Wide Area Network (LPWAN) such as Long Range Wide Area Network (LoRaWAN) became one of the most important Internet of Things (IoT) technologies. Due to its low power consumption, simple setup, and large communication range, LoRaWAN smart parking devices are already employed to reduce congestion and improve quality of life. This paper studies privacy leakage of LoRaWAN smart parking communication devices. Namely, when a vehicle as a metallic obstacle obscures the LoRaWAN smart parking device, the signal strength will be significantly reduced on the receiver side. Consequently, the variation in the signal strength of LoRaWAN parking systems transmits information about parking space occupancy, allowing the implementation of a passive side-channel attack at large distances. Using supervised machine learning techniques based on Neural Network, the attacker can estimate parking lot occupancy with accuracy up to 97%, while Random Forrest approach reaches the accuracy over 98%.

**Cite as:**

Rodić, Lea Dujić, et al. "Privacy leakage of LoRaWAN smart parking occupancy sensors." *Future Generation Computer Systems* 138 (2023): 142-159.





**DOI:**

<https://doi.org/10.1016/j.future.2022.08.007>

## **APPENDIX C**

Article

# Tag Estimation Method for ALOHA RFID System Based on Machine Learning Classifiers

Lea Dujic Rodic \* , Ivo Stančić \* , Kristina Zovko , Toni Perković  and Petar Šolić 

Faculty of Electrical Engineering, Mechanical Engineering and Naval Architecture, University of Split,  
32 Rudjera Boskovic, 21000 Split, Croatia

\* Correspondence: dujic@fesb.hr (L.D.R.); istancic@fesb.hr (I.S.)

**Abstract:** In the last two decades, Radio Frequency Identification (RFID) technology has attained prominent performance improvement and has been recognized as one of the key enablers of the Internet of Things (IoT) concepts. In parallel, extensive employment of Machine Learning (ML) algorithms in diverse IoT areas has led to numerous advantages that increase successful utilization in different scenarios. The work presented in this paper provides a use-case feasibility analysis of the implementation of ML algorithms for the estimation of ALOHA-based frame size in the RFID Gen2 system. Findings presented in this research indicate that the examined ML algorithms can be deployed on modern state-of-the-art resource-constrained microcontrollers enhancing system throughput. In addition, such utilization can cope with latency since the execution time is sufficient to meet protocol needs.

**Keywords:** Internet of Things; RFID tags, RFID reader; Machine Learning; tag estimate method; microcontroller



**Citation:** Dujic Rodic, L.; Stančić, I.; Zovko, K.; Perković, T.; Šolić, P. Tag Estimation Method for ALOHA RFID System Based on Machine Learning Classifiers. *Electronics* **2022**, *11*, 2605. <https://doi.org/10.3390/electronics11162605>

Academic Editors: Cristina Stojescu-Crisan and Alexandru Isar

Received: 27 June 2022

Accepted: 17 August 2022

Published: 19 August 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

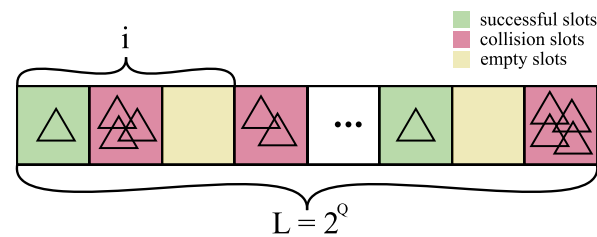
## 1. Introduction

The Internet of Things (IoT) has become a pervasive environment in which smart objects interact and exchange information by sensing the ambiance of their surroundings. One of the major technologies that enable IoT is Radio Frequency Identification (RFID), which can utilize Wireless Information and Power Transfer (WIPT) in its applications that include access control, parking management, logistics, retail, etc. [1]. In large-scale infrastructures, such as commercial warehouses, reading RFID tags, such as ultra-high-frequency ones, comes at a high cost and can involve a large volume of data [2]. In general, RFID presents radio technology that acts as a communication medium between a reader and the tag, with a unique identifier used for communication [3]. In general, the RFID tag is distinguished by the presence or absence of the battery [4]. Passive tags are self-powered and communicate using the same RF waves emitted by the reader antennas, known as backscattering technology [5]. Among the existing technologies, passive Gen2 technology is considered the most attractive in IoT applications due to its simple design, flexibility, cost and performance [6,7]. Gen2 as a standard is used on the physical and MAC levels to establish reliable communication between the reader and a tag. Readers must provide sufficient power to energize tags and respond to the necessary information since they are not equipped with batteries. The energy levels that tags can extract are quite low and, therefore, cannot afford energy-efficient MAC schemes [8]. In general, the MAC of RFID is random, and there are two widely used methods to achieve it: the first is a binary tree, and the second is the ALOHA-based protocol [4]. In the binary tree protocol, continuous YES/NO communication is achieved between a reader and tags, while with the ALOHA protocol tag initiates communication with a request from the reader [9–11].

One of the commonly utilized ALOHA-based protocols is the Dynamic Framed Slotted ALOHA (DFSA) since it has the most prominent performance, which is the highest throughput. DFSA belongs to a group of time division multiple access (TDMA) protocols,



where communication between a reader and tags is divided into time frames, which are, in turn, divided into time slots [12]. The beginning of the interrogation process in DFSA is induced by the reader's announcement of the frame size [6]. This is performed by the reader sending a QUERY command and the value of the main protocol parameter  $Q$  for the tags [13]. The value of  $Q$  is an integer ranging from 0 to 15 that sets the frame size at  $L = 2^Q$ . From there on, all tags that are being interrogated will occupy a randomly selected position in the frame (commonly known as a slot) and will onward reply back to the reader when their slot is being interrogated. Based on such an access control scheme, three diverse scenarios may happen: (a) only one tag is in the slot (the successful slot), (b) no tags in the slot (empty slot) and (c) numerous tags have taken the same spot (collision) [6]. The overall number of successful, empty and collision slots is denoted with  $S$ ,  $E$  and  $C$ , respectively. An example of an interrogating frame is exhibited in Figure 1.



**Figure 1.** An example of an interrogating frame of frame size  $L = 2^Q$ .  $i$  represents the size of a particular part of the frame.

Therefore, the frame size is equal to the sum of successful, empty and collision slots, i.e.,  $L = E + S + C$ . According to the previous notation, the throughput is defined using Equation (1) as :

$$\eta = \frac{S}{L}. \quad (1)$$

therefore, the main goal in DFSA systems is to increase the number of successful slots  $S$  within the frame size  $L$ . As tags are fitting their slots randomly, previous studies [14] show that the maximum throughput will reach its maximum value of approximately 37% when the size of the frame equals the number of tags. In usual situations, the number of tags is unknown and has to be estimated in order to set an adequate frame size and, consequently, achieve maximum throughput.

Aiming to improve the throughput of RFID systems, the research presented in this paper utilizes Machine Learning classifiers as an approach for efficient tag number estimation. The performance of the ML algorithms is compared with state-of-the-art solutions, specifically the Improved Linearized Combinatorial model (ILCM) for tag estimation (elaborated in [8]). The study presented in this paper shows that ML classifiers, which use the maximum of the available information gathered from Monte Carlo simulations, can be implemented in standard, mobile RFID readers. What is more, they outperform the state-of-the-art model achieving better throughput. The advances result in achieving optimal performance in tag identification by the reasonable processing time and energy requirements. The paper is structured as follows: Section 2 gives an insight into some commonly applied methods for tag number estimation and provides more detail on the ILCM model. Section 3.1 gives details on the experimental setup and elaboration on the Machine Learning models and the ILCM model used on a particular set of data, continuing with model performance analyses and comparison. Section 5 examines the feasibility and mandatory means for the efficient deployment of ML models on microcontrollers. Overall, an articulate conclusion that emphasizes the obtained results is provided in Section 6.

## 2. Related Works

Over the past few years, various methods and approaches have been employed for tag estimation. In [15], Vogt presented a method based on Minimum Squared Error

(MSE) estimation by minimizing the Euclidean norm of the vector difference between the actual frame statistics and their expected values. The number of empty, successful and collision slots is taken into account. However, the predicted values are calculated under the assumption that the tags in each slot have independent binomial distributions, which leads to unreliable findings. In the research presented by Chen in [14], the authors apply probabilistic modeling of the tag distribution within the frame, which they consider to be a multinomial distribution. By doing so, they obtain the tag number estimation. For each slot, binomial distributions provide occupancy information. However, it does not consider the fact that the number of tags in the interrogation area is limited [16]. An improvement of the previous model was given by research in [17], although this improvement tends to have a high computational cost of implementation for genuine RFID systems [16]. Furthermore, a study presented in [18] offered a unique tag number estimation scheme termed ‘Scalable Minimum Mean Square Error’ (sMMSE), which enhanced accuracy and reduced estimation time. The efficient modification of the frame size is derived from two principal parameters: the first one puts a limit on the slot occupancy, whereupon frame size needs to be expanded, and the second determines the frame size expansion factor. In the research presented in [19], the authors provided an in-depth analysis of some of the most relevant anti-collision algorithms, taking into account the limitations imposed by EPCglobal Class-1 Gen-2 for passive RFID systems. The study classified and compared some of the most important algorithms and optimal frame-length selection. Based on their research results, the authors point out that the maximum-likelihood algorithms achieve the best performance in terms of mean identification delay. Finally, the researchers concluded that the algorithm performance also depends on the computation time for estimating the number of tags.

A study presented in [20] introduced a new MFML-DFSA anti-collision protocol. In order to increase the accuracy of the estimate, it uses a maximum-likelihood estimator that makes use of statistical data from many frames (multiframe estimation). The algorithm chooses the ideal frame length for the following reading frame based on the anticipated number of tags, taking into account the limitations of the EPCglobal Class-1 Gen-2 standard. The MFML-DFSA algorithm outperforms earlier suggestions in terms of average identification time and computing cost (which is lower), making it appropriate for use in commercial RFID readers. The rather novel research given in [21] proposes an RFID tag anti-collision method that applies adjustable frame length modification. The original tag number is estimated based on the initial assumption that the number of tags identified in the first frame is known. The authors present a non-linear transcendental equation-based DFSA (NTEBD) algorithm and compare it to the ALOHA algorithm demonstrating the error rate for experimental results to be less than 5% and improved tag identification throughput by 50%. The authors of [22] present an extension for an anti-collision estimator based on a binomial distribution. They have constructed a simulation module to examine estimator performance in diverse scenarios and have shown that the proposed extension has enhanced performance in comparison to other estimators, no matter if the number of tags is 1000 or 10,000.

As can be observed, the previously mentioned algorithms tend to have high computational costs since they are commonly funded by calculating probabilities. This might present an issue for standard microprocessors that are not adjusted to perform computations of factorials that produce large numbers. To solve the issue, a diverse method for tag estimation has been introduced by researchers in [8], namely the Improved Linearized Combinatorial model. Their approach bypasses the conditional probability calculations by doing them in advance. Further, the estimation model is uncomplicated and provides an effective tag estimation  $\hat{n}$  based on linear interpolation given by Equation (2):

$$\hat{n} = kS + L, \quad (2)$$

where coefficients  $k$  and  $l$  are derived from Equations (3) and (4), respectively:

$$k = \frac{C}{(4.344L - 16.28) + \left(\frac{L}{-2.282 - 0.273L}\right) \cdot C} + 0.2407 \cdot \ln(L + 42.56), \quad (3)$$

$$L = (1.2592 + 1.513L) \tan(1.234L^{-0.9907} \cdot C). \quad (4)$$

In the event of no collision, the formula gives  $\hat{n} = S$ , whereas for cases when  $k < 0$ ,  $k$  must be set to 0. Following the tag estimation, the  $Q$  value is calculated using Equation (5) as

$$Q = \text{round}(\log_2(\hat{n} - S)). \quad (5)$$

The results obtained by the authors have indicated that the ILCM shows comparable behavior to state-of-the-art algorithms regarding the identification delay (slots) but is not computationally complex. An extension of their study was performed in [23] by presenting a C-MAP anti-collision algorithm for an RFID system that has lower memory demands.

### 3. Materials and Methods

#### 3.1. Machine Learning Classifiers for Tag Estimation

The IoT surroundings rich with sensor devices that are interconnected have also yielded the demand for the more efficient monitoring of sensor activities and events [24]. To support diverse IoT use-case scenarios, Machine Learning has emerged as an essential area of scientific study and employment to enable computers to automatically progress through experience [25]. Commonly, ML incorporates data analyses and processing followed by training phases to produce “a model”, which is onward tested. The overall goal is to expedite the system to act based on the results and inputs given within the training phase [26]. For the system to successfully achieve the learning process, distinctive algorithms and models along with data analyses are employed to extract and gain insight into data correlation [27]. Thus far, Machine Learning has been fruitfully applied to various problems such as regression, classification and density estimation [28]. Specific algorithms are universally split into disjoint groups known as Unsupervised, Supervised, Semi-supervised and Reinforcement algorithms. The selection of the most appropriate ML algorithm for a specific purpose is performed based on its speed and computational complexity [27].

Machine Learning applications range from prediction, image processing, speech recognition, computer vision, semantic analysis, natural language processing, as well as information retrieval [29]. In a problem like the estimation of the tag number based on the provided input, one must consider the most applicable classifiers that can deal with a particular set of data. Currently, classification algorithms have been applied for financial analyses, bio-informatics, face detection, handwriting recognition, image classification, text classification, spam filtering, etc. [30]. In a classification problem, a targeted label is generally a bounded number of discrete categories, such as in the case of estimating the number of tags. State-of-the-art algorithms for classification incorporate Decision Tree (DT), k-Nearest Neighbour ( $k$ -NN), Support Vector Machine (SVM), Random Forest (RF) and Bayesian Network [31]. In the last decade, Deep Learning (DL) has manifested itself as a novel ML technique that has efficiently solved problems that have not been overcome by more traditional ML algorithms [26]. Considered to be one of the most notable technologies of the decade, DL utilizations have obtained remarkable accuracy in various fields such as image and audio-related domains [26,32]. DL has the remarkable ability to discover the complex configuration of large datasets using a backpropagation algorithm indicating in what manner the machine’s internal parameters need to be altered to calculate and determine each layer’s representation based on the representation of previous layers [33]. The essential principle of DL has been displayed throughout growing research performed in Neural Networks or Artificial Neural Networks (ANN). This approach

allows for a layered structure of concepts with multiple levels of abstraction, in which every layer of concepts is made from some simpler layers [26]. Deep Learning has made improvements in problem-solving with regards to discovering intricate configurations of large-sized data and thus has been applied in various domains ranging from image recognition, speech recognition, natural language understanding, sentiment analysis and many more [33]. Machine Learning classifier performance has been extensively analyzed in the last decade [34–37], providing a systematical insight into the classifiers' key attributes. Table 1 provides a comparison of the advantages and limitations of commonly utilized ML classifiers.

**Table 1.** The advantages and limitations of commonly utilized ML classifiers [34–37].

ML Classifier	Advantages	Limitations
DT	Solves multi-class and binary problems; Fast Can handle missing values; Easily interpretable	Prone to overfitting; Sensitive to outliers
<i>k</i> -NN	Solves multi-class and binary problems; Easy to implement	Sensitive to noisy attributes; Poor interpretability Slow to evaluate large training sets
SMV	Solves binary problems; High accuracy  Durable to Noise; Excellent to model non-linear relations	Training is slow; High complexity and memory requirements
RF	Solves multi-class and binary problems; Higher accuracy compared to other models Robust to noise	Can be slow for real-time predictions; Not very interpretable
Naive Bayes	Solves multi-class and binary problems; Simple to implement; Fast	Ignores underlying geometry of data; Requires predictors to be independent
ANN	Solves multi-class and binary problems; Handles noisy data Detects non-linear relation amongst data; Fast	Prone to over-fitting on small datasets; Computationally intensive

Tag number estimation can be regarded as a multi-class classification problem. Amongst many classifiers, Random Forest has been considered a simple yet powerful algorithm for classification, successfully applied in numerous problems such as image annotation, text classification, medical data etc. [38]. RF has been proven to be very accurate when dealing with large datasets; it is robust to noise and has a parallel architecture that makes it faster than other state-of-the-art classifiers [39]. Furthermore, it is also very efficient in stabilizing classification errors when dealing with unbalanced datasets [40]. On the other hand, Neural Networks offer great potential for multi-class classification due to their non-linear architecture and prominent approximation proficiency to comprehend tangled input–output relationships between data [41].

Discriminative models, such as Neural Networks and Random Forest, can model the decision boundary between the classes [42], thus providing vigorous solutions for non-linear discrimination in high-dimensional spaces [43]. Therefore, their utilization for classification proposes has proven to be successful and efficient [44]. Both algorithms are able to model linear as well as complex non-linear relationships. However, Neural Networks have a greater potential here due to their construction [45]. On the other hand, RF outperforms NN in arbitrary domains, particularly in cases when the underlying data sizes are small, and no domain-specific insight has been used to arrange the architecture of the underlying NN [21]. Although NN is an expressively rich tool for complex data modeling, they are prone to overfitting on small datasets [46] and are very time-consuming and computationally intensive [45]. Furthermore, their performance is frequently sensitive to the specific architectures used to arrange the computational units [21] in contrast to

the computational cost and time of training a Random Forest, requiring much less input preparation [45]. Finally, although RF needs less hyper-parameter tuning than NN, the acquisitive feature space separation by orthogonal hyper-planes produces typical stair or box-like decision surfaces, which can be beneficial for some datasets but sub-optimal for others [46].

Following the stated reasoning, Neural Network and Random Forest have been applied in this research for tag number estimation based on the scenarios that occur during slot interrogation.

### 3.1.1. Experimental Setup

To obtain valuable data for model comparison, Monte Carlo simulations were performed to produce an adequate number of possible scenarios that may happen during the interrogation procedure in DFSA. Detailed elaboration of the mathematical background of the Monte Carlo method that has been applied for this research has been elaborated in research prior to this one and presented in [11]. The selected approach for Monte Carlo simulations of the distribution of tags in the slots follows the research performed in studies [8] and [23]. Simulations were executed for frame sizes  $L = 4, 8, 16, 32, 128$  and  $256$ , i.e., for  $Q = 2, 3, 4, 5, 7$  and  $8$ , where the number of tags was in the range of  $[1, 2^{Q+2}]$  (this range was chosen based upon experimental findings elaborated in [23]). For each of the frame sizes and the number of tags, random 100,000 distributions of  $E$  empty slots, successful slots  $S$  and collision slots  $C$  were realized and are presented in Table 2.

**Table 2.** Snapshot of the obtained data.

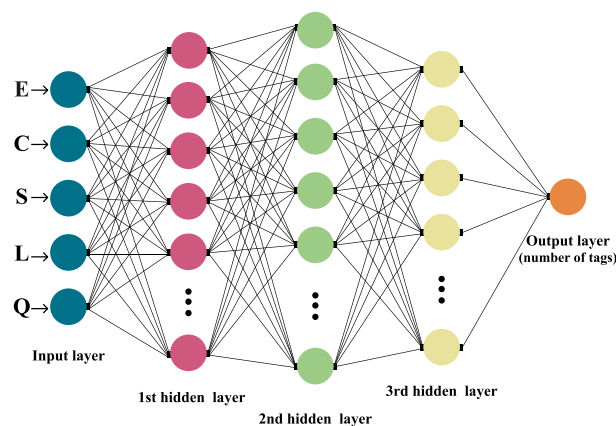
Q	L	S	C	E	N (Number of Tags)
2	4	2	1	1	6
2	4	0	3	1	15
...	...	...	...	...	...
8	256	79	122	55	401
8	256	18	229	9	943

Data obtained from the simulations were given to Neural Network, Random Forest and the ILCM models for adequate performance comparison and analyses of the accuracy of tag estimation.

All of the models and simulations were performed on a dedicated computer for such tasks. To be precise, the machine has an Intel(R) Core(TM) i7-7700HQ@2.80 GHz processor, 16 GB of RAM and NVIDIA GeForce GTX 1050 Ti with 768 existing cores running on a 64-bit Windows 10 operating system. Furthermore, to realize more efficient computing with the GPU, the Deep Neural Network NVIDIA CUDA library (cuDNN) [47] was applied. The Keras 2.3.1. Python library was employed, which operates on top of a source built upon Tensorflow 2.2.0 with CUDA support for different batch sizes.

### 3.1.2. Neural Network Model

In general, a Neural Network is made out of an artificial neuron and layer: the input, the hidden layer (or layers) and the output layer are all interconnected [26], as exhibited in Figure 2.



**Figure 2.** Architecture of the Neural Network model.

Aiming to imitate the behavior of real biological neurons, the course of learning within a NN unfolds through uncovering hidden correlations amongst the sequences of input data throughout layers of neurons. The outputs from neurons in one layer are onward given as inputs to the neurons in the next layer. A formal mathematical definition of an artificial neuron given by Equation (6) is as follows.

**Definition 1.** An artificial neuron  $l_i$  is the output of the non-linear mapping  $\theta$  applied to a weighted sum of input values  $x$  and a bias  $\beta$  defined as :

$$l_i(x) = \theta(\omega_i x + \beta), \quad (6)$$

where  $\omega_i$  represents the matrix of weights and is called an artificial neuron.

The weights are appointed considering the inputs' relative significance to the other inputs, and the bias ensures a consistent value is added to the mapping to ensure successful learning [48]. Generally, the mapping  $\theta$  is known as the activation (or transfer) function. Its purpose is to keep the amplitude of the output of the neuron in an adequate range of  $[0, 1]$  or  $[-1, 1]$  [49]. Although activation functions may be linear and non-linear, usually the non-linear ones are more frequently utilized. The most recognizable ones are ordinary Sigmoids or the Softmax function, such as the hyperbolic tangent  $\Phi(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$ , in contrast to Rectified Linear Unit (ReLU) function:  $\theta(a) = \max(0, a)$  [33]. The selection of a particular activation function is based on the core problem to be solved by applying the Neural Network [50].

The architecture of the NN model displayed in this research is constructed of five layers, as depicted in Figure 2. The first one is the input layer, followed by three hidden layers (one Dropout layer), and the final is the output layer. Applied activation functions were ReLU (in hidden layers) and Softmax (within the output layer). Data used for the input layer were number  $Q$ , frame size  $L$  and the number of  $S$  successful,  $E$  empty and  $C$  collision slots. The number of tags that are associated with a particular distribution of slots within a frame is classified in the final exit layer.

The data were further partitioned in a 70% : 30% ratio, with 70% of the data used for training and the other 30% for testing, with the target values being the number of tags, and all other values were provided as input. The training data were pre-processed and normalized, whereas target values were coded with One Hot Encoded with Keras library for better efficiency. By doing so, the integer values of the number of tags are encoded as binary vectors. The dropout rate (probability of setting outputs from the hidden layer to zero) was specified to be 20%. The number of neurons varies based on the frame size, ranging from 64 to 1024 for the first four layers.

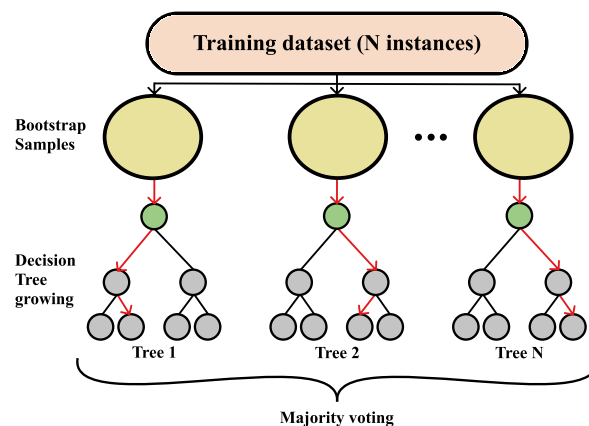
Since the classification of the number of tags is a multi-class classification problem, for this research, the Categorical Cross-Entropy Loss function was applied as the loss

(cost) function with several optimizer combinations. Another important aspect of the NN model architecture was thoroughly examined, and that is the selection of optimizers and learning rates. Optimizers attempt to help the model converge and minimize the loss or error function, whereas the learning rate decides how much the model needs to be altered in response to the estimated error every time the model weights are updated [48]. Tested optimizers were Root Mean Square Propagation (RMSProp), Stochastic Gradient Descent (SGD) and Adaptive Moment Optimization (Adam). Adam provided the most accurate estimation results and was onward utilized in the learning process with 100 epochs and a 0.001 learning rate.

### 3.1.3. Random Forest Model

At the beginning of this century, L. Breiman proposed the Random Forest algorithm, an ensemble-supervised ML technique [51]. Today, RF is established as a commonly utilized non-parametric method applied for both classification and regression problems by constructing prediction rules based on different types of predictor variables without making any prior assumption of the form of their correlation with the target variable [52]. In general, the algorithm operates by combining a few arbitrary decision trees and onward aggregating their predictions by averaging. Random Forest has been proven to have exceptional behavior in scenarios where the amount of variables is far greater than the number of observations and can have good performance for large-scale problems [53]. Studies have shown RF to be a very accurate classifier in different scenarios, it is easily adapted for various learning tasks, and one of its most recognizable features is its robustness to noise [54]. That is why Random Forest has been used for numerous applications such as bioinformatics, chemoinformatics, 3D object recognition, traffic accident detection, intrusion detection systems, computer vision, image analysis, etc. [11,53].

A more formal definition of Random Forest is as follows. A Random Forest classifier is a collection made out of tree-structured classifiers, namely  $\{r(x, \Theta_k), k = 1, \dots, L\}$ , where  $\Theta_k$  are independent random vectors that are identically distributed for an input  $x$ , every tree will toss a unit vote for the most favored class [55], as shown in Figure 3.



**Figure 3.** Visualization of the Random Forest classification process.

The bagging approach is used for producing the tree, i.e., by generating random slices of the training sets using substitution, which means that some slices can be selected more than once and others not at all [56]. Given a particular training set  $S$ , generated classifiers  $\{r(x, \Theta_k)\}$  toss a vote, thus making a *bagged* predictor and, at the same time, for every pair  $y, x$  from the training set and for every  $\Theta_k$  that did not contain  $y, x$ , votes from  $\Theta_k$  are set aside as *out-of-the-bag* classifiers [51]. Commonly, a partition of samples is on the training set by taking two-thirds for tree training and leaving one-third for inner cross-validation, thus removing the need for cross-validation or a separate test set [56]. The user is the one defining the number of trees and other hyper-parameters that the algorithm uses for independent tree creation performed without any pruning, where the key is to have a low

bias and high variance, and the splitting of each node is based on a user-defined number of features that are randomly chosen [52]. In the end, the final classification is obtained by majority vote, i.e., the instance is classified into the class having the most votes over all trees in the forest [54].

Aiming to produce the best classification accuracy, in this research, hyper-parameter tuning has been performed by utilizing the *GridSearchCV* class from the *scikit-learn* library with five-fold cross-validation.

This is performed following the above reasoning for making a structure for each particular tree. By controlling the hyper-parameters, one can supervise the architecture and size of the forest (e.g., the number of trees (*n\_estimators*)) along with the degree of randomness (e.g., *max\_features*) [52]. Therefore, for every frame size, the hyper-parameters presented in Table 3 were tested, resulting in a separate RF model for each of the frame sizes, as presented in Table 4.

**Table 3.** Tested Hyper-parameters for Random Forest.

Hyper Parameter	Values
<i>n_estimators</i>	50, 100, 200, 500
<i>criterion</i>	gini, entropy
<i>max_depth</i>	3, 5, 10, 20
<i>max_features</i>	auto, sqrt
<i>min_samples_split</i>	2, 4, 6, 10

**Table 4.** Grid search results of RF Hyper-parameters for a particular frame size.

Frame Size	<i>n_Estimators</i>	Criterion	<i>max_Depth</i>	<i>max_Features</i>	<i>min_Samples_Plit</i>
L = 4	50	gini	5	auto	2
L = 8	50	gini	5	auto	2
L = 16	100	gini	10	auto	4
L = 32	100	entropy	20	sqrt	2
L = 128	500	gini	20	sqrt	2
L = 256	200	gini	20	sqrt	4

#### 4. Results and Comparison

For ILCM, Neural Network and Random Forest, the same data were used to make a comprehensive performance comparison. To provide a comprehensive classifier performance comparison, several measures were taken into account. First, to compare the performance of each classifier as a Machine Learning model, the accuracy measure was taken (since it is a standard metric for evaluation of a classifier), this being the categorical accuracy. Categorical accuracy is a Keras built-in metric that calculates the result by finding the largest percentage from the prediction and then compares it to the actual result. If the largest percentage matches the index of 1, then the measured accuracy increases. If it does not match, the accuracy goes down. Our experimental results point out that RF has out-performed the NN model in the classification task, as shown in Table 5, but this measure alone is not enough to determine which of the two ML models would be preferred for utilization in the scenario of tag estimation. Therefore, due to the nature of the problem of tag estimation, we have considered Mean Absolute Errors (MAE) and Absolute Errors (AE) as measures of classifier performance (see Equations (7) and (8), respectively). An accumulated estimation error will degrade the whole performance [57], meaning that the overall smaller MAE and AE for a classifier would determine the overall estimator efficiency, i.e., better system throughput. For the approximated number of tags  $\hat{n}$  and the exact number of tags  $n$ , MAE is defined as:

$$MAE = \frac{1}{m} \sum_{i=1}^m |\hat{n}^{(i)} - n^{(i)}|. \quad (7)$$



For every frame size, AE was calculated as:

$$AE = |n - \hat{n}|. \quad (8)$$

**Table 5.** Classification accuracy of NN, RF and the ILCM model for a particular frame size.

Frame Size	ACCURACY		
	NN	ILCM	RF
L = 4	33.54%	23.55%	33.59%
L = 8	28.56%	27.28%	28.22%
L = 16	24.05%	23.27%	24.37%
L = 32	19.78%	17.06%	19.54%
L = 128	11.25%	4.42%	12.12%
L = 256	5.74%	2.8%	9.46%

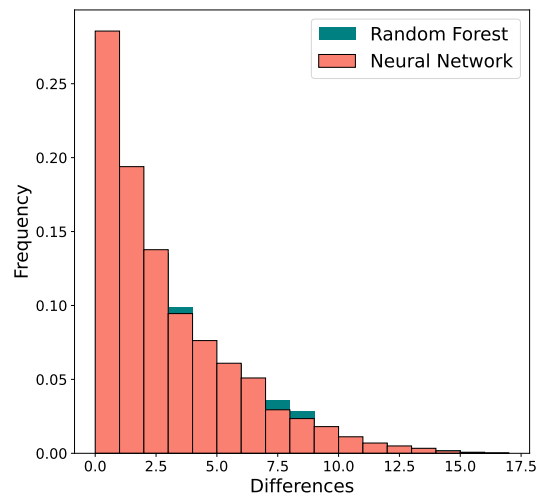
As can be noticed from Table 5, as frame size rises, the accuracy decreases for all of the three compared models. Furthermore, Random Forest seems to outperform other classifiers for the most challenging task for  $L = 256$ . Furthermore, ILCM performed similarly to NN and RF for smaller frame sizes.

On the other hand, the results presented in Table 6 indicate that the Neural Network model produces error rates comparable to RF, although RF has better accuracy. What is more, for the largest frame size, NN will have an overall smaller MAE, as can be seen from Table 6 for frame sizes  $L = 128$  and  $L = 256$ . Overall, both Machine Learning classifiers perform substantially better than the ILCM model.

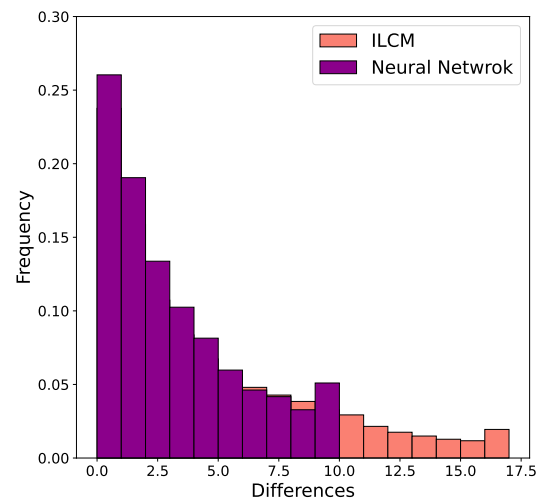
**Table 6.** MAE of NN, RF and the ILCM model for a particular frame size.

Frame Size	MAE		
	NN	ILCM	RF
L = 4	2.23	2.182	2.23
L = 8	2.56	2.61	2.5
L = 16	3.57	4.31	3.69
L = 32	5.23	6.98	5.324
L = 128	11.27	17.38	11.93
L = 256	16.06	27.29	18.19

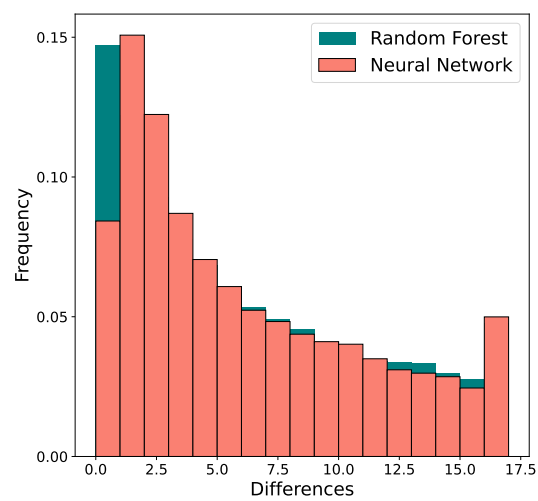
This observation is further emphasized in the calculations of Absolute Errors of classification for RF, NN and the ILCM model. AE was derived for every frame size, and the histograms presented in Figure 4 provide a pictorial comparison of the errors. As can be seen from Figure 4a, for smaller frame sizes, the NN model performs quite complementary to the RF model, but for the largest frame size, the NN (see Figure 4c) will have an overall smaller AE. These histograms are consistent with the MAE results from Table 6, confirming that the NN classifies values  $\hat{n}$  nearer to the true values of the number of tags  $n$ . This observation is important for estimating the length of the next frame because the closer the estimated number of interrogating tags is to the actual number of tags, the better the frame size setting. Incorrect estimates of the total number of tags result in lower throughput. Results from this analysis show that, in comparison to the RF model, the NN model is generally “closer” to the real tag number.



(a)



(b)

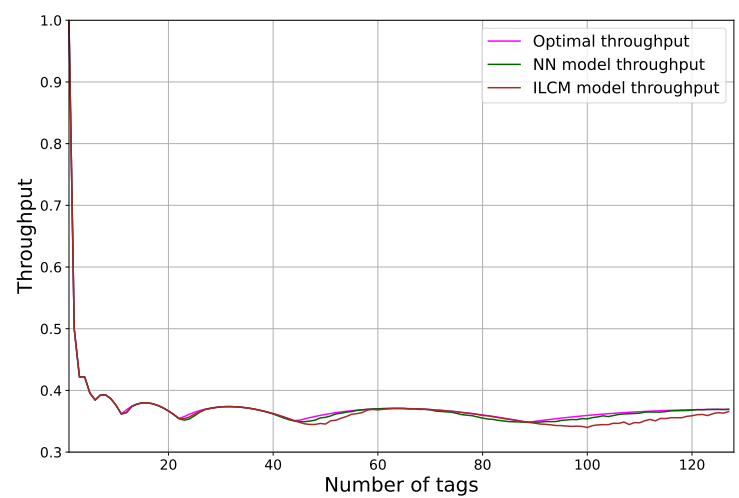


(c)

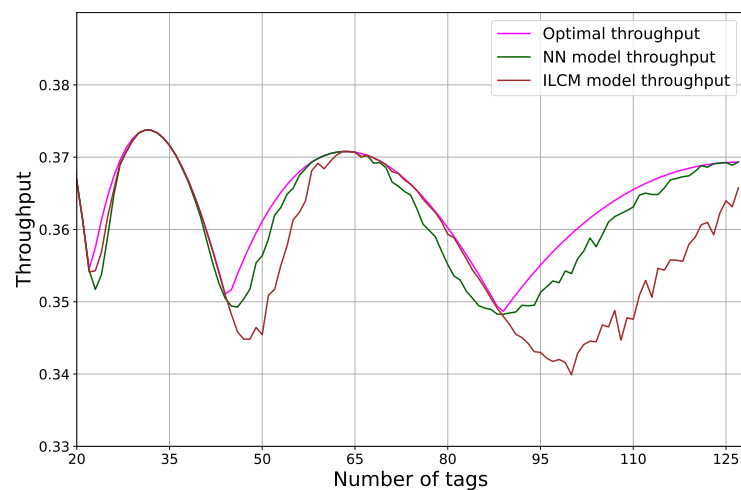
**Figure 4.** Comparison of absolute errors for Neural Network, Random Forest and ILCM model for frame sizes (a)  $L = 8$ , (b)  $L = 16$  and (c)  $L = 256$ .

The overall goal is to reach maximum throughput, and this cannot be achieved if the frame size adaptation is inadequate. The development of an effective and simple tag

estimator is burdened by the variables that must be taken into account, i.e., the frame size, the number of successful slots, and the number of collisions or empty slots. As was stated in Section 2, the major drawbacks of current estimators lie in their estimation capabilities, computational complexity and memory demands. Therefore, to achieve a better setting of the next frame size, the focus of the estimation should be on the variable that contributes the most to the overall proficiency of the system [23]. Based on the obtained results, one final measure was performed, i.e., a comparison of throughput for the NN model, ILCM and Optimal model. The Optimal being used as the benchmark is the one where the frame adaptation was set by the known number of tags. Results of the comparison are presented for the scenario of frame size  $L = 32$  realizations and are exhibited in Figure 5. As can be observed from Figure 5, the Neural Network model is close to the optimal one and outperforms the state-of-the-art ILCM model. This is particularly shown as the tag number increases, as can be seen in Figure 5b.



(a)



(b)

**Figure 5.** Comparison of throughput for the NN model, ILCM and Optimal model for the scenario of frame size  $L = 32$  realizations. (a) Throughput for the NN model, ILCM and Optimal model; (b) Throughput for the NN model, ILCM and Optimal model for a larger number of tags.

Based on the result of this examination of the performance of classifiers and comparison to the ILCM model, architectures of the Neural Network models were selected for further utilization.

## 5. Mobile RFID Reader—Implementation Feasibility

### 5.1. Current State-of-the-Art

In recent years, there has been an evident effort to enable the execution of Neural Networks on low-power and limited-performance embedded devices, such as microcontrollers (MCU) and computer boards [58]. The most common benefit that this approach can bring is not needing to transmit data (e.g., radio) to a remote location for computation. With the local implementation of ML capabilities on the MCU-like devices, everything can be executed on a device itself, thus saving the power and time that would be used for data transmission. The sensitivity of the data collected and sent to the remote device also raises concerns about expected security. With this approach, IoT devices that locally run trained ML models significantly reduce the amount of data exchanged with the server through secured or unsecured communication channels. While keeping most of the collected data on the local embedded system, some of the aforementioned privacy concerns are reduced [59]. Researchers are increasingly working on adapting existing embedded Machine Learning algorithms or applications on MCU-like devices, which were previously only possible on high-performance computers [60–62].

Led by the idea of implementing ML on embedded systems, several IT industry giants have released support for such devices. As a notable example, Google has released the TensorFlow Lite platform, which enables the user to convert TensorFlow Neural Network (NN) models, which were commonly trained for high-performance computers (e.g., Personal Computers), into a reduced model that can be stored and executed on compatible resource-constrained machines [63]. With a similar idea, Microsoft has published EdgeML, which is also designed to work on common Edge Devices [64], and even reported to work on 8-bit AVR-based Arduino (which holds only 2 KB of RAM and 32 KB of FLASH memory) on common single-board computers, such as the Raspberry Pi family. Such an example is that the same ML model (e.g., produced by TensorFlow Lite library), of course, if device resources allow it, can be executed on a microcontroller (running on Arm Cortex-M7 MCU at 600 MHz and only 51KB RAM) and computer board (e.g., Raspberry Pi4B running on quad-core Cortex-A72 1.5 GHz and 4 GB RAM). Power consumption and possible autonomy on batteries, which is a common requirement for some IoT devices, prefer microcontroller implementation (typical 3 A/5 V for RaspberryPi4 vs. 0.1 A/3.3 V typical for Arm Cortex-M7-based MCU), and for this reason, the following text is focused on ML implementation on microcontrollers with comments on implementation on computer boards. Some semiconductor manufacturers have notably supported the effort to implement ML capabilities in MCU devices. STM has released X-Cube-AI with deep learning capabilities on STM 32-bit microcontrollers [65]. An open-source library Microcontroller Software Interface Standard Neural Network (CMSIS-NN), published by ARM enables today's most popular series of Cortex-M processors' execution of ML models [66]. Another interesting example can be seen by third-party microcontroller board manufacturer OpenMV [67]. They produce an OpenMV microcontroller board (based on Arm Cortex-M7 MCU), which is a smart vision camera that is capable of executing complex machine vision algorithms for a low-cost device (typically below 80 USD). The common bottleneck of current microcontroller boards that run deep CNN is a relatively low amount of RAM, which stores NN weights and data. OpenMV H7plus board overcomes this limitation by adding another 32 MB SDRAM in addition to 1 MB RAM, which already came embedded with the microcontroller itself. Additional RAM, in turn, enables the microcontroller to store and run several complex NN models, a task that is commonly impossible on standard MCU-s and can be executed on more complex computer boards, such as the Raspberry Pi family of computer boards. Today's increasing demand for ML-enabled embedded end-devices, constant improvement in computing power and affordability would introduce new industry standards for smart cities and smart homes.

## 5.2. Experimental Setup

Common microcontroller boards, as compared to dedicated Personal Computers, are exceptionally bad with floating number calculations (in terms of execution times) as they are designed to work flawlessly with peripheral components rather than execute complex calculations. TensorFlow library can be configured to use 32-bit floating-point data types in a model for both data and weights, which generates a large model. MCU-compatible models implement an approach where integer numbers (8-bit or 16-bit) are used instead of floating-point numbers in calculations, which would considerably decrease the model size but dramatically increase execution speed. The original model (with 32-bit or 64-bit weights) can be executed on a microcontroller, of course, if the model size is small enough for the microcontroller's available RAM, but the full power of TFLiteConverter will not be used. TensorFlow Lite library for microcontrollers allows us to optimize a pre-trained Neural Network model to a selected microcontroller and implement it on the device. This ability is possible only with smart quantization, which, in turn, approximates 32-bit floating-point values into either 16-bit float-point values or 8-bit integer values. In some scenarios (such as in most complex models presented in this work), there is an evident loss in inaccuracy, which is, on the other hand, greatly compensated by a reduction in memory requirements and improvement in execution times. Finally, for some models, quantization makes all the difference if the model can or cannot be run on a memory-restricted microcontroller. TFLiteConverter, which is part of the TFLite library, can offer several optimization options; float16 quantization of weights and inputs that cut the original model size in just half, with a barely visible reduction in accuracy, dynamic range quantization where weights are 8-bit while activations are floating-point and computation is still performed in floating-point operations (optimal trade-off for some low-performance but still capable computer boards). The third optimization option showed to be ideal for low-power microcontroller devices, with forces of full integer quantization, where weight and activations are both 8-bit and all operations are integers. The aforementioned quantization is slightly more complicated than the other approaches, as the converter is required to be fed with a representative dataset before the quantization of the whole model. The data shown in Table 7 provide a simple insight into the accuracy decrease due to the performed quantization for models used in our paper.

**Table 7.** Model accuracy before and after quantization.

	Original Model	Quantized Model
Model L = 4	33.33%	32.72%
Model L = 8	28.53%	27.58 %
Model L = 16	23.11%	22.04%
Model L = 32	19.00%	12.08%
Model L = 128	8.03%	4.08%
Model L = 256	6.71%	3.03%

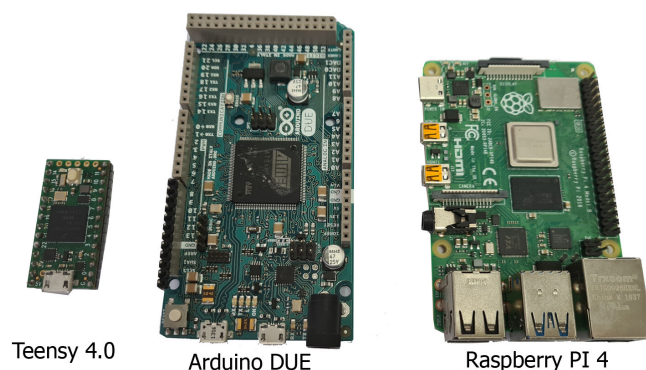
It can be observed that a decrease in accuracy is observable for the two most complex NN architectures (L = 128 and L = 256), while for the least complex NN architectures (L = 4 and L = 8) the loss due to quantization is merely measurable. The loss inaccuracy for the two most complex architectures is possibly the result of output quantization where more than 256 classes are possible (notably 512 for L = 128 and 1024 for L = 256).

After the quantized model is created, a file containing the model that the microcontroller will understand is created. The Linux command tool xxd takes a data file and outputs a text-based hex dump, which we copy-paste as a c array, and add to a microcontroller project source code (as an additional header file).

## 5.3. MCU hardware

The Teensy 4.0 microcontroller board features an ARM Cortex-M7 processor with an NXP iMXRT1062 chip clocked up to 600 MHz without additional cooling (faster stable

speeds are possible with an additional heatsink). To our knowledge, this computer board is the fastest microcontroller board available on today's market that can be used out-of-the-box [68] for complex calculations. When running at 600 MHz, Teensy 4.0 consumes approximately 100 mA current (at 3.3 V supplied voltage), considerably more than some common microcontrollers, such as the Arduino AVR family, but significantly less than any desktop computers or computer boards. The Teensy 4.0 microcontroller features 1024K RAM (of which 512K is tightly coupled) available for storing local data. The ML model is stored in FLASH memory during programming, after which it is read in whole or segment-per-segment into the RAM (this option is library dependant and tweakable). Another microcontroller board in the ML domain that was considered was AMR M3-based Arduino Due, which sports an Atmel SAM3X8E microcontroller clocked at 84 Mhz and offers a significantly smaller amount of RAM (96 KB) [69]. Considerably lower amounts of available RAM for this microcontroller can make it unusable in executing more complex ML models, where constant reading data from slower FLASH memory can lead to significantly longer execution times. The third microcontroller board that was initially considered was the popular STM32F103C8T6 (known as "blue pill"), which is also based on the Arm Cortex-M3 microcontroller. This board clocks 72 MHz but holds only 20 KB of RAM and 64 KB of FLASH memory, which makes holding and execution of most of our opposed NN models impossible. All three devices that have been tested are presented in Figure 6.



**Figure 6.** Devices used in the test: Teensy 4.0 (left), Arduino DUE (center) and Raspberry Pi4 (right). Source: Own photo.

To provide better insight into the microcontroller's performance in executing proposed NN models, the same quantized TensorFlow models were tested on a common computer board. A Raspberry Pi 4B computer board was used, which holds a quad-core Cortex-A72 1.5 GHz SOC with 4 GB of RAM, and runs Raspian desktop OS with kernel version 5.10. With simplicity in mind, all coding for the microcontroller side was performed in Arduino IDE [70], which offers a simple and intuitive interface and the availability of numerous additional libraries for a project extension. The ML model was implemented to the project by adding a hexdump file as an additional header file, which is then converted into a binary format and transferred to the microcontroller's FLASH memory during programming. The Raspberry Pi computer uses a simple Python script with an additional TFLite interpreter library.

Several proposed model architectures on the Teensy 4.0 microcontroller board were trained that have been considered as the optimal solution for executing the proposed NN models. The presented analysis aims to indicate the real limits of the NN architecture that can be fluently run on selected hardware. ANN layers' configuration was kept intact, while the complexity of the model was achieved by increasing the number of neurons in the third and fifth layers. By utilizing a microcontroller-integrated timer, the average ANN execution time has been measured on the microcontroller. Another interesting piece of information obtained was the quantized model size and amount of RAM commonly assigned for storing global variables after initial programming. Please note that microcontrollers usually do

not possess the possibility of measuring free RAM space during execution, as compared to computers. The used library offers some tweaking of tensor size, which may reduce or increase available RAM size and consequently affect execution time, but we kept this option on default for all tested models and all devices. It is recommended to keep at least 10 % of available RAM for local variables for stable performance. The results for all six models' execution times and model size on Teensy 4.0 ARM Cortex M7 microcontroller, Arduino DUE ARM Cortex M3 microcontroller and the Raspberry Pi4 computer board are listed in Table 8.

**Table 8.** Model performance on Teensy 4.0 MCU, Arduino DUE and Raspberry Pi4.

Frame Size	Model Size	Execution Time (ms)		
	(Bytes)	Teensy 4.0	Arduino DUE	Raspberry Pi4
L = 4	4320	22	897	143
L = 8	5152	32	1284	159
L = 16	6592	48	1983	173
L = 32	13,824	120	4928	187
L = 128	75,776	692	29,615	270
L = 256	283,264	1669	111,374	648

#### 5.4. Discussion

As can be observed from Table 8, increasing the number of neurons in hidden layers (notably hidden layers 3 and 5) and in output layers increases the model size and prolongs execution. As an example, comparing models for L = 4 and with the model for L = 16, which have exactly twice as many neurons as in layers 3 and 5, the total model size increases by a factor of 1.5, while the execution time on the Teensy 4.0 microcontroller observes an increase of 2.2. The last presented model (L = 256) features an increase in model size by a factor of 65 and in execution time by a factor of 75, as compared to the simplest model (L = 4). It is worth mentioning that the last model represents an example of the most complex ANN model that our microcontroller can hold, where after importing it to the microcontroller, only 13% of the RAM was free for local variables. We also observed that increasing the depth and/or increasing the number of neurons per layer of an NN poses a significant memory demand, which can be afforded only by high-end edge devices (e.g., Raspberry Pi). The average execution time for the most complex exemplary model was 1.7 ms, which is surprisingly fast for this type of device and can offer real-time performance. The ARM Cortex M3-based Arduino DUE behaved similarly to the Teensy 4.0 microcontroller with significantly longer execution times (41 times slower on the simplest model and 67 times slower on the most complex model). The execution of the most complex model took 111 ms, which makes it impractical for some real-time scenarios. Execution times on the Raspberry Pi4 computer varied greatly (due to non-real-time OS architecture) and surprisingly showed to be much slower for less complex models (up to L = 32). For more complex models, Raspberry Pi was able to benefit from its enormous computing power, and the most complex model executed in 0.6 ms, which, when compared to Teensy 4.0, is not significantly better to persuade us to use computer boards instead of the microcontroller. This once more proves that if the loss in accuracy due to quantization is acceptable, the only real limitation is available RAM and FLASH memory on the used microcontroller.

In some scenarios, RF can offer better or comparable results to deep NN with only a fraction of the execution time required on MCU [71]. As the aim of our study was to increase throughput, which is achieved by better estimating the number of interrogating tags, which is best performed by the NN model, only the NN model was considered for implementation on the microcontrollers. Additionally, NN models can offer numerous optimization and quantization possibilities, which is worth further investigation.

Based on the overall result, one final observation is made. As can be noticed in Figure 5, the  $\eta$  for ILCM and NN is quite different. Such diversity is a result of the ILCM's interpolation, even though it contributes to lower computation complexity. When

examining the worst case for both models, i.e., frame size  $L = 256$ , the Neural Network model reaches  $\eta_{NN} = 0.2498$  in contrast to  $\eta_{ILCM} = 0.2265$ . This results in a difference of 0.0233, which is approximately 6 successful slots per given frame. The reader setting determines the execution time per frame and such a time cost needs to be compared with the time for a successful tag read, i.e., successful slot time. Based on empirical evidence from research studies, such as the ones in [72], the time for standard reader setting in a general scenario is 3 ms. Therefore, the read tags that are marked as Neural Network computational burden are equal to  $1.7 \text{ ms}/3 \text{ ms} = 0.57$ .

### 5.5. Limitations to the Study

There are a few noteworthy limitations to this study. First, this study is based on data obtained from Monte Carlo simulations. Although simulated data are not the same as experimentally measured data, the Monte Carlo method creates sampling distributions of relevant statistics and can be efficiently implemented on a computer. What is more, the method allows the creation of the desired amount of sample data. The number of sample data used for training and testing ML algorithms is a crucial parameter when testing algorithms' performance. Furthermore, by using data obtained by Monte Carlo simulations, this research has remained consistent with the methodology presented in research [8] regarding the ILCM model.

Secondly, more frame sizes could have been considered to obtain better insight into models' performance and as such, they can be considered in future work. Thirdly, other Machine Learning classifiers could be employed, tested and compared to the utilized Random Forest and Neural Network.

Finally, as one of the aims of this study was to find a model that can be executed in resource-constrained microcontrollers, we were bounded by the relative "simplicity" of the proposed models (e.g., the model presented in this research is based on multilayer perceptron), while more complex NN models were not considered at this moment (e.g., recurrent networks, convolutional network), which is also planned for future work.

## 6. Conclusions

The research presented in this study aimed to explore how broadly utilized Machine Learning classifiers can be applied for tag number estimation within ALOHA-based RFID systems to increase the systems' throughput. The strengths and weaknesses of each of the models are explored on a particularly designed dataset obtained from Monte Carlo simulations. The state-of-the-art algorithm, namely the Improved Linearized Combinatorial Model (ILCM) for tag estimation, is used to compare the performance of the ML algorithms, namely Neural Network and Random Forest.

The obtained results demonstrate that the Neural Network classifier outperforms the ILCM model and achieves higher throughput.

Furthermore, this study tested to see if ML classifiers can be deployed on mobile RFID readers, aiming to maximize tag identification performance with suitable energy and processing demands. Experimental results show that the NN model architecture can be executed on resource-limited MCUs. These results imply that the conventional RFID readers may be equipped with Machine Learning classifiers that use the maximum of the available information acquired from Monte Carlo simulations. The overall results prove that the execution time on MCU is enough to meet protocol needs, keep up with the latency and improve system throughput.

**Author Contributions:** Conceptualization, L.D.R., I.S. and P.Š.; methodology, L.D.R., I.S. and P.Š.; software, L.D.R. and I.S.; validation, L.D.R., I.S., P.Š. and K.Z.; formal analysis, L.D.R. and P.Š.; investigation, L.D.R. and I.S.; resources, L.D.R., I.S. and K.Z.; data curation, L.D.R.; writing—original draft preparation, L.D.R., I.S., P.Š., K.Z. and T.P.; writing—review and editing, L.D.R., I.S., P.Š., K.Z. and T.P.; visualization, L.D.R., T.P. and I.S.; supervision, P.Š. and T.P. All authors have read and agreed to the published version of the manuscript.



**Funding:** This research was funded by by the Croatian Science Foundation under the project “Internet of Things: Research and Applications”, UIP-2017-05-4206.

**Conflicts of Interest:** The authors declare no conflict of interest.

### Abbreviations

The following abbreviations are used in this manuscript:

IoT	Internet of Things
RFID	Radio Frequency Identification
WIPT	Wireless Information and Power Transfer
DFSA	Dynamic Framed Slotted ALOHA
TDMA	Time-Division multiple-access
ILCM	Improved Linearized Combinatorial model
ML	Machine Learning
DT	Decision Tree
k-NN	k-Nearest Neighbour
SVM	Support Vector Machine
RF	Random Forest
DL	Deep Learning
ANN	Artificial Neural Networks
NN	Neural Network
ReLU	Rectified Linear Unit
SGD	Stochastic Gradient Descent
Adam	Adaptive Moment Optimization
RMSProp	Root Mean Square Propagation
MAE	Mean Absolute Errors
CMSIS-NN	Microcontroller Software Interface Standard Neural Network
MCU	microcontrollers

### References

1. Cui, L.; Zhang, Z.; Gao, N.; Meng, Z.; Li, Z. Radio Frequency Identification and Sensing Techniques and Their Applications—A Review of the State-of-the-Art. *Sensors* **2019**, *19*, 4012. [[CrossRef](#)] [[PubMed](#)]
2. Santos, Y.; Canedo, E. On the Design and Implementation of an IoT based Architecture for Reading Ultra High Frequency Tags. *Information* **2019**, *10*, 41. [[CrossRef](#)]
3. Landaluce, H.; Arjona, L.; Perallos, A.; Falcone, F.; Angulo, I.; Muralter, F. A Review of IoT Sensing Applications and Challenges Using RFID and Wireless Sensor Networks. *Sensors* **2020**, *20*, 2495. [[CrossRef](#)]
4. Dobkin, D.D. *The RF in RFID*; Elsevier: Burlington, MA, USA, 2008.
5. Škiljo, M.; Šolić, P.; Blažević, Z.; Patrono, L.; Rodrigues, J.J.P.C. Electromagnetic characterization of SNR variation in passive Gen2 RFID system. In Proceedings of the 2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN), Milan, Italy, 4–7 July 2017; pp. 172–175. [[CrossRef](#)]
6. Šolić, P.; Maras, J.; Radić, J.; Blažević, Z. Comparing Theoretical and Experimental Results in Gen2 RFID Throughput. *IEEE Trans. Autom. Sci. Eng.* **2017**, *14*, 349–357. [[CrossRef](#)]
7. EPCglobal Inc. *Class1 Generation 2 UHF Air Interface Protocol Standard*; Technical Report; EPCglobal Inc.: Wellington, New Zealand, 2015.
8. Šolić, P.; Radić, J.; Rožić, N. Energy Efficient Tag Estimation Method for ALOHA-Based RFID Systems. *IEEE Sens. J.* **2014**, *14*, 3637–3647. [[CrossRef](#)]
9. Law, C.; Lee, K.; Siu, K.Y. *Efficient Memoryless Protocol for Tag Identification (Extended Abstract)*; Association for Computing Machinery: New York, NY, USA, 2000. [[CrossRef](#)]
10. Capetanakis, J. Tree algorithms for packet broadcast channels. *IEEE Trans. Inf. Theory* **1979**, *25*, 505–515. [[CrossRef](#)]
11. Rodić, L.D.; Stančić, I.; Zovko, K.; Šolić, P. Machine Learning as Tag Estimation Method for ALOHA-based RFID system. In Proceedings of the 2021 6th International Conference on Smart and Sustainable Technologies (SpliTech), Bol and Split, Croatia, 8–11 September 2021; pp. 1–6. [[CrossRef](#)]
12. Schoute, F. Dynamic Frame Length ALOHA. *IEEE Trans. Commun.* **1983**, *31*, 565–568. [[CrossRef](#)]
13. Škiljo, M.; Šolić, P.; Blažević, Z.; Rodić, L.D.; Perković, T. UHF RFID: Retail Store Performance. *IEEE J. Radio Freq. Identif.* **2021**, *6*, 481–489. [[CrossRef](#)]
14. Chen, W. An Accurate Tag Estimate Method for Improving the Performance of an RFID Anticollision Algorithm Based on Dynamic Frame Length ALOHA. *IEEE Trans. Autom. Sci. Eng.* **2009**, *6*, 9–15. [[CrossRef](#)]

15. Vogt, H. Efficient object identification with passive RFID tags. In Proceedings of the International Conference on Pervasive Computing, Zürich, Switzerland, 26–28 August 2002; pp. 98–113.
16. Šolić, P.; Radić, J.; Rozić, N. Linearized Combinatorial Model for optimal frame selection in Gen2 RFID system. In Proceedings of the 2012 IEEE International Conference on RFID (RFID), Orlando, FL, USA, 3–5 April 2012; pp. 89–94. [\[CrossRef\]](#)
17. Vahedi, E.; Wong, V.W.S.; Blake, I.F.; Ward, R.K. Probabilistic Analysis and Correction of Chen’s Tag Estimate Method. *IEEE Trans. Autom. Sci. Eng.* **2011**, *8*, 659–663. [\[CrossRef\]](#)
18. Arjona, L.; Landaluce, H.; Perallos, A.; Onieva, E. Scalable RFID Tag Estimator With Enhanced Accuracy and Low Estimation Time. *IEEE Signal Process. Lett.* **2017**, *24*, 982–986. [\[CrossRef\]](#)
19. Delgado, M.; Vales-Alonso, J.; Gonzalez-Castao, F. Analysis of DFSA anti-collision protocols in passive RFID environments. In Proceedings of the 2009 35th Annual Conference of IEEE Industrial Electronics, Porto, Portugal, 3–5 November 2009; pp. 2610–2617. [\[CrossRef\]](#)
20. Vales-Alonso, J.; Bueno-Delgado, V.; Egea-Lopez, E.; Gonzalez-Castano, F.J.; Alcaraz, J. Multiframe Maximum-Likelihood Tag Estimation for RFID Anticollision Protocols. *IEEE Trans. Ind. Inform.* **2011**, *7*, 487–496. [\[CrossRef\]](#)
21. Wang, S.; Aggarwal, C.; Liu, H. Using a Random Forest to Inspire a Neural Network and Improving on It. In Proceedings of the 2017 SIAM International Conference on Data Mining (SDM), Houston, TX, USA, 27–29 April 2017; pp. 1–9. [\[CrossRef\]](#)
22. Filho, I.E.D.B.; Silva, I.; Viegas, C.M.D. An Effective Extension of Anti-Collision Protocol for RFID in the Industrial Internet of Things (IIoT). *Sensors* **2018**, *18*, 4426. [\[CrossRef\]](#) [\[PubMed\]](#)
23. Radić, J.; Šolić, P.; Škiljo, M. Anticollision algorithm for radio frequency identification system with low memory requirements: NA. *Trans. Emerg. Telecommun. Technol.* **2020**, *31*, e3969. [\[CrossRef\]](#)
24. Boovaraghavan, S.; Maravi, A.; Mallela, P.; Agarwal, Y. MLIoT: An End-to-End Machine Learning System for the Internet-of-Things. In Proceedings of the International Conference on Internet-of-Things Design and Implementation, Charlottesville, VA, USA, 18–21 May 2021; IoTDI ’21, p. 169–181. [\[CrossRef\]](#)
25. Mitchell, T.M. *Machine Learning*; McGraw-Hill: New York, NY, USA, 1997.
26. Sarkar, D.; Bali, R.; Sharma, T. *Practical Machine Learning with Python: A Problem-Solver’s Guide to Building Real-World Intelligent Systems*, 1st ed.; Apress: New York, NY, USA, 2017.
27. Zantalis, F.; Koulouras, G.; Karabetos, S.; Kandris, D. A Review of Machine Learning and IoT in Smart Transportation. *Future Internet* **2019**, *11*, 94. [\[CrossRef\]](#)
28. Hussain, F.; Hussain, R.; Hassan, S.A.; Hossain, E. Machine Learning in IoT Security: Current Solutions and Future Challenges. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 1686–1721. [\[CrossRef\]](#)
29. Dujčić Rodić, L.; Perković, T.; Županović, T.; Šolić, P. Sensing Occupancy through Software: Smart Parking Proof of Concept. *Electronics* **2020**, *9*, 2207. [\[CrossRef\]](#)
30. Sen, P.C.; Hajra, M.; Ghosh, M. Supervised Classification Algorithms in Machine Learning: A Survey and Review. In *Emerging Technology in Modelling and Graphics*; Mandal, J.K., Bhattacharya, D., Eds.; Springer: Singapore, 2020; pp. 99–111.
31. Soofi, A.A.; Awan, A. Classification Techniques in Machine Learning: Applications and Issues. *J. Basic Appl. Sci.* **2017**, *13*, 459–465. [\[CrossRef\]](#)
32. Zhu, X.X.; Tuia, D.; Mou, L.; Xia, G.; Zhang, L.; Xu, F.; Fraundorfer, F. Deep Learning in Remote Sensing: A Comprehensive Review and List of Resources. *IEEE Geosci. Remote. Sens. Mag.* **2017**, *5*, 8–36. [\[CrossRef\]](#)
33. LeCun, Y.; Bengio, Y.; Hinton, G. Deep Learning. *Nature* **2015**, *521*, 436–444. [\[CrossRef\]](#)
34. Kotsiantis, S.B.; Zaharakis, I.; Pintelas, P. Supervised machine learning: A review of classification techniques. *Emerg. Artif. Intell. Appl. Comput. Eng.* **2007**, *160*, 3–24.
35. Nikam, S.S. A comparative study of classification techniques in data mining algorithms. *Orient. J. Comput. Sci. Technol.* **2015**, *8*, 13–19.
36. Charoenpong, J.; Pimpunchat, B.; Amornsamankul, S.; Triampo, W.; Nuttavut, N. A Comparison of Machine Learning Algorithms and their Applications. *Int. J. Simul. Syst. Sci. Technol.* **2019**, *20*, 8.
37. Mothkur, R.; Poornima, K. Machine learning will transfigure medical sector: A survey. In Proceedings of the 2018 International Conference on Current Trends towards Converging Technologies (ICCTCT), Coimbatore, India, 1–3 March 2018; pp. 1–8.
38. Cutler, A.; Cutler, D.; Stevens, J. Random Forests. In *Ensemble Machine Learning*; Springer: Boston, MA, USA, 2012; Volume 45, pp. 157–176. [\[CrossRef\]](#)
39. Paul, A.; Mukherjee, D.P.; Das, P.; Gangopadhyay, A.; Chintla, A.R.; Kundu, S. Improved Random Forest for Classification. *IEEE Trans. Image Process.* **2018**, *27*, 4012–4024. [\[CrossRef\]](#)
40. Kulkarni, V.Y.; Sinha, P.K. Pruning of Random Forest classifiers: A survey and future directions. In Proceedings of the 2012 International Conference on Data Science Engineering (ICDSE), Kerala, India, 18–20 July 2012; pp. 64–68. [\[CrossRef\]](#)
41. Lipton, Z.C. A Critical Review of Recurrent Neural Networks for Sequence Learning. *arXiv* **2015**, arXiv:1506.00019.
42. Ng, A.Y.; Jordan, M.I. On Discriminative vs. Generative Classifiers: A comparison of logistic regression and naive Bayes. In Proceedings of the Advances in Neural Information Processing Systems 14 (NIPS 2001), Vancouver, BC, Canada, 3–8 December 2001; Dietterich, T.G., Becker, S., Ghahramani, Z., Eds.; MIT Press: Cambridge, MA, USA, 2001; pp. 841–848.
43. Joo, R.; Bertrand, S.; Tam, J.; Fablet, R. Hidden Markov Models: The Best Models for Forager Movements? *PLoS ONE* **2013**, *8*, e71246. [\[CrossRef\]](#)

44. Zhang, G.P. Neural networks for classification: A survey. *IEEE Trans. Syst. Man, Cybern. Part C Appl. Rev.* **2000**, *30*, 451–462. [[CrossRef](#)]
45. Roßbach, P. *Neural Networks vs. Random Forests—Does It Always Have to Be Deep Learning*; Frankfurt School of Finance and Management: Frankfurt am Main, Germany, 2018.
46. Biau, G.; Scornet, E.; Welbl, J. Neural random forests. *Sankhya A* **2019**, *81*, 347–386. [[CrossRef](#)]
47. NVIDIA; Vingelmann, P.; Fitzek, F.H. *CUDA, Release: 11.2*; 2021. Available online: <https://developer.nvidia.com/cuda-toolkit> (accessed on 21 July 2021).
48. Provoost, J.; Wismans, L.; der Drift, S.V.; Kamilaris, A.; Keulen, M.V. Short Term Prediction of Parking Area states Using Real Time Data and Machine Learning Techniques. *arXiv* **2019**, arXiv:1911.13178.
49. Dongare, A.D.; Kharde, R.R.; Kachare, A.D. Introduction to Artificial Neural Network. *Int. J. Eng. Innov. Technol.* **2012**, *2*, 189–194.
50. Hayou, S.; Doucet, A.; Rousseau, J. On the Impact of the Activation function on Deep Neural Networks Training. In Proceedings of the 36th International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019; Chaudhuri, K., Salakhutdinov, R., Eds.; 2019; Volume 97, pp. 2672–2680.
51. Breiman, L. Random Forests. *Mach. Learn.* **2001**, *45*, 5–32. [[CrossRef](#)]
52. Probst, P.; Wright, M.N.; Boulesteix, A.L. Hyperparameters and tuning strategies for random forest. *Wires Data Min. Knowl. Discov.* **2019**, *9*, e1301. [[CrossRef](#)]
53. Biau, G.; Scornet, E. A random forest guided tour. *Test* **2016**, *25*, 197–227. [[CrossRef](#)]
54. Prinzie, A.; Van den Poel, D. Random Forests for multiclass classification: Random MultiNomial Logit. *Expert Syst. Appl.* **2008**, *34*, 1721–1732. [[CrossRef](#)]
55. Oshiro, T.M.; Perez, P.S.; Baranauskas, J.A. How Many Trees in a Random Forest? In Proceedings of the Machine Learning and Data Mining in Pattern Recognition, Berlin, Germany, 20–21 July 2012; Perner, P., Ed.; Springer: Berlin/Heidelberg, Germany, 2012; pp. 154–168.
56. Belgiu, M.; Drăguț, L. Random forest in remote sensing: A review of applications and future directions. *ISPRS J. Photogramm. Remote. Sens.* **2016**, *114*, 24–31. [[CrossRef](#)]
57. Su, J.; Sheng, Z.; Leung, V.C.M.; Chen, Y. Energy Efficient Tag Identification Algorithms For RFID: Survey, Motivation And New Design. *IEEE Wirel. Commun.* **2019**, *26*, 118–124. [[CrossRef](#)]
58. Sakr, F.; Bellotti, F.; Berta, R.; De Gloria, A. Machine Learning on Mainstream Microcontrollers. *Sensors* **2020**, *20*, 2638. [[CrossRef](#)]
59. Shi, W.; Cao, J.; Zhang, Q.; Li, Y.; Xu, L. Edge Computing: Vision and Challenges. *IEEE Internet Things J.* **2016**, *3*, 637–646. [[CrossRef](#)]
60. Magno, M.; Cavigelli, L.; Mayer, P.; von Hagen, F.; Benini, L. FANNCortexM: An Open Source Toolkit for Deployment of Multi-layer Neural Networks on ARM Cortex-M Family Microcontrollers: Performance Analysis with Stress Detection. In Proceedings of the 2019 IEEE 5th World Forum on Internet of Things (WF-IoT), Limerick, Ireland, 15–18 April 2019; pp. 793–798. [[CrossRef](#)]
61. Alameh, M.; Abbass, Y.; Ibrahim, A.; Valle, M. Smart Tactile Sensing Systems Based on Embedded CNN Implementations. *Micromachines* **2020**, *11*, 103. [[CrossRef](#)]
62. Sharma, R.; Biookaghazadeh, S.; Li, B.; Zhao, M. Are Existing Knowledge Transfer Techniques Effective for Deep Learning with Edge Devices? In Proceedings of the 2018 IEEE International Conference on Edge Computing (EDGE), San Francisco, CA, USA, 2–7 July 2018; pp. 42–49. [[CrossRef](#)]
63. TensorFlow Lite. Available online: <http://www.tensorflow.org/lite> (accessed on 21 July 2021).
64. EdgeML Machine LEARNING for Resource-Constrained Edge Devices. Available online: <https://github.com/Microsoft/EdgeML> (accessed on 21 July 2021).
65. STM32CubeMX—STMicroelectronics, X-CUBE-AI—AI. Available online: <https://http://www.st.com/en/embedded-software/x-cube-ai.html> (accessed on 21 July 2021).
66. CMSIS NN Software Library. Available online: [https://arm-software.github.io/CMSIS\\_5/NN/html/index.html](https://arm-software.github.io/CMSIS_5/NN/html/index.html) (accessed on 21 July 2021).
67. COepnMV. Available online: <https://openmv.io> (accessed on 21 July 2021).
68. Teensy 4.0 Development Board. Available online: <https://www.pjrc.com/store/teensy40.html> (accessed on 21 July 2021).
69. Atmel SAM3X8E ARM Cortex-M3 MCU. Available online: [http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-1105-7-32-bit-Cortex-M3-Microcontroller-SAM3X-SAM3A\\_Datasheet.pdf](http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-1105-7-32-bit-Cortex-M3-Microcontroller-SAM3X-SAM3A_Datasheet.pdf) (accessed on 21 July 2021).
70. Arduino IDE. Available online: <https://www.arduino.cc/en/software> (accessed on 21 July 2021).
71. Elsts, A.; McConville, R. Are Microcontrollers Ready for Deep Learning-Based Human Activity Recognition? *Electronics* **2021**, *10*, 2640. [[CrossRef](#)]
72. Šolić, P.; Šarić, M.; Stella, M. RFID reader-tag communication throughput analysis using Gen2 Q-algorithm frame adaptation scheme. *Int. J. Circuits Syst. Signal Proc.* **2014**, *8*, 233–239.

## **APPENDIX D**

**Paper Title:**

Tangible interfaces in early years' education: a systematic review

**Abstract:**

This paper presents a systematic review of the literature on Tangible User Interfaces (TUIs) and interactions in young children's education by identifying 155 studies published between 2001 and 2019. The review was based on a set of clear research questions addressing application domains, forms of tangible objects, TUI design and assessment. The results indicate that (i) the form of tangible object is closely related to the application domain, (ii) the manipulatives are the most dominant form of tangible object, (iii) the majority of studies addressed all three stages of TUI development (design, implementation and evaluation) and declared a small sample of young children as a major shortcoming, and (iv) additional empirical research is required to collect evidence that TUIs are truly beneficial for children's acquisition of knowledge. This review also identifies gaps in the current work, thus providing suggestions for future research in TUIs application in educational context expected to be beneficial for researchers, curriculum designers and practitioners in early years' education. To the authors' knowledge, this is the first systematic review specific to TUIs' studies in early years' education and is an asset to the scientific community.

**Cite as:**

Rodić, L.D., Granić, A. Tangible interfaces in early years' education: a systematic review. *Pers Ubiquit Comput* 26, 39–77 (2022).

**DOI:**

<https://doi.org/10.1007/s00779-021-01556-x>

## **APPENDIX E**

Article

# Sensing Occupancy through Software: Smart Parking Proof of Concept

Lea Dujić Rodić \*, Toni Perković \*, Tomislav Županović and Petar Šolić \*

Faculty of Electrical Engineering, Mechanical Engineering and Naval Architecture in Split (FESB),  
University of Split, 21000 Split, Croatia; tzupan01@fesb.hr

\* Correspondence: dujic@fesb.hr (L.D.R.); toperkovic@fesb.hr (T.P.); psolic@fesb.hr (P.Š.)

Received: 29 November 2020; Accepted: 17 December 2020; Published: 21 December 2020



**Abstract:** In order to detect the vehicle presence in parking slots, different approaches have been utilized, which range from image recognition to sensing via detection nodes. The last one is usually based on getting the presence data from one or more sensors (commonly magnetic or IR-based), controlled and processed by a micro-controller that sends the data through radio interface. Consequently, given nodes have multiple components, adequate software is required for its control and state-machine to communicate its status to the receiver. This paper presents an alternative, cost-effective beacon-based mechanism for sensing the vehicle presence. It is based on the well-known effect that, once the metallic obstacle (i.e., vehicle) is on top of the sensing node, the signal strength will be attenuated, while the same shall be recognized at the receiver side. Therefore, the signal strength change conveys the information regarding the presence. Algorithms processing signal strength change at the receiver side to estimate the presence are required due to the stochastic nature of signal strength parameters. In order to prove the concept, experimental setup based on LoRa-based parking sensors was used to gather occupancy/signal strength data. In order to extract the information of presence, the Hidden Markov Model (HMM) was employed with accuracy of up to 96%, while the Neural Network (NN) approach reaches an accuracy of up to 97%. The given approach reduces the costs of the sensor production by at least 50%.

**Keywords:** parking occupancy; RSSI; SNR; LoRa; Hidden Markov Model; Deep Learning; Neural Networks

## 1. Introduction

Intense technological development currently is reshaping many areas of everyday life and impacting human behavior. The Internet of Things (IoT) vision of ubiquitous and pervasive connection of smart things gives rise to a future environment that is composed out of physical and digital world. In this environment, it is possible to receive information about or from the physical world that was previously not available to us and, moreover, interconnect it to exchange and use this information with the digital world [1]. The IoT applications are being employed in diverse areas of industry, communication, wireless sensor networks, data mining, assisted living, etc., giving rise to the concept of Smart City.

The Smart City is constituted out of gathered and processed information, covering a wide range of entities, such as transportation, health, food, and education for the overall improvement of life quality [2]. One of the most important topics addressed by the European Commission and most nations in the world is the development of an urban city model that aimed at increasing the quality of life of people working and living in them. Smart and Sustainable Mobility is one of the central concepts in the vision of the Smart City, where IoT plays an important role [3,4]. In urban city areas, due to the rise of cars, existing parking systems are inadequate or unable to handle parking loads [5]. Moreover, parking facilities are not accessible in an adequate manner, since it is estimated that drivers

spend around 7.8 min. in finding free parking lots [6]. Studies have shown that, in traffic dense environments in urban areas, 30–50% of drivers are in search of free parking [7]. In addition, the IBM survey (IBM Survey. Available online: <https://www-03.ibm.com/press/us/en/pressrelease/35515.wss> (accessed on 20 November 2020)) reported that due to traffic in metropolitan cities such as Beijing or Madrid drivers spend, on average, 30 to 40 min. searching a free parking space. The increase of fuel consumption and air pollution is one of the major issues that arise from this [8]. Furthermore, the consequences of traffic jams are the frustration of drivers and higher probability of accidents [7]. Finally, traffic congestion leads to cost-effective losses, since, in a city of 50,000 inhabitants, having, on average, 250 parking lots, generates an annual cost of 216,000 US dollars [9].

In the last decade, the development of dynamic and complex IoT system and number of connected devices keeps increasing exponentially as well as the data that were collected by these devices that need to be properly analyzed [10]. Effective analysis of big data can extract meaningful information and correlation amongst vast quantities of data that are generated by sensor devices, which are a key factor in the success in many domains and, especially, in the Smart City applications [11]. Therefore, IoT devices need to be able to manage data collection, Machine-to-Machine (M2M) communication, pre-processing of the data if needed, whilst compensating among cost, processing power, and energy consumption [12].

Along with great advancements in technology, including the availability of cheap and massive computing, hardware and storage arose Machine Learning (ML) holding a vast potential for data analysis, and precise predictions made from the past observations for given new measurements [13]. Machine learning is the most prominent artificial intelligence (AI) algorithm, which has been utilized in various fields from computer vision, computer graphics, natural language processing (NLP) to speech recognition, decision-making, and intelligent control [14], as well as in intrusion detection systems [15]. Within IoT devices, the application of ML can enable users to gain deeper insight into data correlations and mine the information and features that are hidden within this data [16]. With that regard, IoT applications that use sensor technology, RFID technology, network communication, data mining and machine learning could prove to be quite efficient in solving the previously presented problem of free parking space [17].

Deep Learning methods, such as deep long short term memory network (LSTM), have been recently applied for the prediction of available free parking space [18]. A recent review of literature presented in [19] pointed to several open issues and challenges with regards to the design of Smart Parking spaces, emphasizing the utilization of car parking with emerging technologies, such as Deep Learning. A commonly and widely used model for sequential or time series data in ML and statistics is the Hidden Markov Model (HMM) [20]. HMMs are based on the concept of Markov Chain, and they can represent any random sequential process that undergoes transitions from one state to another [21]. In the last two decades, HMM has been used in various areas as a data-driven modeling approach in automatic speech recognition, pattern recognition, signal processing, telecommunication, bioinformatics, etc. [22]. Recent works of researches incorporated Markov models for parking space occupancy predictions. For instance, in [23], the authors propose a model-based framework in order to predict future occupancy from historical occupancy data. The foundation of this predictive framework is continuous-time Markov queuing model, which is employed to describe the stochastic occupancy change of a parking facility. The model was evaluated while using a mean absolute relative error (MARE), ranging from 5.23% to 1.86% for different case studies. Furthermore, in [24], an agent-based service combined with a learning and prediction system that uses a time varying Markov chain to predict parking availability is proposed. Agents predict the parking availability in a given parking garage and communicate with other agents in order to produce a cumulative prediction achieving prediction accuracy of about 83%.

In recent years, the field of Deep Learning has become rather prominent, and the concept of artificial Neural Networks (NN), which are inspired by brain nervous system, has gained significant interest amongst researches [25]. Neural Networks have the capacity to learn hierarchical



representations and are well suited for machine perception tasks, where the crude underlying features cannot be individually interpreted [26]. This makes them a powerful ML tool that achieves state-of-the-art results in a wide range of supervised and unsupervised machine learning tasks. Neural Networks have been efficiently implemented in a variety of fields like pattern recognition, signal processing and control of complex nonlinear systems [25]. Moreover, NNs have also been applied in prediction of future occupancy status such as in [27,28]. Using the data regarding the duration of free parking space and occupancy status, the researchers in [27] have developed a short-term and long term parking availability prediction system based on Neural Network. They have concluded that NNs can adequately capture the temporal transformations of parking status providing accurate prediction of occupancy up to half an hour ahead. More recently, in [28], the authors use a Deep Learning Neural Network for parking lot occupancy status classification that is based on images of parking spaces, giving 93% correct classification rate for a particular data set.

Existing Smart Parking solutions for detecting occupancy include the usage of adequate sensing technologies and transmission to a centralized system for further processing (using appropriate radio technology, such as LoRa, NB-IoT, Sigfox, BLE5, etc.). Such devices use detection techniques that are based on sensors, such as light, magnetometer, infrared detector, distance sensors, or a combination of sensing technologies [29–32]. Moreover, the researchers in [33] point out that the employment of a purposeful Smart Parking solution must take into account people with special needs and enable parking for disabled. Therefore, they have utilized RFID and database authentication for the use of ultrasonic sensors, LED, and cloud technology method for better and improved disabled parking management. However, these solutions are rather power hungry, due to the consumption of a large number of sensors, microcontrollers (MCUs), and radio communication peripherals, which impact the lifetime of an otherwise battery-powered device. Consequently, the existence of sensing technologies in Smart Parking sensor devices often requires from manufacturer the implementation of circuitry that requires, from MCU, a state-machine capable methodology, adequate software for sensor activation and sensor readings, and decision making about, and radio communication upon, parking status changes. In addition, such devices are usually implemented with the capability to receive communication over the radio from centralized systems/gateways for making updates (e.g., duty cycle period, time synchronization), but also perform online firmware updates. Taking into account additional requirements from the end user to calibrate sensors prior installing them, there is a need for an alternative solution that would be easier to implement. The research that is presented in this paper proposes a hardware sensing solution through software that uses signal strength information to achieve cost savings. A novel software approach would employ appropriate ML algorithms to gain a high level of occupancy status detection, thus achieving cost saving by reducing the price of the sensing device. The idea of such a solution has found its basis in some recent research that observed a scenario in which the signal strength at the receiver side is significantly reduced [6,34] when a vehicle occupies a parking lot. Emerging techniques, like Machine Learning and intelligent sensing in car parks, might be able to efficiently reduce the parking search time and improve mobility [19]. Similarly, it has been observed that the measurement of the received signal strength from the LoRa radio module could serve as a humidity indicator for the purpose of soil moisture detection [35,36]. In the parking environment, when something changes, such as when the car goes over the sensor device covering the parking lot, signal strength at the receiver side will change. This indicates that the signal strength change also holds information about the vehicle presence. Using the above principle in which the vehicle presence can cause a drop in signal power at the receiver side from a LoRa-based device, this paper introduces a novel system for the cost-effective and low-power detection of parking slot occupancy. Because signal strength change and parking occupancy present highly correlated processes, it is reasonable to use machine learning techniques, such as Hidden Markov Model and Neural Networks, for detecting/estimating occupancy from signal strength change with a low error rate. This way, the hardware problem of sensing occupancy is solved through software while using HMM and NN, where a high estimation detection result is achieved, which, at the end, will result in

reducing the overall price of the sensing device. Consequently, the device will become a simple beacon device (without any sensor), where occupancy is detected with a significant change in signal strength. Moreover, the proposed solution could also serve as additional sensor to already existing parking lot detection techniques in order to improve parking lot occupancy monitoring, without any hardware changes to existing sensing techniques. Using techniques that are based on the Hidden Markov Model, it is possible to estimate parking space occupancy based on signal strength with an accuracy of up to 96%. When introducing machine learning techniques that are based on Neural Networks, parking lot occupancy can be correctly estimated with an accuracy of up to 97%.

## 2. State of the Art

Smart Parking solutions vary with regards to sensing technologies and methods that are used for parking space occupancy prediction and classification. When regarding the architecture of these solution it can be noticed that it is generally constituted out of three distinguishing components: type of sensors, network protocols, and software solutions [37]. In [38], the authors designed a prototype of a parking occupancy monitoring and visualization system that uses an ultrasonic sensor being controlled by an Arduino Uno which uses a Wireless XBee shield and an XBee Series 2 module for communication. The data collected from the sensor is then given as an input to a algorithm that detects parking space statues and reports to a database in a real-time basis. Moreover, in [39], the authors presented a novel system for detecting the cruising behavior in vehicle journeys and developed a real-time parking information system. The system uses GPS sensors as an application that sends the user's location and allows for the system to create a heat map with the acquired information showing free and unavailable parking lots. The proposed method relies on the principle of detecting a significant local minimum in the GPS trace with respect to the distance from the destination. In addition to GPS data, other sensing data from the driver's smartphone, such as accelerometer, gyroscope, and magnetometer, were also collected. Classification using Decision Trees (DT), Support Vector Machines (SVM) and  $k$ -Nearest Neighbors ( $k$ -NN) is used to detect cruising behavior. The system then automatically annotates parking availability on road segments based on the classified data and displays this information as a heat-map of parking availability information on the user's smartphone. Using this approach, the researches were able to detect cruising on average 81% of the time. In [40], the authors used a light detection and ranging optical sensor (LIDAR) in order to measure the distance between a car and an object next to it. They have combined this sensor with a GPS receiver to determine the speed of a vehicle in a particular pair of geographic coordinates and a web camera to track tests. The information were then sent to a Raspberry Pi connected to the cloud via LTE-IEEE 802.11p protocol for further data processing and analyses. Parking situations were estimated by applying machine learning. Research that was conducted in [41] uses video camera sensors for detecting multiple parking space occupancy. Using image processing techniques: the Histogram of oriented Gradient (HOG) descriptor, the Scale-invariant feature transform (SIFT) corner detector, and Metrics on Color Spaces YUV, HSV, and YCrCb authors achieved an accuracy rate of over 93% for parking lot occupancy detection.

In the last decade, a number of solutions aiming at predicting the occupancy in the future have emerged with the goal of simplifying the search of free parking spaces. These solutions are based on Machine Learning techniques that involve learning, predicting, and the exploiting of cloud based architectures for data storage [42]. Generally, data regarding occupancy are the history of occupancy for a parking lot, containing date-time information with a specific occupancy status. For instance, in the work [43], while using ML, the authors present two smart car parking scenarios based on real-time car parking information that has been collected from sensors in the City of San Francisco, USA, and the City of Melbourne, Australia. The historic data contained features, like area name, street name, side of street, street marker, arrival time, departure time, duration of parking events (in seconds), sign, in violation, street ID, and device ID. From these data, the occupancy rate was calculated. The evaluation revealed that the Regression Tree, when compared to NN and SVR, using a feature set that includes the history of the occupancy rates along with the time and the day of the week

performed best for prediction of a free parking space on both the data sets. Moreover, in research [42], the authors applied a Recurrent Neural Network (RNN)-based approach for the prediction of the number of free parking spaces. They have used parking data of Birmingham, U.K., which contained the parking occupancy rate for each parking area given the time and date. They achieved the median of mean absolute error of 0.077 for prediction of occupancy. The results show that the approach used is accurate to the point of being useful for being utilized in Smart Parking solutions. In [44], the authors discuss the problem of predicting the number of available parking spaces in a parking lot by regarding the vehicle's arrival as a Poisson distribution process. They model the parking lot as a continuous-time Markov chain. With the predicted occupancy status, each parking lot can provide availability information to the drivers via vehicular networks. The work presented in [45] investigates the changing characteristics of short-term available parking spaces. The availability data were collected from parking in several off-street parking garages in Newcastle. This forecasting model is based on the Wavelet Neural Network (WNN) method and it is compared with the largest Lyapunov exponents (LEs) method in the aspects of accuracy, efficiency, and robustness. They conclude that WNN gives a more accurate short-term forecasting prediction with a average mean square error (MSE) is  $6.4 \pm 3.1$ . More recently, the authors in [18] presented a framework that is based on LSTM in order to predict the availability of parking space with the integration of Internet of Things (IoT). They have also used the previously mentioned Birmingham parking sensors data set for performance evaluation of free parking space prediction that is based on location, days of a week, and working hours of a day. The authors show that, from all performance measurement parameters, the minimum prediction accuracy is 93.2% (RMSE) and maximum prediction accuracy is 99.8% (MSLE). They present the experimental results that show that their proposed model outperforms the state-of-the-art prediction models. Finally, they point to some limitations of the study regarding the decision support system: it predicts the availability of parking lots only considering the parking occupancy information.

Table 1 gives a short comparison of identified researches regarding the technological architecture of these existing Smart Parking solutions and the concept that is presented in this paper.

**Table 1.** Comparison table of various sensing technologies and it applications in Smart Parking.

Paper	Sensing Device (Network Protocol)	Data Type	Application	ML Model	Detection Rate
J. Xiao et al. [23]	/	occupancy history	parking traffic	Markov M/M/C/C model	MARE 1.486%
Tilahun and Di Marzo Serugendo [24]	/	occupancy history	parking availability prediction	agent-based service (Markov chain)	83% prediction accuracy
Vlahogianni et al. [27]	ferromagnetic parking sensor (802.15.4 protocol)	occupancy history	parking occupancy prediction	NN	0.004 MAE
Farag et al. [28]	camera	parking spaces images	parking occupancy classification	NN	93% classification rate
Grodi et al. [38]	ultrasonic sensor (XBee Series)	occupancy history	parking occupancy detection	none	none
Jones et al. [39]	GPS sensors	location data	detection of cruising behaviour	DT, SVM, $k$ -NN	81% detection accuracy
Hiesmair et al. [40]	LIDAR(LTE-IEEE 802.11p), GPS	distance, speed	estimation of parking situation	NN, DT, $k$ -NN, SVM	95% accuracy
Tatulea et al. [41]	video camera sensor	images	parking occupancy	none	93% accuracy
Camero et al. [42]	/	parking occupancy rate history	prediction of occupancy rate	RNN	MAE of 0.077
Zheng et al. [43]	/	history of observations	parking occupancy rate	Regression Tree, NN, SVR	Mean MAE 0.019–0.079
Ji et al. [45]	/	availability data over time	prediction of occupancy	WNN	MSE $6.4 \pm 3.1$
Ali et al. [18]	/(LoRaWAN)	occupancy history	prediction of occupancy	LSTM	93.2–99.8%
This paper	Libelium parking(LoRa)	Signal Strenght	occupancy classification	HMM,NN	97% accuracy

The majority of papers focused their research in obtaining parking occupancy or availability prediction using the history of occupancy for a specific parking lot, containing date-time information with a specific occupancy status, as can be observed from the above presented table. Some researches do not employ a specific sensing device, but rather use public data sets that are provided [18,23,42,43] concentrating the goal of their study in finding the most appropriate Machine Learning technique for prediction or classification of a free parking space. They do not discuss or propose an overall technological architecture of their solution, but rather present a ML model based framework that can be employed in future systems. Moreover, it can be noticed that the majority of research used Neural Networks as a ML technique for the prediction or classification of free parking for a variety of data type. This is due to their ability to learn from complex, large scale structure and unclear information, which provides a high performance result, as shown in researches [18,27,28,40,45]. These researches point out that Neural Networks show high levels of accuracy in the prediction and classification of free parking space out performing other ML algorithms. The work presented in our paper gives a rather unique version of sensing the occupancy status, since it is based on the idea of eliminating the costly and energy hungry device with a beacon that will only send the data about the Received Signal Strength at a certain time. This distinguishes our work from the identified research in this filed in terms of used radio technology (LoRa) as well as the data type used for building the ML model. With that goal, it was decided to examine HMM and NN as ML approaches for classification of free parking. As previously elaborated, NN were selected, due to their dominant performance. Moreover, the Hidden Markov Model was employed due to its flexible mathematical structure, which makes a firm mathematical basis for modeling [46]. What is more, they are easy to implement (for instance, the Viterbi algorithm can be directly implemented as a computer algorithm) and they explicitly model the actual distribution of classes in classification problems, such as classifying a free parking space. With regards to radio technology, research of literature in [37] has shown that only 5% of researches up to date have employed LoRa for their Smart Parking Solution and, among these, none have used ML models for estimation or prediction. Although LoRa targets a wide range of applications, it has not yet been employed in a considerable amount in Smart Parking solutions [29]. The long range nature of LoRa technology allows for devices to communicate over larger distances (as far as 10 km) in comparison to XBee Series 2 and 802.15.4 radio technology presented in researches [27,38,40]. Hence, a single gateway device could simultaneously collect signal strength measurements data from multiple beacons scattered over a large parking lot, and classify parking status in real-time while using related Machine Learning techniques, thus enabling energy and cost savings.

### 3. LoRa-Based Smart Parking Sensor Device

In this paper LoRa radio technology was employed for transmitting information regarding parking lot occupancy. As a representative of a Low-Power Wide Area Networks (LPWANs), LoRa allows for battery-enabled devices such sensors to communicate low throughput data over long distances. As such, they are suitable for Smart Parking sensors deployed over a parking lot, since the information about parking status change can be transmitted to base stations (gateways) that are placed hundreds of meters from the parking lot. This enables a single gateway to potentially cover large parking area.

Figure 1 depicts the architecture of implemented LoRaWAN parking mechanism. The core of a Smart Parking sensor device is a commercial LoRaWAN-based Smart Parking sensor device from Libelium that comprises radar and magnetometer sensor for parking lot occupancy detection. These nodes are equipped with waterproof enclosure and they are fully powered with built-in lithium-thionyl chloride (Li-SOCl<sub>2</sub>) batteries with an overall capacity of 10.4 Ah that allows autonomous operation for a couple of years (Libelium: <https://www.libelium.com/iot-products/smart-parking/> (accessed on 20 November 2020)). Five Libelium parking sensor devices were placed at the surface of faculty parking lot. In the implementation of Libelium LoRaWAN parking sensor, the device periodically wakes up (every 60 s) and activates internal sensor devices (such as radar, magnetometer) for checking the change in parking status. If parking lot status changes (from free goes

to occupied or from occupied to free), the sensor device transmits the message over a radio while using LoRaWAN protocol to the gateway. In addition, Libelium parking devices employ keep-alive message transmissions, where the parking status is periodically sent if parking lot status does not change, for example, during nighttime hours.

As a LoRaWAN provider, The Things Network (TTN) was selected for its simplicity and good documentation. Furthermore, TTN forwards all of the messages from Libelium Smart Parking sensor to our personal server comprising Node-RED, InfluxDB, and Grafana services for visualization and further processing, as shown in Figure 1. Three TTN gateways were placed within a close vicinity of parking sensor devices, as shown in Figure 2 (left). Once the message arrives to the gateway (the base station), it is forwarded to the TTN Network and Application server, where the message payload is decoded and prepared for further processing and forwarding while using MQTT protocol or HTTP integration. In a given implementation, Node-RED was used for message aggregation from TTN. Afterwards, Node-RED parses and prepares the message for storage in the InfluxDB database. One entry into the database stores information regarding parking lot occupancy, timestamp entry (InfluxDB is a time series database), signal strength measurements on every gateway device (RSSI and SNR), and sensor ID. Figure 2 (right) shows a snapshot of parking lot occupancy along with RSSI measurement captured on three LoRaWAN gateways. As can be seen, when vehicle occupies a parking lot, a drop in RSSI values is detected at all gateways, which could serve as indicator of occupancy.

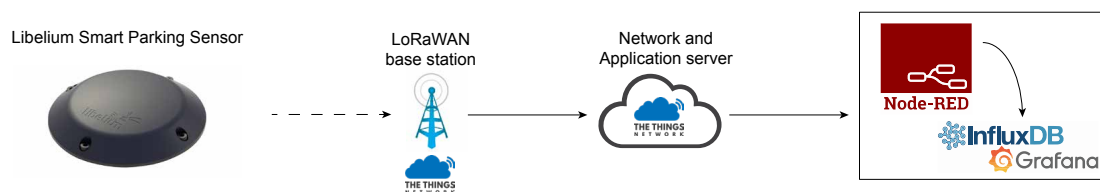


Figure 1. Network architecture of Libelium Smart Parking sensors.

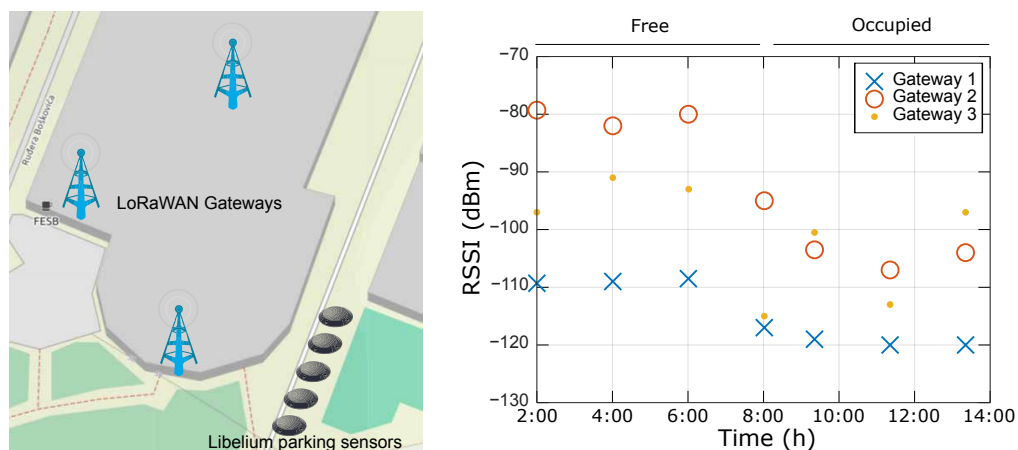


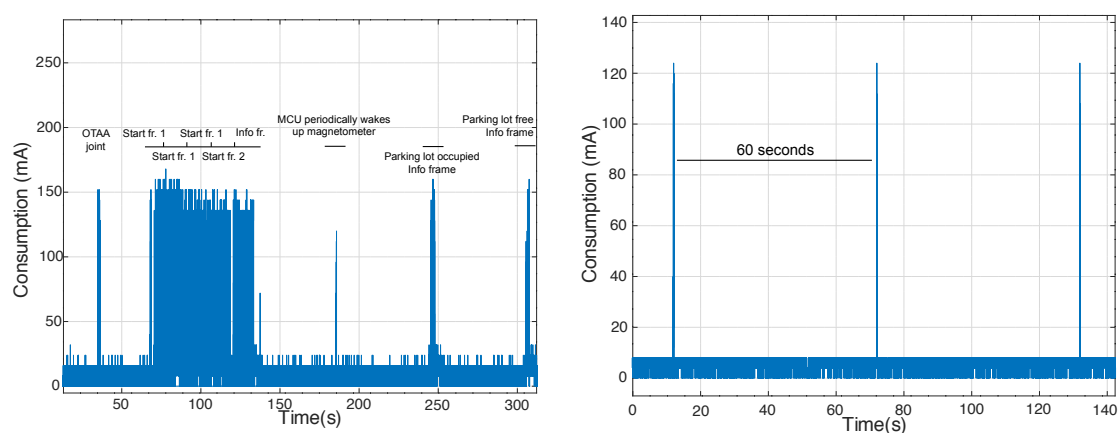
Figure 2. (left) Three LoRaWAN Gateways in the close vicinity of Libelium smart parking sensors, (right) Received Signal Strength Indicator (RSSI) captured on three LoRaWAN gateways from one Libelium Parking sensor device.

### Consumption of Libelium Smart Parking Sensor Device

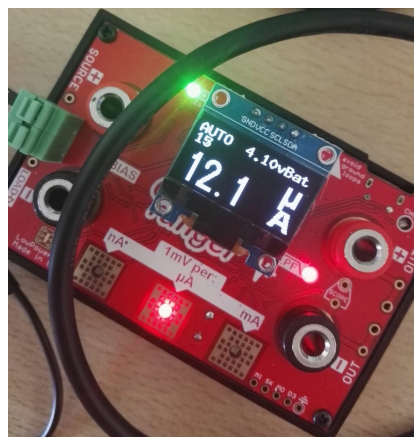
Figure 3 shows the consumption of Libelium smart parking sensor device that utilizes LoRaWAN radio technology for occupancy transmission detection. To capture detailed measurements of current consumption, node was connected to the oscilloscope via Current Ranger. As can be seen, device first utilizes LoRaWAN OTAA authentication protocol for establishing Network and Application session keys, which is followed by sending two Start frames and Info frame, as specified by the Libelium documentation. During active period, in which node sends occupancy status update, MCU with

sensors and radio peripheral (radar and magnetometer) are powered on, where the overall average consumption will slightly be above 100 mA. In contrast, during inactive period, where MCU with radio peripheral and sensors is in inactive mode, overall consumption falls to 12  $\mu$ A (Figure 4). The duty cycle (sleep period) of node is 60 s. After waking up, MCU powers the sensors and checks whether parking status occupancy changed from previous measurement. If parking lot occupancy has changed, MCU wakes up radio for sending status update. Otherwise, a node will enter into sleep mode.

Libelium parking sensor devices are equipped with lithium-thionyl chloride (Li-SOCl<sub>2</sub>) batteries that have an overall capacity of 10.4 Ah. Assuming consumption in sleep period is 0.012 mA, where device consumption in active period, on average, is 100 mA, along with one LoRaWAN message sent every 60 min and 6 s of wake-up duration, the device lifetime will be 2061.8 days, or 5.65 years. In this calculation, it is assumed that the capacity is automatically derated by 15% from 10.4 Ah in order to account for some self discharge.



**Figure 3.** (left) Libelium Smart Parking sensor consumption during the initial OTAA connection to the gateway. (right) Libelium Smart Parking sensor consumption between two microcontroller wakeup periods.

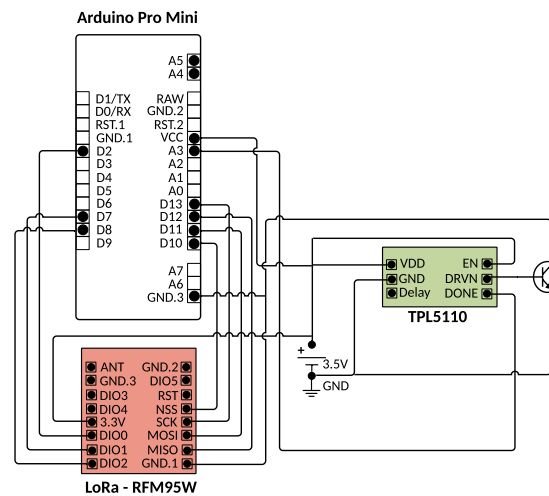


**Figure 4.** The consumption of Libelium Smart Parking sensor with Current Raider in sleep mode.

#### 4. Beacon-Based LoRaWAN Parking Sensor Device

In a concept of a cost effective smart parking occupancy detection device, Machine Learning techniques are employed for estimating occupancy from a beacon device. Such a novel solution will not require any sensor or sensor readings from devices, such as magnetometer and radar employed in Libelium smart parking sensor device. This beacon device would be comprised out of a simple MCU with LoRa radio module, as depicted in Figure 5. In order to minimize energy consumption during inactive period and periodically wake-up MCU from deep sleep, a TPL5110 Nano Timer could be

employed. During deep sleep, TPL5110 would cut off power from both the MCU and LoRa module, thus minimizing the overall consumption.



**Figure 5.** Scheme of LoRaWAN parking beacon device.

It is necessary to select MCU that supports the library for LoRaWAN message communication. Besides ATmega328P, which is standard MCU for Arduino Uno and Arduino Mini Pro, MCUs that employ libraries for LoRaWAN-based connection are also ATtiny 84, ATtiny 85 and STM32. Table 2 gives consumption comparison of MCUs during active period. Clearly, for the purposes of creating a simple beacon device, besides ATmega328P, which is found on Arduino Pro Mini, ATtiny 84 or ATtiny 85 could also be used, since their consumption is around 3mA in active state. During inactive state, the TPL5110 Nano Timer is selected, since its consumption is below 1  $\mu$ A. Table 3 depicts the energy consumption of every component that builds the Beacon Device. Because the beacon device does not hold any sensor, the active period basically comprises of waking up MCU and LoRa radio module, and sending LoRaWAN message, which can be reduced to approximately 5.5 s. In an active period, the average consumption is 25 mA, which includes 116 mA of consumption during LoRaWAN communication during a smaller portion of time, and 3 mA of MCU consumption during active period. Note that LoRa communication only occupies small portion of active period. During deep sleep period, device consumption could be around 4  $\mu$ A, which includes TPL5110 timer and low power voltage regulator. Assuming a duty cycle of 10 min., with a battery capacity of 10.4 Ah and 15% self discharge, the battery lifetime should be approximately 4.33 years.

Table 4 shows price of the proposed beacon device, and its components used in its development. As an alternative to TPL5110 timer, DS3231 low-power and low-cost RTC clock could be used in order to periodically wake-up Arduino from deep sleep which can lower the consumption of the complete beacon to below 1  $\mu$ A. The DS3231 can be found for a price of around 1 USD. Microcontroller that supports LoRaWAN library is ATmega328P, whereas its representative, Pro Mini, can be found in a price range of around 1.5 USD. In order to convey information over the radio, RFM95 LoRa module could be employed, with price of approximately 4.18 USD. Hence, the overall price of the module goes below 7 USD. For comparison, the price of FMCW Radar sensor device that is typically found in Libelium smart parking device is 14 USD, i.e., twice the price of the developed beacon device. Taking the price of magnetometer as well as the MCU and timer into account, the overall price goes well above the price of the proposed beacon device.

**Table 2.** Comparison of microcontroller (MCU) consumption that support LoRaWAN communication.

MCU	ATmega328P	ATtiny 84	ATtiny 85	STM32F103C8T6 (STM32)
Consumption (3.3. V and 8MHz)	3.9 mA	3 mA	3 mA	8 mA

**Table 3.** Consumption of every element of the beacon prototype along with lifetime duration estimation.

LoRa RF96 IC	116.1 mA
Aduino mini pro (ATmega328p)	4 mA
LDO	0.00377 mA
Timer TPL5110	0.000310 mA
Active period duration	5.5 s
Average overall consumption in active period	25 mA
Average consumption in inactive period	4 $\mu$ A
Lifetime duration (10.4 Ah battery)	1594 days

**Table 4.** Price of the overall beacon device.

Module	Price (USD)
ATMega328P Pro Mini	1.57
RFM95	4.18
DS3231	1.02
Overall	6.77

## 5. Experimental Setup and Data Analysis

### 5.1. Experimental Setup

For the purpose of collecting parking occupancy detection, five Libelium Smart Parking sensor devices were placed at the faculty parking lot next to each other (Figure 2 (left)). These parking sensor devices are placed in the center of the parking lot at the surface. The devices are equipped with a magnetometer and radar sensor devices, in such a way that, when the parking lot status changes (when vehicle approaches or leaves the parking lot), the sensing technology detects change and sends information over a radio channel. As a communication peripheral, Libelium devices employ LoRa radio capabilities in order to convey information regarding changes in parking lot occupancy. Besides sensing event driven packets, Libelium sensors also send keep-alive packets periodically every two hours. Three LoRaWAN gateways were placed in the radio range of Libelium parking sensors in order to collect data from Libelium devices. Two gateways were placed indoor within faculty facilities, while the third gateway was placed outdoor on top of the faculty. Namely, the first gateway was placed on the faculty first floor (4 m from the ground and around 30 m from the sensor), the second gateway was placed at the faculty fifth floor (around 15 m from the ground and 75 m from the sensor), while the third gateway was placed on the ninth floor of the faculty (around 30 m from the ground and 145 m from the sensor), as can be depicted in Figure 2 (left). Installed gateways employ TTN technology that allows for the collection of data from gateways and their storage into a designated database. The data were stored into the InfluxDB database, as shown in Figure 1. The collected data comprised information about parking lot occupancy status, Received Signal Strength Indicator (RSSI) in dBm, Signal to Noise Ratio (SNR) for every gateway, Gateway ID, Sensor ID, as well as the timestamp of the moment at which the data were received by TTN gateway. The data have been collected from five different sensors and three different gateways in period from 13th of December 2019 until 6th of September 2020. During that period, 130,984 raw data were collected from all five sensors. InfluxDB further allowed for the exportation of the collected data into csv. format for further processing. Machine learning techniques were employed on the collected data while using dedicated computing machine for performing such a task. Namely, Intel core i5-7300HQ@2.50GHz processor with 8GB of RAM and NVIDIA GTX1050 GPU running 64 bit Windows 10 operating system has

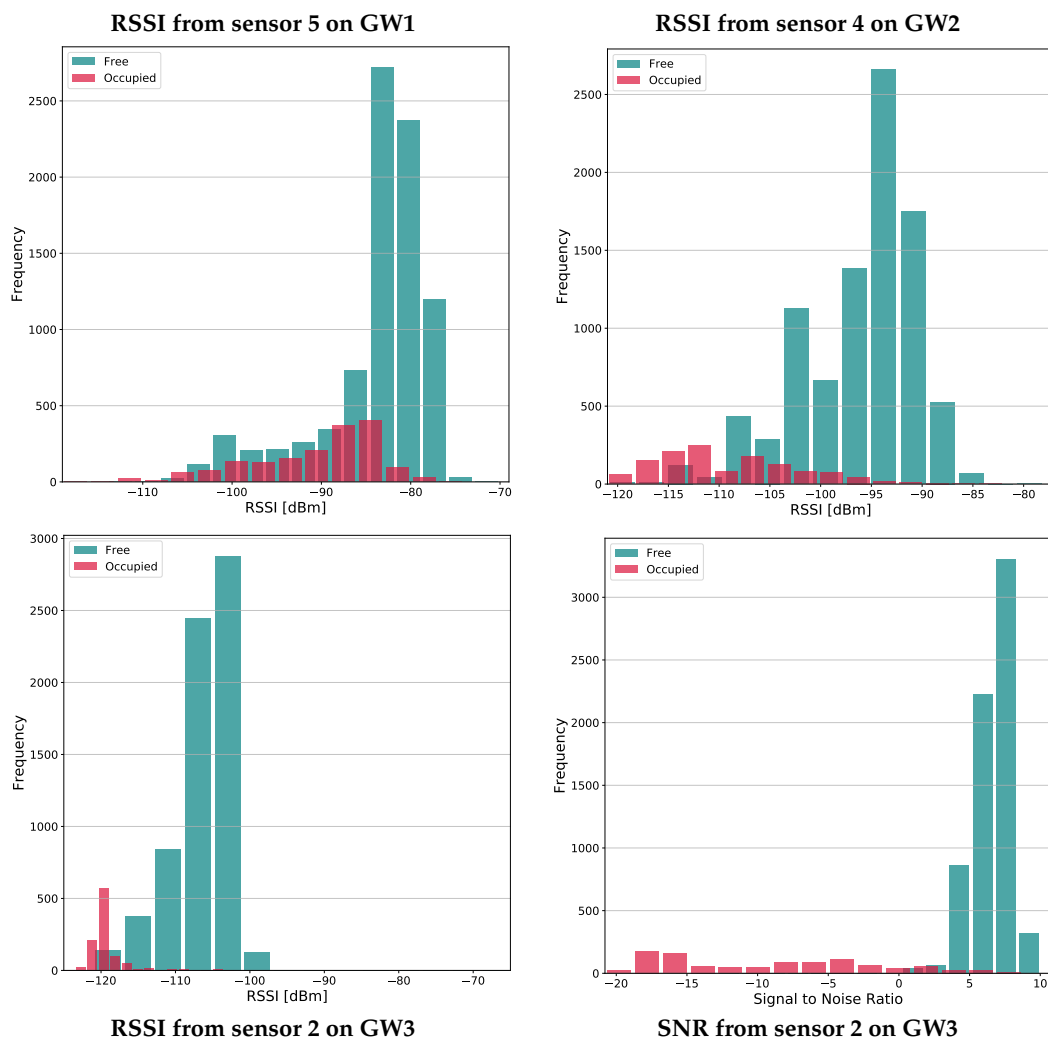


been used. NVIDIA CUDA Deep Neural Network library (cuDNN) was employed to utilize the performances of fast computing using GPU. The Keras 2.3.1. Python library was used running on top of a source build of Tensorflow 2.2.0 with CUDA support.me for different batch sizes.

### 5.2. Data Analysis

Because all of the gateways did not receive the same amount of data and, moreover, not in the same timestamp, it was decided to extract relevant data for each sensor and each of the three different gateways separately for the analysis.

In accordance with the goal of detecting how Occupancy Status is related with Received Signal Strength Indicator and Signal to Noise Ratio, it was proceeded with a plotting of the histograms presented in Figure 6 in order to gain a general illustration of the relationship between the aforementioned variables.

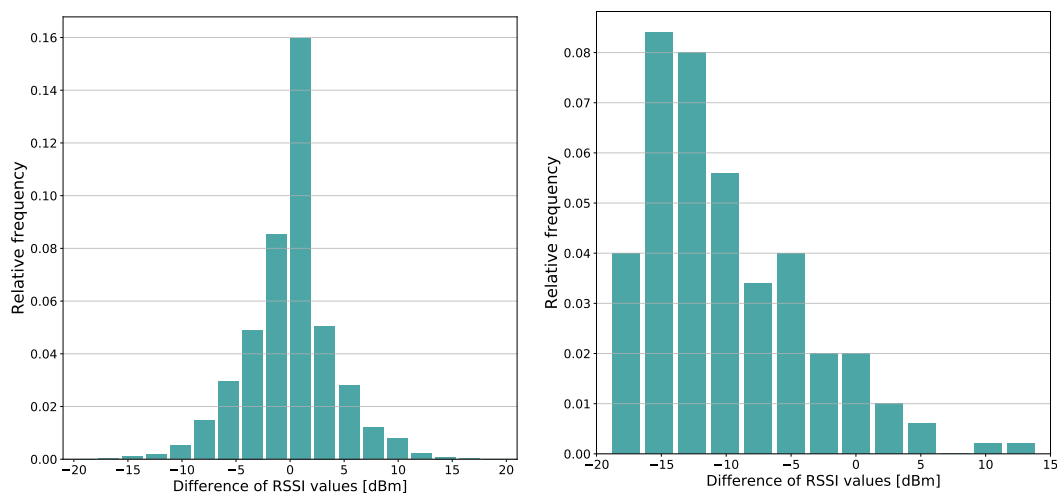


**Figure 6.** Histograms of RSSI and SNR values changes for different parking status and sensors from GW1, GW2 and GW3, respectively.

Firstly, it was noticed that parking lots are free considerably more than they are occupied. This is an important property of the parking place indicating its stochastic behavior. The parking is located on University grounds and, therefore, is usually free during night time or over the weekend periods. Secondly, the histograms depict that the RSSI values for free and RSSI values for occupied parking status overlap in Gateway 1 (GW1) and Gateway 2 (GW2). The same reasoning applies for SNR values for all sensors in GW1 and GW2. However, result that were gained for Gateway 3 (GW3) and sensors 2,

3, and 4 differ from above mentioned. The aforementioned sensors gave the least overlapping of RSSI and SNR values for a particular occupancy state. It was also noticed that higher RSSI values indicate a free parking space, while lower indicate the occupied one. The difference between the results gained for different gateways could be a consequence of the distance of GW1 (30 m), GW2 (75 m), and GW3 (145 m) from the parking sensor. GW3 is furthest away and on top of the University building and outdoor. This would imply that the closer the gateway, the channel influences RSSI and SNR stronger than the change of the parking status.

Because of the overlapping of RSSI and SNR values in different occupancy states, it was important to explore the change of RSSI and SNR values when the parking status does not change and when it changes from one state to another. Further analyses showed that, when parking status does not change, the values of RSSI and SNR change very little or not at all. However, when the parking status does change, there is a significant change in the RSSI and SNR values. Figure 7 presents histograms of changes of values for RSSI when parking space remains free and when parking space becomes occupied prior to being free for sensor 2 from GW3.



**Figure 7.** (left) Difference of RSSI values when the parking space remains free, (right) Difference of RSSI values for change of state from occupied to free for sensor 2 from GW 3.

In light of the above reasoning, the conclusions were twofold: (1) RSSI, SNR, and occupancy status are considerably correlated and (2) the adequate ML algorithm must be able to comprise the complexity of the data correlations in order to provide an appropriate estimation of occupancy status.

## 6. Machine Learning as an Approach to Parking Occupancy Detection

In the IoT paradigm of numerous smart connected devices, Machine Learning has emerged as an essential field of research and application aiming at providing computer programs the ability to automatically improve through experience [47]. The most distinguished attribute of a learning machine is that the trainer of learning machine is ignorant of the processes within it [48]. Machine learning generally includes data processing, training, and testing phases with the aim of making the system able to carry out decisions based on the input received from the training phase [13]. In order to archive the learning process, systems use various algorithms and statistical models to analyze the data and gain information about the correlation between the data features [12]. The algorithms that are used in these processes can be divided into four distinctive groups, as Supervised, Unsupervised, Semi-supervised, and Reinforcement learning algorithms:

- Supervised learning algorithms demand external monitoring by a supervisor with the goal of learning how to map input values to the output values where the accurate values are given by a supervisor [49].

- Unsupervised learning algorithms make computers learn how to perform a specific task only with the provided unlabeled data. These types of algorithms need to find existing relationships, irregularities, similarities, and regularities in provided input data [50].
- Semi-supervised learning is a hybrid approach of the previous two categories that uses both labeled data and unlabeled data. These algorithms generally act like the unsupervised learning algorithms with the improvements that are brought from a portion of labeled data [51].
- Reinforcement learning algorithms operate with a restricted insight of the environment and with limited feedback on the quality of the decisions. In order to operate effectively and provide the most positive outcome, these algorithms have the ability to selectively ignore irrelevant details [52].

ML has been ideally suited for various types of problems, such as as classification, clustering, predictions, pattern recognition, etc. The most appropriate ML algorithm is chosen based on the swiftness of the technique and its computational intensity, depending on the application type [12].

Nowadays, Deep Learning (DL) has become one of the leading Machine Learning techniques efficient in solving complex problems that have otherwise been impossible to solve while using more traditional ML approaches [13]. Deep Learning has been recognized as one of the ten breakthrough technologies of 2013 and fastest-growing trend in big data analysis [53]. Deep Learning applications have achieved remarkable accuracy and popularity in various fields, especially in image and audio related domains [13]. Deep Learning (DL) techniques effectively give insights from the data, comprehend the patterns from the data, and classify or predict the data [54]. Neural Networks that involve more than two hidden layers have been considered to be a characterization of DL and the word 'deep' signifies the large number of hidden layers that compose the Neural Network [53]. Implementations of Deep Learning technology today is achieving a large success in a variety of engineering and technical problems, including object detection, traffic engineering, traffic classification, and prediction [23,55–57].

### 6.1. Hidden Markov Model

Hidden Markov Models (HMMs) have been known for decades and, today, are making a large impact with regard to their applications, especially in form of Machine Learning models and applications in reinforcement learning. They are widely being used for pattern recognition [58], i.e., namely speech recognition [21] as well as in biological sequence analysis [59], gene sequence modeling, activity recognition [60], and analyses of ECG signal [61,62]. Markov Chains and process were first introduced by Markov in 1906 as a time-varying random phenomenon for which the Markov properties are attained. Its practical importance is the use of the hypothesis that the Markov property holds for a certain random process in order to build a stochastic model for that process [22].

In the broadest sense, a Hidden Markov Model (HMM) is a Markov process that can be divided into two parts: an observable component and an unobservable or hidden component. The observation is a probabilistic function of the state, i.e., the resulting model is a doubly embedded stochastic process, which is not necessarily observable, but it can be observed through another set of stochastic processes that produce the sequence of observations. A machine learning algorithm can apply Markov models to decision making processes regarding the prediction of an outcome.

In 1986, Rabiner and Juang [63] gave the structure of the first order Hidden Markov Model, denoted as  $\lambda(A, B, \pi)$ , where  $A = \{a_{ij}\}$  is the matrix of transition probabilities,  $B = \{b_j(k)\}$  is the matrix of observation probability distribution in each state, and  $\pi$  is the initial state distribution. Rabiner (1989) presented [64] three different types of problems in HMM: The Evaluation Problem, Decoding problem and Learning. The first problem is commonly solved by using the Forward or Backward algorithm, where as the last problem is, the most difficult of the three problems, usually solved while using Baum–Welch method. With regards to the second problem, the central issue is to find the optimal sequence of states to a given observation sequence and model used. The most common method to this is by using the Viterbi algorithm, which was introduced by Andrew Viterbi in

1967 as a decoding algorithm for convolution codes over noisy digital communication links. It is the answer to the decoding problem resulting in the Viterbi path, since the algorithm can be interpreted as a search in a graph whose nodes are formed by the states of the HMM in each of the time instant [22]. Let  $\lambda(A, B, \pi)$  be a HMM and  $O = (o_1, o_2, \dots, o_T)$  given observations. The Viterbi algorithm finds the single best state sequence  $q = (q_1, q_2, \dots, q_T)$  for the given model and observations. The probability of observing  $o_1, o_2, \dots, o_t$  using the best path that ends in state  $i$  at the time  $i$  given the model  $\lambda$  is:

$$\delta_t(i) = \max_{q_1, q_2, \dots, q_{t-1}} \mathbb{P}(q_1, q_2, \dots, q_{t-1}, q_t = i, o_1, o_2, \dots, o_t \mid \lambda) \quad (1)$$

$\delta_{t+1}(i)$  can be found using induction as:

$$\delta_{t+1}(i) = b_j(o_{t+1}) \max_{1 \leq j \leq N} [\delta_t(j) a_{ij}] \quad (2)$$

In order to return the state sequence, the argument that maximizes Equation (2) for every  $t$  and every  $j$  is stored in a array  $\psi_t(j)$  [63]. It is important to point out that the Viterbi algorithm can be directly implemented as a computer algorithm. Moreover, the algorithm succeeds in splitting up a global optimization problem, so that the optimum can be computed recursively: in each step, we maximize over one variable only, rather than maximizing over all  $n$  variables simultaneously.

Hidden Markov Models have been used now for decades in signal-processing applications, such as speech recognition, but the interest in models has been broaden to fields of all kind of recognition, bioinformatics, finance etc. [65].

With regards to the first order Markov model, if the past and the present information of the process is known, the statistical behavior of the future evolution of the process is determined by the present state. Thus, the past and future are conditionally independent (the system has no memory) [66]. Therefore, it is reasonable to ask whether there can be a model that can gather and somewhat keep information from the past. The answer lies within a higher-order Markov models, where the hidden process is a higher order Markov chain and it is dependent on previous states. This gives memory to the model and such a modeling is more appropriate for processes in which memory is evident and important, for example, a stock market time series.

## Model and Results

The collected and visualised data, as presented in Data analyses section, revealed the general proprieties of our data, their correlation, and enabled us in designing the appropriate model for the Machine Learning approach in reaching the desired goal. The aim is to determine the occupancy of a parking space based solely on Received Signal Strength Indicator and Signal to Noise Ratio values. Hidden Markov Model of second order, which is presented in the following, was designed and used in order to classify the occupancy status of a parking space, while using RSSI and SNR values.

From previously presented and discussed histograms of RSSI and SNR values with regards to occupancy status, it was concluded that, when parking status does not change, the values of RSSI and SNR change very little or, in most cases, not at all. In contrast, when the parking status does change, there is a significant change in RSSI and SNR values. Therefore, the variables “bring memory” with them that is dependent of the previous state of occupancy. The process it self is of a time series that can be designed and modeled using a second-order HMM. In this model, the Hidden States are the aim of prediction, which is Occupancy status. In order to “bring memory” into our model, the Observable (Visible) States are defined to be the changes of RSSI (the same reasoning and model applies for SNR) values that are calculated as the difference between these values from two previous states. The notation and model illustration are as follows:

- $F$ —free,  $O$ —occupied.
- $RSSI_s$ —value of Received Signal Strength in occupancy state  $s$  in a timestamp.

- $\Delta RSSI = RSSI_s - RSSI_{s+1}$ . Difference between two values of RSSI in two consecutive occupancy states.
- *FF*—state that is free which previous state was free.
- *FO*—state that is occupied after previously being free.
- *OF*—state that is free after being occupied
- *OO*—state that is occupied which previous state was occupied.

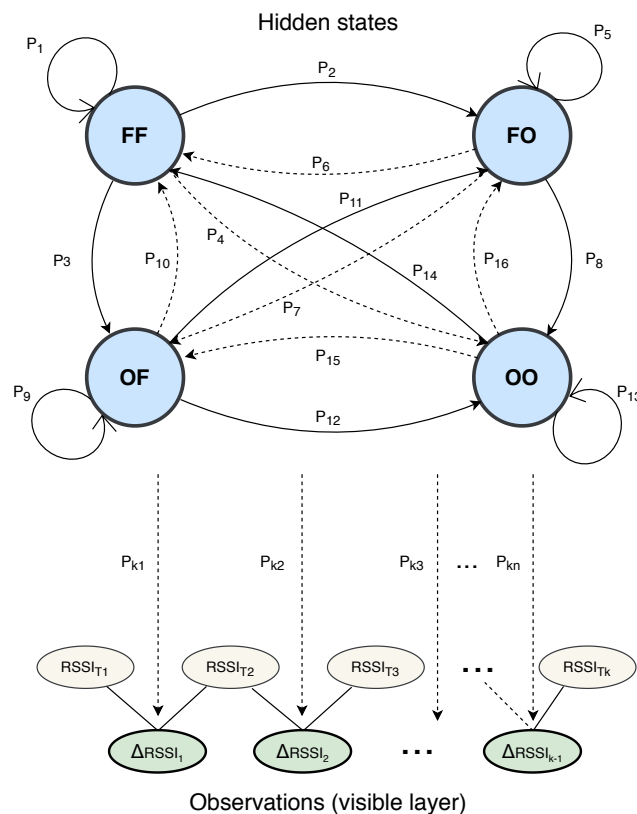
States *FF*, *FO*, *OF*, and *OO* bring with them “memory of occupancy”, since they remember what was the occupancy status from the “past”. These states represent Hidden Layer of states. The Hidden Markov Model model is denoted as  $\lambda (A, B, \pi)$ , where:

- *A* is the transition matrix. It stores probabilities of transition from one state to another. The matrix holds some zero values, due to the fact that some transitions are impossible. For instance, you cannot transit from state *FF* into the state *OO*,
- $\pi$  is the initial state distribution (stationary distribution) and it is calculated by solving the matrix equation

$$\pi = \pi \cdot A.$$

- *B* is the matrix containing the observation probability distribution in each state. In this model, the observations are the changes of *RSSI* values in two consecutive occupancy states— $\Delta RSSI$ .

Figure 8 visualizes the architecture of our second order HMM.



**Figure 8.** Illustration of second-order Hidden Markov Model for detecting occupancy status based on change of *RSSI* values.

As stated, because it was decided to extract the relevant data for each sensor and each of the three different gateways separately, the implementation took all of these possibilities into account. The used decoding algorithm for finding the optimal sequence of states to a given observation sequence and model is previously defined Viterbi algorithm. All of the data were effectively used as an observation

for a chosen step and given as input to the Viterbi algorithm. The chosen step determines the length of observation sequence. For example, if the chosen step is 4, then the whole data set from selected sensor and gateway is divided into subsets of sequences containing four consecutive values of a chosen variable (RSSI or SNR). Every one of this sequences is then given as an observation input to the Viterbi algorithm.

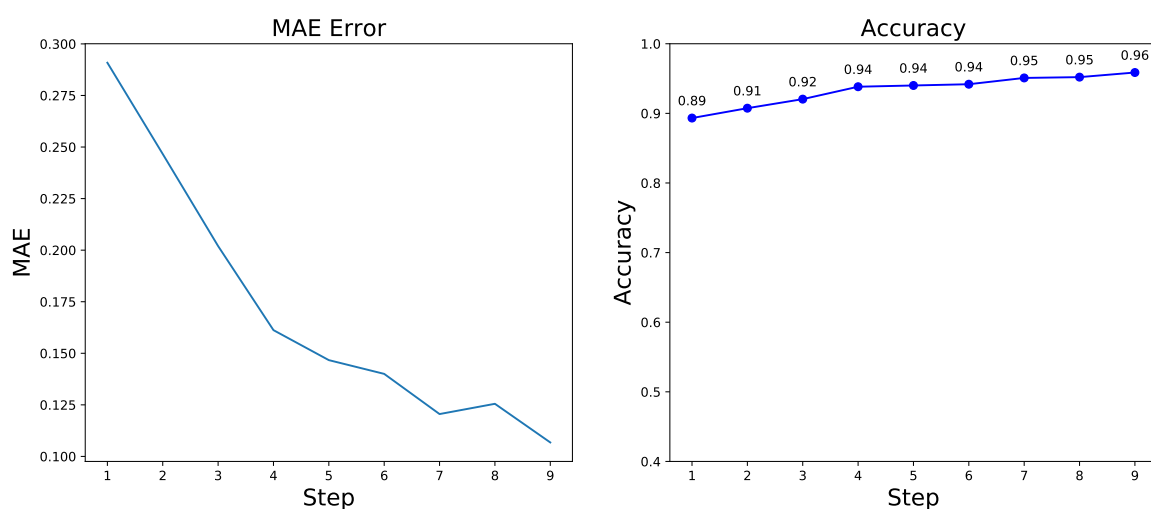
The classified and the true values are stored separately and the accuracy is calculated while using accuracy score function. This function computes subset accuracy, which is the fraction of samples classified correctly. The set of labels classified for a sample must exactly match the corresponding set of labels of true values. Moreover, the model’s evaluation is done while using Mean Absolute Error (MAE). The model was tested for all variables from all sensors and gateways, and the best results are given in Table 5.

**Table 5.** Table of best results using the Hidden Markov Model (HMM) model obtained for each gateway.

Gateway	Variable (Sensor Number)	Accuracy (Best Results)	MAE
GW1	RSSI (4)	87%	0.30
GW1	SNR (4)	87%	0.35
GW2	RSSI (3)	89%	0.27
GW2	SNR (3)	92%	0.20
GW3	RSSI (2)	93%	0.17
GW3	SNR (2)	96%	0.11

The least promising results were gained from the closest gateway GW1, as can be seen from the table. On GW1, the second HMM model only reached 87% accuracy with a MAE of 0.30. This is due to the previously explained overlapping in the RSSI (or SNR) values with regards to different occupancy status. With regards to GW2, the best results are slightly better with regards to Accuracy and MAE than GW1. This also is consistent with the reasoning of overlapping values for different occupancy status.

Finally, the best results were obtained for the furthest GW3, giving 96% accuracy for observation values of SNR from sensor 2 and sensor 4 and MAE of 0.17 and 0.11, respectively. Figure 9 illustrates the best result that is obtained while using the HMM model from sensors 2 from Gateway 3.



**Figure 9.** Mean Absolute Error (MAE) and classification Accuracy for SNR values from sensors 2 from Gateway 3.

Despite the results that were obtained while using the second order HMM, this approach has limitations; the states must be drawn from a modestly sized discrete state space and each hidden state

can depend only on the immediate previous state [26]. In order to model  $N$  bits of information about the past history, HMM requires  $2^N$  hidden states [67], which makes it computationally impractical for large data sets. HMM are generative classifiers which means that they explicitly model the actual distribution of each class, in contrast to discriminative models, such as Neural Networks, which model the decision boundary between the classes [68]. Discriminative models can provide robust solutions for non-linear discrimination in high-dimensional spaces [69] and they have been shown to be quite effective for applications in classification [70]. Therefore it reasonable to examine Neural Networks as another approach that can encompass complex, high-dimensional, and noisy real-world data.

## 7. Neural Network Models

Neural Networks, or Artificial Neural Networks (ANN), have gained significant attention in the last two decades as a Machine Learning technique in a variety of areas for prediction and classification task [12]. Inspiration for their architecture was taken from the brain nervous system in a form of a mathematical model that is designed to mimic the structure and functionalities of the real biological Neural Networks [71]. They have been applied in many divers areas of scientific research, such as pattern recognition [72], image classification [73], language processing [74], computer vision [75], as well as time series forecasting [76].

Generally, the Neural Network consists out of three basic layers as shown in Figure 10, namely the input layer, the hidden layers, and the output layer. The Neural Network can have more than one hidden layer, which represents the depth of the Neural Network. The imitation of the brain learning processes is done by searching the hidden links between a series of input data while using hidden layers of neurons, where the output of a neuron of a layer becomes the input of a neuron of the next layer. An artificial neuron  $y_i$  can be defined as a function

$$y_i = f_i(x) = \varphi(\langle w_i, x \rangle + b_i), \quad (3)$$

which acts on a linear combination of the input vector  $x = (x_1, \dots, x_n)$  and a neuron bias  $b_i$  [77]. The input vector is weighted with the connection weight vector  $w_i = (w_{1,i}, \dots, w_{n,i})$  and the  $\varphi$  is called activation function. The performance of the training process and estimation (or prediction) accuracy of the NN is highly influenced by the weight initialization and activation function [78]. The activation function will control the amplitude of the output of the neuron, keeping it in a usually acceptable range of  $[0, 1]$  or  $[-1, 1]$  [79]. The activation functions are divided into linear and non-linear activation function and non-linear ones are most commonly used. Some of most frequently applied non-linear activation functions are Sigmoid, Rectified Linear Unit (ReLU), and Tanh function. Sigmoid function can be defined as  $S(x) : \mathbb{R} \rightarrow [0, 1]$

$$S(x) = \frac{1}{1 + e^{-x}}.$$

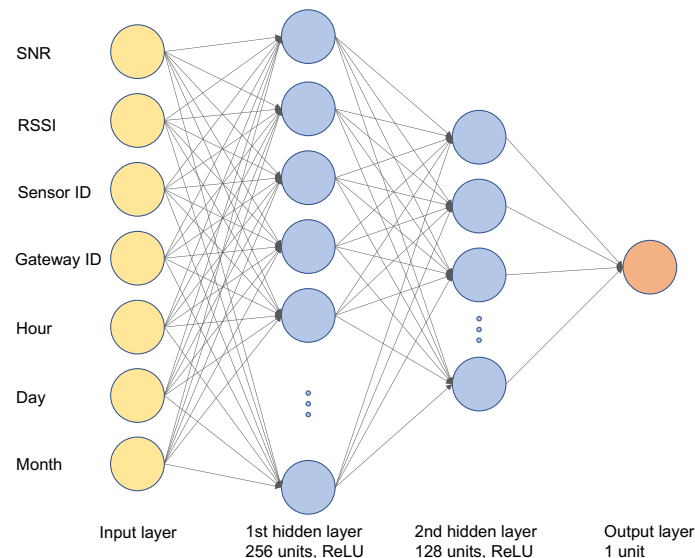
The Sigmoid function is continuously differentiable, but it suffers from gradient vanishing [78], which can significantly slow down the learning process. This problem has been resolved while using the ReLU activation function.

ReLU function can be defined as  $\phi : \mathbb{R} \rightarrow \mathbb{R}^+$

$$\phi(x) = \begin{cases} \lambda x, & x > 0 \\ \beta x, & x \leq 0 \end{cases},$$

where commonly  $\lambda = 1$  and  $\beta = 0$ . The derivative of the function will be quite simple, 1 for positive values and 0 otherwise, as can be seen from the function's definition. Therefore, the average derivative is rarely close to 0, which allows gradient descent to keep progressing. Hence, ReLU has been mainly used as an activation function for the neurons that are placed in hidden layers [78], while Sigmoid has been used as a activation function for the neurons that are placed in the output layer. This paper implements a Neural Network comprised out of two hidden layers (Figure 10). The input layer takes

data, such as the sensor ID, RSSI, and SNR, of the LoRa packet sent from the sensor to the Gateway, and ID of that gateway, along with the timestamp of the event when the packet was sent. The exit layer predicts parking space occupancy (free or occupied).



**Figure 10.** Architecture of Neural Network model for parking space occupancy classification.

As seen, the layers comprise artificial neurons, where every neuron has multiple weights and some form of transfer or activation function. The Neural Network is a supervised learning algorithm, in which the weight of the neurons is calculated during the training process. Because of the training procedure, the input data to the network should cause the output as close to the ground truth. In order to accomplish this, during the training procedure, which is an iterative procedure, a loss (cost) function is used to determine the quality of the network with specific weights. For a binary classification problem, such as parking lot occupancy, Binary Cross-Entropy Loss, as one of the commonly used loss functions, and it has been utilized in this research.

In order to minimize loss function during the training phase in which the weight of neurons is determined, a good deal of optimization algorithms have been implemented, many of which are first-order iterative optimization algorithms. The algorithms used in this paper were Stochastic Gradient Descent (SGD), Adaptive Moment Optimization (Adam), and Root Mean Square Propagation (RMSProp).

### 7.1. Evaluation Metrics

The proposed Neural Network model has been evaluated while using different metrics to evaluate different characteristics of the classifier. Namely, the metrics used were Accuracy, F1 score, Area under the Receiver Operating Characteristic Curve Accuracy (ROC AUC) and Average Precision (AP).

- Accuracy—it is defined as the overall accuracy or proportion of correct predictions of the model and it is given with the formula:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN'} \quad (4)$$

where  $TP$  and  $TN$  denote the number of positive and negative instances that are correctly classified.  $FP$  and  $FN$  denote the number of misclassified negative and positive instances, respectively.

- F1 score—F1 score is the harmonic mean of the Precision and Recall. Precision is defined as the number of correct predictions out of all the predictions based on the positive class, whereas Recall



is the number of instances of the positive class that were correctly predicted [13]. F1 score is calculated while using formula:

$$F1\ score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \quad (5)$$

The F1 score takes values from the  $[0, 1]$  interval, reaching minimum for  $TP = 0$ , that is, when all the positive samples are misclassified, and the maximum for  $FN = FP = 0$ , which is for perfect classification [80].

- ROC AUC—the Receiver Operator Characteristic (ROC) curve is an evaluation metric for binary classification problems and it is a probability curve that is created by plotting the True Positive Rate (TPR) versus the False Positive Rate (FPR) [13]. The Area Under the Curve (AUC) represents a separability measure of classifiers, i.e., the ability of the classifier to distinguish between classes [81]. The ideal classifier will have the unit area under the curve and a worst case classifier will have  $FPR = 100\%$  and  $TPR = 0$  [13].
- Average Precision—it is the measure that considers both Recall and Precision and can be expressed as a function  $p(r)$  of the recall and it is given with [82]:

$$Average\ Precision = \int_0^1 p(r) dr. \quad (6)$$

## 7.2. Results and Discussion

Data used for building the NN model were previously described in Data Analysis section. The pre-processing of data comprised of data normalization due the different value scales of variables in the collected data. The inputs to the model were values of RSSI and SNR for a specific sensor and gateway, whereas the target values were numeric values of parking lot occupancy (0—free, 1—occupied). Therefore, for each of the sensors, data from all three gateways were given as input. There is a slight imbalance regarding the number of instances of each class, depending on the sensor and gateway, as was observed in Data Analyses section. This does not represent a problem for (GW3), since it gives the smallest overlapping of RSSI and SNR values for a particular occupancy state, and good results can be obtained, regardless of class disproportion, if both groups are well represented and their distributions do not entirely overlap [83]. Therefore, the data were first split into training and test set while using stratification in order to preserve the distribution of classes in training and test set, with the test set size being 10%. Moreover, the training set was further split into train and validation set also using stratification, with the validation set size being 10%. Stratification will equalize the ratios of the number of training and validation samples for each class and it is able to achieve lower biases and small variances in estimated accuracies [84], providing consistent predictive performance scores. This way, potential biases that could be caused by the some imbalance in the data set are minimized.

Different optimizers, namely, Adaptive Moment Optimization (Adam), Root Mean Square Propagation (RMSprop), and Stochastic Gradient Descent (SGD), were tested, as well as other hyper-parameters that are presented in Table 6.

**Table 6.** Selection of the hyper parameters for evaluation.

Hyper Parameter	Values
Number of neurons	Layer1—256, Layer2—128
Learning rate	0.001 , 0.01
Number of epochs	50, 100, 150
Batch size	64

The first experimental results of the Neural Network model exposed that Adam, as an optimizer, has achieved the best performance results. This is reported for all sensors in Table 7. As can be noticed,

the highest Accuracy and ROC AUC were again achieved for sensors 2 and sensor 4, namely 91% and 94% Accuracy, respectively, for 100 epochs, and a learning rate of 0.001 on the Test and Validation set. These results seem to be rather consistent with the result that was obtained with the second order HMM model.

**Table 7.** Results of first Neural Network model for Adaptive Moment Optimization (Adam) optimizer.

Sensor ID	Learn. Rate	Epochs	TRAINING				VALIDATION				TEST			
			Acc.	F-Score	AUC	AP	Acc.	F-Score	AUC	AP	Acc.	F-Score	AUC	AP
1	0.001	50	0.885	0.471	0.896	0.638	0.883	0.503	0.925	0.750	0.869	0.448	0.892	0.647
1	0.001	100	0.885	0.546	0.904	0.662	0.889	0.551	0.891	0.634	0.897	0.570	0.908	0.679
1	0.01	50	0.879	0.395	0.898	0.635	0.869	0.412	0.899	0.646	0.874	0.394	0.892	0.626
1	0.01	100	0.883	0.512	0.893	0.633	0.880	0.522	0.908	0.674	0.879	0.538	0.907	0.651
2	0.001	50	0.911	0.576	0.934	0.749	0.921	0.608	0.931	0.745	0.914	0.573	0.938	0.749
2	0.001	100	0.915	0.592	0.935	0.754	0.911	0.590	0.936	0.765	0.915	0.587	0.942	0.766
2	0.01	50	0.912	0.546	0.935	0.753	0.913	0.527	0.934	0.742	0.905	0.512	0.929	0.742
2	0.01	100	0.911	0.547	0.931	0.745	0.917	0.585	0.937	0.771	0.914	0.566	0.932	0.734
3	0.001	50	0.912	0.601	0.933	0.744	0.908	0.621	0.940	0.781	0.911	0.568	0.928	0.720
3	0.001	100	0.913	0.560	0.916	0.714	0.913	0.560	0.901	0.718	0.906	0.519	0.906	0.711
3	0.01	50	0.911	0.540	0.911	0.708	0.906	0.487	0.902	0.675	0.919	0.575	0.911	0.716
3	0.01	100	0.911	0.525	0.912	0.702	0.912	0.557	0.918	0.743	0.919	0.571	0.902	0.717
4	0.001	50	0.937	0.749	0.952	0.833	0.940	0.756	0.956	0.847	0.937	0.750	0.945	0.835
4	0.001	100	0.935	0.729	0.953	0.838	0.943	0.758	0.959	0.858	0.936	0.701	0.957	0.838
4	0.01	50	0.936	0.745	0.954	0.840	0.937	0.733	0.943	0.802	0.935	0.738	0.946	0.819
4	0.01	100	0.935	0.729	0.956	0.836	0.933	0.708	0.956	0.823	0.941	0.762	0.959	0.856
5	0.001	50	0.852	0.306	0.807	0.555	0.862	0.343	0.829	0.585	0.845	0.311	0.816	0.566
5	0.001	50	0.852	0.312	0.814	0.566	0.854	0.324	0.817	0.560	0.859	0.352	0.817	0.575
5	0.01	50	0.853	0.291	0.805	0.551	0.837	0.312	0.792	0.581	0.856	0.294	0.803	0.543
5	0.01	50	0.852	0.436	0.808	0.551	0.849	0.447	0.805	0.574	0.848	0.449	0.823	0.580

The result of the Area Under the Curve point out that the Neural Network performs well as a classifier with just RSSI and SNR values as an input. However, HMM achieved better Accuracy, since it contained “memory” of previous occupancy. Consequently, it was decided to upgrade the model with more input using the time variables, namely, hour, day and month. Therefore, for each sensor hour, day and month for a specific occupancy was taken into the account. Time variables can grasp effects such as seasonality and temporal dependence, giving a more in-depth display of occupancy history. The data were again pre-processed in the previously described manner.

The second experimental results that were obtained are presented in Table 8. These were accomplished for a learning rate of 0.001, 100 epochs and Adam optimizer. As can be noticed, incorporating time data into the model resulted in better classification performance. Yet again, the highest Accuracy and AUC was gained for sensor 2 and sensor 4. For sensor 2 and sensor Accuracy and AUC on test set was 96% and 98% respectively, where as on the validation set Accuracy and AUC was 95% and 98% for sensor 2 and 97% and 98% for sensors 4. Moreover, it can be noticed that the Accuracy and AUC have risen up for all other sensors when time data were included, consequently justifying our reasoning for their incorporation. A high F1 score on test and validation set for all sensors implies rather good precision and recall, and an overall high AUC indicates that the presented Neural Network model is very good in distinguishing between classes. This is confirmed with high Average Precision, which indicates that NN correctly handles positives.

**Table 8.** The results of second Neural Networks (NN) model for Adam optimizer, learning rate 0.001 and 100 epochs.

	TRAINING				VALIDATION				TEST			
	Acc.	F-Score	AUC	AP	Acc.	F-Score	AUC	AP	Acc.	F-Score	AUC	AP
Sensor 1	0.958	0.853	0.987	0.942	0.960	0.865	0.986	0.935	0.955	0.831	0.983	0.911
Sensor 2	0.971	0.888	0.993	0.962	0.954	0.814	0.980	0.905	0.962	0.857	0.989	0.942
Sensor 3	0.962	0.845	0.988	0.939	0.955	0.820	0.981	0.889	0.961	0.838	0.980	0.904
Sensor 4	0.972	0.896	0.993	0.963	0.970	0.883	0.989	0.947	0.969	0.882	0.989	0.954
Sensor 5	0.945	0.838	0.983	0.932	0.933	0.809	0.975	0.911	0.942	0.832	0.978	0.921

In accordance with previously discussed results, a final model of Neural Network was designed and tested. This model would give an overall parking lot classification. The model encompassed all of the collected data, namely sensor ID, RSSI, and SNR values for all three gateways and time variables of a specific occupancy status of a particular sensor, as can be seen in Table 9.

**Table 9.** Snapshot of a dataframe used for the model of Neural Network.

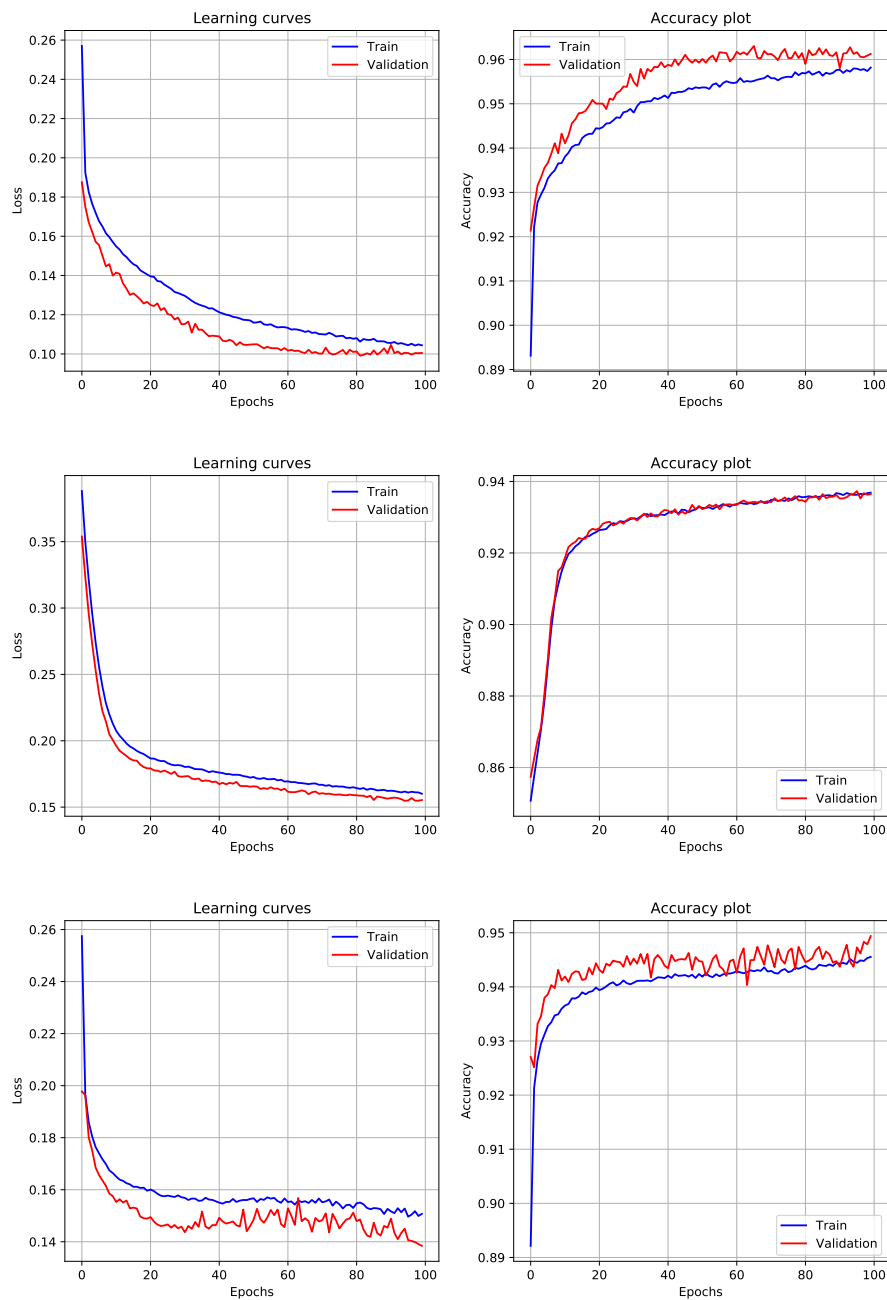
Time	Sensor ID	Month	Day	Hour	GW ID	RSSI [dBm]	SNR	Status
2019-12-13 14:52:00	5	12	13	14	1	−112.0	4.2	0
2019-12-13 14:52:00	5	12	13	14	2	−80.0	8.0	0
2019-12-13 14:53:00	5	12	13	14	1	−111.0	4.2	0
2019-12-13 14:53:00	5	12	13	14	2	−85.0	7.0	0
2019-12-13 14:55:00	3	12	13	14	1	−120.0	−5.2	1
...	...	...	...	...	...	...	...	...
2020-09-06 18:16:00	3	9	6	18	2	−90.0	6.2	1
2020-09-06 18:18:00	2	9	6	18	2	−95.0	9.8	0
2020-09-06 18:31:00	5	9	6	18	2	−80.0	8.8	0
2020-09-06 19:21:00	4	9	6	19	2	−81.0	8.5	0
2020-09-06 19:45:00	1	9	6	19	2	−91.0	7.8	0

The data were further pre-processed in a similar manner as in the first two versions of the model, with one major difference. In the process of train, test, and validation split, it was important in order to ensure that the amount of data from all sensors was equally distributed. Therefore, stratification was done with regards to sensor ID. Different combinations of optimizers, learning rates, and epochs were again tested and the results are presented in the Table 10.

**Table 10.** Result of different combinations of optimizers, learning rates and epochs for the final Neural Network model.

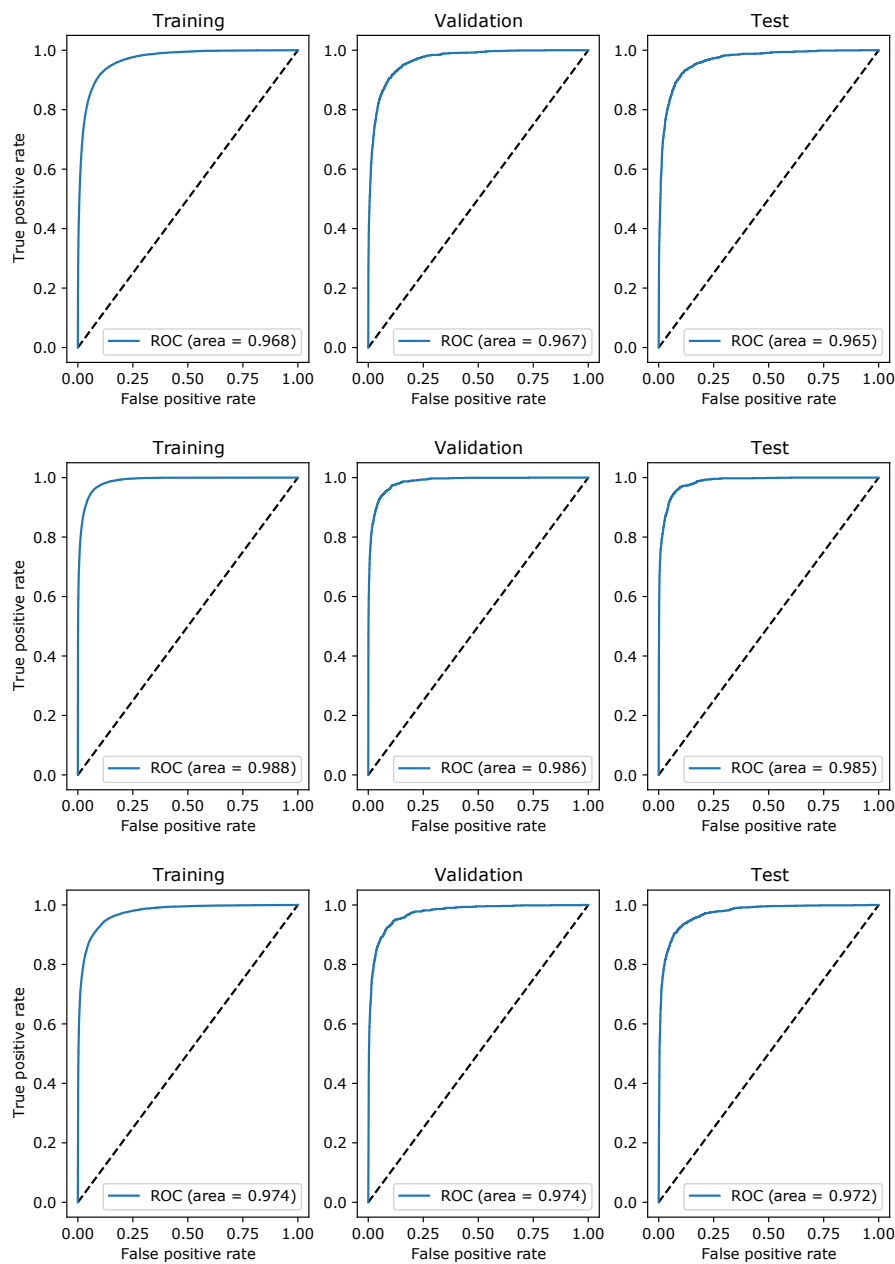
Opt.	Learn. Rate	Epochs	TRAINING				VALIDATION				TEST			
			Acc.	F-Score	AUC	AP	Acc.	F-Score	AUC	AP	Acc.	F-Score	AUC	AP
sgd	0.01	50	0.944	0.798	0.974	0.887	0.946	0.803	0.975	0.891	0.941	0.798	0.972	0.886
sgd	0.01	100	0.954	0.837	0.982	0.919	0.953	0.833	0.982	0.918	0.950	0.831	0.979	0.912
sgd	0.001	50	0.932	0.731	0.963	0.848	0.929	0.718	0.961	0.845	0.932	0.726	0.957	0.827
sgd	0.001	100	0.936	0.753	0.968	0.866	0.936	0.747	0.967	0.866	0.939	0.756	0.965	0.856
sgd	0.001	150	0.941	0.799	0.971	0.877	0.938	0.795	0.970	0.880	0.945	0.809	0.972	0.878
adam	0.01	50	0.952	0.836	0.982	0.918	0.949	0.826	0.980	0.911	0.945	0.813	0.977	0.901
adam	0.01	100	0.955	0.842	0.985	0.930	0.952	0.828	0.980	0.909	0.951	0.826	0.981	0.913
adam	0.001	50	0.956	0.834	0.985	0.931	0.949	0.809	0.981	0.916	0.949	0.812	0.981	0.915
adam	0.001	100	0.961	0.862	0.988	0.944	0.961	0.857	0.986	0.936	0.957	0.851	0.985	0.934
adam	0.001	150	0.964	0.868	0.990	0.954	0.956	0.841	0.983	0.930	0.955	0.838	0.981	0.923
rmsprop	0.01	50	0.939	0.772	0.964	0.856	0.937	0.766	0.959	0.846	0.934	0.764	0.962	0.854
rmsprop	0.01	100	0.940	0.775	0.964	0.861	0.939	0.776	0.965	0.865	0.943	0.787	0.970	0.875
rmsprop	0.001	50	0.946	0.809	0.972	0.887	0.945	0.808	0.971	0.888	0.942	0.789	0.966	0.882
rmsprop	0.001	100	0.950	0.816	0.974	0.900	0.949	0.818	0.974	0.897	0.950	0.814	0.972	0.886
rmsprop	0.001	150	0.948	0.802	0.973	0.894	0.945	0.787	0.971	0.888	0.943	0.788	0.970	0.883

Adam achieved best performance for a 0.001 learning rate and 100 epochs, which is consistent with our previously obtained results. Specifically, this combination reached 96% and 95% Accuracy on the validation and test set, respectively, and 96% AUC on both the validation and test set. Figure 11 visualizes the learning curves on the train and validation set and Accuracy plot with regards to different optimizers for a learning rate of 0.001 and 100 epochs. It can be noticed that, for the Adam optimizer, the Accuracy plot on the Train and Validation set seem to overlap, and the learning curves are almost in a optimal fit.



**Figure 11.** (left) Learning path of model with training and validation loss, (right) Accuracy plot for Stochastic Gradient Descent (SGD), Adam, and RMSprop optimizers respectively with the learning rate of 0.001 and 100 epochs.

Finally, the performance of the Neural Network as a classifier, for a different combination of optimizers for a learning rate of 0.001 and 100 epochs, is evaluated with the ROC curve visualized in Figure 12. As can be seen, with Adam optimizer, NN as a classifier is able to achieve the highest TPR while maintaining a low FPR. High AUC of 98% implies that the final Neural Network model is able to distinguish occupied and free parking space exceptionally well.



**Figure 12.** Receiver Operating Characteristic (ROC) curves for SGD, Adam, and RMSprop optimizers, respectively, with the learning rate of 0.001 and 100 epochs.

Lastly, we have compared our results with other researches that we have identified in the State of the Art section that have also used Neural Network as a ML technique for the classification or prediction of a free parking space. We have compared them with our results in terms of achieved accuracy and presented the comparison in Table 11.

**Table 11.** Comparison table of this paper with other research in terms of accuracy for NN.

Paper	ML Model	Given Accuracy for NN
Vlahogianni et al. [27]	NN (prediction)	0.004 MAE
Farag et al. [28]	NN (classification)	93%
Jones et al. [39]	DT,SVM, k-NN (classification)	81%
Hiesmair et al. [40]	NN, DT, k-NN, SVM (classification)	95% accuracy
Zheng et al. [43]	RT, NN, SVR (prediction)	Mean MAE 0.194–0.059
This paper	HMM, NN (classification)	97% accuracy

The researches [27,43] have used Neural Network for future prediction and, therefore, have presented their results in terms of MAE. As can be seen from the table, this paper achieved best performance results in terms of accuracy of NN for classifying a parking space, in comparison to [28,39,40].

## 8. Conclusions

This paper presents a novel software alternative concept of cost-effective sensor device for parking lot occupancy detection. Namely, a LoRa-based smart parking sensor device measured parking lot occupancy during the period of several months. The parking lot occupancy was sent over a radio channel to three LoRaWAN gateways that collected measurements of signal strength from five sensor devices that were placed on a University parking lot. The analysis of collected data indicates a correlation between RSSI, SNR, and parking lot occupancy. Using related machine learning techniques, it was shown that parking lot occupancy can be estimated from signal strength measurements. Using the Hidden Markov Model with Viterby algorithm and Deep Learning approach based on Neural Networks showed significant results up to 97% of correctly estimating parking lot occupancy. Our future work will comprise an exploration of other ML techniques for parking space classification, which we further plan to evaluate and compare to the results that we have obtained in this paper.

**Author Contributions:** The individual contributions of each author are provided as follows: conceptualization, L.D.R. and T.P.; methodology, L.D.R. and T.P.; software, L.D.R. and T.Ž.; validation, L.D.R., T.P. and P.Š.; formal analysis, L.D.R. and T.P.; investigation, L.D.R., T.P., T.Ž. and P.Š.; resources, L.D.R. and T.P.; data curation, L.D.R.; writing—original draft preparation, L.D.R. and T.P.; visualization, L.D.R., T.P., T.Ž. and P.Š.; supervision, P.Š.; project administration, P.Š.; funding acquisition, P.Š. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the Croatian Science Foundation under the project “Internet of Things: Research and Applications”, UIP-2017-05-4206.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

IoT	Internet of Things
RSSI	Received Signal Strength Indication
SNR	Signal-to-noise ratio
LPWAN	Low Power Wide Area Network
NN	Neural Network
AUC	Area under the ROC Curve
AP	Average Precision
SGD	Stochastic Gradient Descent
Adam	Adaptive Moment Optimization
RMSProp	Root Mean Square Propagation
ReLU	Rectified Linear Unit

## References

- Zorzi, M.; Gluhak, A.; Lange, S.; Bassi, A. From today's INTRANet of things to a future INTERNet of things: A wireless-and mobility-related view. *IEEE Wirel. Commun.* **2010**, *17*, 44–51. [[CrossRef](#)]
- Ud Din, I.; Guizani, M.; Hassan, S.; Kim, B.; Khurram Khan, M.; Atiquzzaman, M.; Ahmed, S.H. The Internet of Things: A Review of Enabled Technologies and Future Challenges. *IEEE Access* **2019**, *7*, 7606–7640. [[CrossRef](#)]
- Perković, T.; Damjanović, S.; Šolić, P.; Patrono, L.; Rodrigues, J.J. Meeting Challenges in IoT: Sensing, Energy Efficiency and the Implementation. In Proceedings of the Fourth International Congress on Information and Communication Technology in Concurrent with ICT Excellence Awards (ICICT 2019), London, UK, 27–28 February 2019.
- Nizetic, S.; Djilali, N.; Papadopoulos, A.; Rodrigues, J.J. Smart technologies for promotion of energy efficiency, utilization of sustainable resources and waste management. *J. Clean. Prod.* **2019**, *231*, 565–591. [[CrossRef](#)]

5. Singh, A.; Kumar, A.; Kumar, A.; Dwivedi, V. Radio Frequency global positioning system for real-time vehicle parking. In Proceedings of the 2016 International Conference on Signal Processing and Communication (ICSC), Noida, India, 26–28 December 2016; pp. 479–483.
6. Perković, T.; Šolić, P.; Zargariasl, H.; Čoko, D.; Rodrigues, J.J. Smart Parking Sensors: State of the Art and Performance Evaluation. *J. Clean. Prod.* **2020**, 121181. [[CrossRef](#)]
7. Paidi, V.; Fleyeh, H.; Håkansson, J.; Nyberg, R.G. Smart parking sensors, technologies and applications for open parking lots: A review. *IET Intell. Transp. Syst.* **2018**, *12*, 735–741. [[CrossRef](#)]
8. Koster, A.; Oliveira, A.; Volpato, O.; Delvequio, V.; Koch, F. Recognition and Recommendation of Parking Places. In *Advances in Artificial Intelligence—IBERAMIA 2014*; Springer International Publishing: Cham, Switzerland, 2014; pp. 675–685.
9. Shoup, D.C. The High Cost of Free Parking. *J. Plan. Educ. Res.* **1997**, *17*, 3–20. [[CrossRef](#)]
10. Qin, J. Process Data Analytics in the Era of Big Data. *AIChE J.* **2014**, *60*, 3092–3100. [[CrossRef](#)]
11. Hashem, I.A.T.; Chang, V.; Anuar, N.B.; Adewole, K.; Yaqoob, I.; Gani, A.; Ahmed, E.; Chiroma, H. The role of big data in smart city. *Int. J. Inf. Manag.* **2016**, *36*, 748–758. [[CrossRef](#)]
12. Zantalis, F.; Koulouras, G.; Karabetsos, S.; Kandris, D. A Review of Machine Learning and IoT in Smart Transportation. *Future Internet* **2019**, *11*, 94. [[CrossRef](#)]
13. Sarkar, D.; Bali, R.; Sharma, T. *Practical Machine Learning with Python: A Problem-Solver's Guide to Building Real-World Intelligent Systems*, 1st ed.; Apress: New York, NY, USA, 2017.
14. Cui, L.; Yang, S.; Chen, F.; Ming, Z.; Lu, N.; Qin, J. A survey on application of machine learning for Internet of Things. *Int. J. Mach. Learn. Cybern.* **2018**, *9*, 1399–1417. [[CrossRef](#)]
15. Khan, M.A.; Kim, J. Toward Developing Efficient Conv-AE-Based Intrusion Detection System Using Heterogeneous Dataset. *Electronics* **2020**, *9*, 1771. [[CrossRef](#)]
16. Tekouabou, S.C.K.; Alaoui, E.A.A.; Cherif, W.; Silkan, H. Improving parking availability prediction in smart cities with IoT and ensemble-based model. *J. King Saud Univ. Comput. Inf. Sci.* **2020**. [[CrossRef](#)]
17. Yin, Y.; Jiang, D. Research and Application on Intelligent Parking Solution Based on Internet of Things. In Proceedings of the 2013 5th International Conference on Intelligent Human-Machine Systems and Cybernetics, Hangzhou, China, 26–27 August 2013; Volume 2, pp. 101–105.
18. Ali, G.; Ali, T.; Irfan, M.; Draz, U.; Sohail, M.; Glowacz, A.; Sulowicz, M.; Mielnik, R.; Faheem, Z.B.; Martis, C. IoT Based Smart Parking System Using Deep Long Short Memory Network. *Electronics* **2020**, *9*, 1696. [[CrossRef](#)]
19. Khalid, M.; Wang, K.; Aslam, N.; Cao, Y.; Ahmad, N.; Khan, M.K. From smart parking towards autonomous valet parking: A survey, challenges and future Works. *J. Netw. Comput. Appl.* **2021**, *175*, 102935. [[CrossRef](#)]
20. Van Gael, J.; Saatci, Y.; Teh, Y.W.; Ghahramani, Z. Beam Sampling for the Infinite Hidden Markov Model. In Proceedings of the International Conference on Machine Learning, Helsinki, Finland, 5–9 July 2008; Volume 25.
21. Mustafa, M.K.; Allen, T.; Appiah, K. A comparative review of dynamic neural networks and hidden Markov model methods for mobile on-device speech recognition. *Neural Comput. Appl.* **2019**, *31*, 891–899. [[CrossRef](#)]
22. Kouemou, G.L. History and Theoretical Basics of Hidden Markov Models. In *Hidden Markov Models*; Dymarski, P., Ed.; IntechOpen: Rijeka, Croatia, 2011; Chapter 1. [[CrossRef](#)]
23. Xiao, J.; Lou, Y.; Frisby, J. How likely am I to find parking?—A practical model-based framework for predicting parking availability. *Transp. Res. Part B Methodol.* **2018**, *112*, 19–39. [[CrossRef](#)]
24. Luleseged, T.S.; Serugendo, G.D.M. Cooperative Multiagent System for Parking Availability Prediction Based on Time Varying Dynamic Markov Chains. *J. Adv. Transp.* **2017**, *2017*, 1760842.
25. Jiang, Y.; Yang, C.; Na, J.; Li, G.; Li, Y.; Zhong, J. A Brief Review of Neural Networks Based Learning and Control and Their Applications for Robots. *Complexity* **2017**, *2017*, 1895897:1–1895897:14. [[CrossRef](#)]
26. Lipton, Z.C. A Critical Review of Recurrent Neural Networks for Sequence Learning. *arXiv* **2015**, arXiv:abs/1506.00019.
27. Vlahogianni, E.I.; Kepaptsoglou, K.L.; Tsetsos, V.; Karlaftis, M.G. A Real-Time Parking Prediction System for Smart Cities. *J. Intell. Transp. Syst.* **2016**, *20*, 192–204. [[CrossRef](#)]
28. Farag, M.; Din, M.; Elshenbary, H. Deep learning versus traditional methods for parking lots occupancy classification. *Indones. J. Electr. Eng. Comput. Sci.* **2020**, *19*, 964. [[CrossRef](#)]
29. Lin, T.; Rivano, H.; Le Mouël, F. A Survey of Smart Parking Solutions. *IEEE Trans. Intell. Transp. Syst.* **2017**, *18*, 3229–3253. [[CrossRef](#)]

30. Pham, T.N.; Tsai, M.; Nguyen, D.B.; Dow, C.; Deng, D. A Cloud-Based Smart-Parking System Based on Internet-of-Things Technologies. *IEEE Access* **2015**, *3*, 1581–1591. [[CrossRef](#)]
31. Jung, H.G.; Cho, Y.H.; Yoon, P.J.; Kim, J. Scanning Laser Radar-Based Target Position Designation for Parking Aid System. *IEEE Trans. Intell. Transp. Syst.* **2008**, *9*, 406–424. [[CrossRef](#)]
32. Al-Turjman, F.; Malekloo, A. Smart Parking in IoT-enabled Cities: A Survey. *Sustain. Cities Soc.* **2019**, *49*, 101608. [[CrossRef](#)]
33. Al Taweel, Z.; Challagundla, L.; Pagan, A.S. Abuzneid, A. Smart Parking for Disabled Parking Improvement Using RFID and Database Authentication. In Proceedings of the 2020 IEEE 6th World Forum on Internet of Things (WF-IoT), New Orleans, LA, USA, 2–16 June 2020; pp. 1–5. [[CrossRef](#)]
34. Solic, P.; Colella, R.; Catarinucci, L.; Perkovic, T.; Patrono, L. Proof of Presence: Novel Vehicle Detection System. *IEEE Wirel. Commun.* **2019**, *26*, 44–49. [[CrossRef](#)]
35. Šolić, P.; Kapetanović, A.L.; Županović, T.; Kovačević, I.; Perković, T.; Popovski, P. IoT Wallet: Machine Learning-based Sensor Portfolio Application. In Proceedings of the 2020 5th International Conference on Smart and Sustainable Technologies (SpliTech), Split, Croatia, 23–26 September 2020; pp. 1–5. [[CrossRef](#)]
36. Đujić Rodić, L.; Županović, T.; Perković, T.; Šolić, P.; Rodrigues, J.J.P.C. Machine Learning and Soil Humidity Sensing: Signal Strength Approach. *arXiv* **2020**, arXiv:2011.08273.
37. Barriga, J.; Sulca, J.; León, J.; Ulloa, A.; Portero, D.; Andrade, R.; Yoo, S. Smart Parking: A Literature Review from the Technological Perspective. *Appl. Sci.* **2019**, *9*, 4569. [[CrossRef](#)]
38. Grodi, R.; Rawat, D.B.; Rios-Gutierrez, F. Smart parking: Parking occupancy monitoring and visualization system for smart cities. In Proceedings of the SoutheastCon 2016, Norfolk, VA, USA, 30 March–3 April 2016; pp. 1–5. [[CrossRef](#)]
39. Jones, M.; Khan, A.; Kulkarni, P.; Carnelli, P.E.; Sooriyabandara, M. ParkUs 2.0: Automated Cruise Detection for Parking Availability Inference. In Proceedings of the 14th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services, Melbourne, Australia, 7–10 November 2017; ACM: New York, NY, USA, 2017; pp. 242–251. [[CrossRef](#)]
40. Hiesmair, M.; Hummel, K.A. Empowering road vehicles to learn parking situations based on optical sensor measurements. In Proceedings of the Seventh International Conference on the Internet of Things, IOT 2017, Linz, Austria, 22–25 October 2017; ACM: New York, NY, USA, 2017; pp. 39:1–39:2. [[CrossRef](#)]
41. Tatulea, P.; Calin, F.; Brad, R.; Brâncovean, L.; Greavu, M. An Image Feature-Based Method for Parking Lot Occupancy. *Future Internet* **2019**, *11*, 169. [[CrossRef](#)]
42. Camero, A.; Toutouh, J.; Stolfi, D.H.; Alba, E. Evolutionary Deep Learning for Car Park Occupancy Prediction in Smart Cities. In Proceedings of the 12th International Conference on Learning and Intelligent Optimization, LION 12, Kalamata, Greece, 10–15 June 2018; Revised Selected Papers, Lecture Notes in Computer Science; Springer: Berlin, Germany, 2018; Volume 11353, pp. 386–401. [[CrossRef](#)]
43. Zheng, Y.; Rajasegarar, S.; Leckie, C. Parking availability prediction for sensor-enabled car parks in smart cities. In Proceedings of the Tenth IEEE International Conference on Intelligent Sensors, Sensor Networks and Information Processing, ISSNIP 2015, Singapore, 7–9 April 2015; pp. 1–6. [[CrossRef](#)]
44. Klappenecker, A.; Lee, H.; Welch, J.L. Finding available parking spaces made easy. *Ad Hoc Netw.* **2014**, *12*, 243–249. [[CrossRef](#)]
45. Ji, Y.; Tang, D.; Blythe, P.; Guo, W.; Wang, W. Short-term forecasting of available parking space using wavelet neural network model. *IET Intell. Transp. Syst.* **2015**, *9*, 202–209. [[CrossRef](#)]
46. Mor, B.; Garhwal, S.; Kumar, A. A Systematic Review of Hidden Markov Models and Their Applications. *Arch. Comput. Methods Eng.* **2020**. [[CrossRef](#)]
47. Mitchell, T.M. *Machine Learning*; McGraw-Hill: New York, NY, USA, 1997.
48. Turing, A.M. Computing Machinery and Intelligence. *Mind* **1950**, *59*, 433–460. [[CrossRef](#)]
49. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016.
50. Garca, S.; Luengo, J.; Herrera, F. *Data Preprocessing in Data Mining*; Springer Publishing Company, Incorporated: Berlin, Germany, 2014.
51. Mohammed, M.; Khan, M.; Bashier, E. *Machine Learning: Algorithms and Applications*; Taylor & Francis: London, UK, 2016.
52. Ravindran, B. Chapter 23—Relativized hierarchical decomposition of Markov decision processes. In *Decision Making*; Progress in Brain Research; Elsevier: London, UK, 2013; Volume 202, pp. 465–488. [[CrossRef](#)]



53. Zhu, X.X.; Tuia, D.; Mou, L.; Xia, G.; Zhang, L.; Xu, F.; Fraundorfer, F. Deep Learning in Remote Sensing: A Comprehensive Review and List of Resources. *IEEE Geosci. Remote Sens. Mag.* **2017**, *5*, 8–36. [[CrossRef](#)]
54. Bhattacharya, S.; Somayaji, S.; Gadekallu, T.; Maddikunta, M.A.P. A Review on Deep Learning for Future Smart Cities. *Internet Technol. Lett.* **2020**, e187. [[CrossRef](#)]
55. Yamin Siddiqui, S.; Adnan Khan, M.; Abbas, S.; Khan, F. Smart occupancy detection for road traffic parking using deep extreme learning machine. *J. King Saud Univ. Comput. Inf. Sci.* **2020**. [[CrossRef](#)]
56. Provoost, J.C.; Kamilaris, A.; Wismans, L.J.; van der Drift, S.J.; van Keulen, M. Predicting parking occupancy via machine learning in the web of things. *Internet Things* **2020**, *12*, 100301. [[CrossRef](#)]
57. Yang, S.; Ma, W.; Pi, X.; Qian, S. A deep learning approach to real-time parking occupancy prediction in transportation networks incorporating multiple spatio-temporal data sources. *Transp. Res. Part Emerg. Technol.* **2019**, *107*, 248–265. [[CrossRef](#)]
58. Fink, G.A. *Markov Models for Pattern Recognition—From Theory to Applications*; Advances in Computer Vision and Pattern Recognition; Springer: Berlin, Germany, 2014. [[CrossRef](#)]
59. Yoon, B.J. Hidden Markov Models and their Applications in Biological Sequence Analysis. *Curr. Genom.* **2009**, *10*, 402–415. [[CrossRef](#)]
60. Raman, N.; Maybank, S.J. Activity recognition using a supervised non-parametric hierarchical HMM. *Neurocomputing* **2016**, *199*, 163–177. [[CrossRef](#)]
61. Liang, W.; Zhang, Y.; Tan, J.; Li, Y. A Novel Approach to ECG Classification Based upon Two-Layered HMMs in Body Sensor Networks. *Sensors* **2014**, *14*, 5994–6011. [[CrossRef](#)] [[PubMed](#)]
62. Coast, D.A.; Stern, R.M.; Cano, G.G.; Briller, S.A. An approach to cardiac arrhythmia analysis using hidden Markov models. *IEEE Trans. Biomed. Eng.* **1990**, *37*, 826–836. [[CrossRef](#)] [[PubMed](#)]
63. Rabiner, L.; Juang, B. An introduction to hidden Markov models. *IEEE ASSP Mag.* **1986**, *3*, 4–16. [[CrossRef](#)]
64. Rabiner, L.R. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE* **1989**, *77*, 257–286. [[CrossRef](#)]
65. Zucchini, W.; Macdonald, I. *Hidden Markov Models for Time Series: An Introduction Using R*; Chapman and Hall/CRC: Boca Raton, FL, USA, 2009. [[CrossRef](#)]
66. Xi, X. *Further Applications of Higher-Order Markov Chains and Developments in Regime-Switching Models*; Ph.D. Thesis, University of Western Ontario, London, ON, Canada, 2012.
67. Långkvist, M.; Karlsson, L.; Loutfi, A. A review of unsupervised feature learning and deep learning for time-series modeling. *Pattern Recognit. Lett.* **2014**, *42*, 11–24. [[CrossRef](#)]
68. Ng, A.Y.; Jordan, M.I. On Discriminative vs. Generative Classifiers: A comparison of logistic regression and naive Bayes. In Proceedings of the Advances in Neural Information Processing Systems 14 Neural Information Processing Systems: Natural and Synthetic, NIPS 2001, Vancouver, BC, Canada, 3–8 December 2001; Dietterich, T.G., Becker, S., Ghahramani, Z., Eds.; MIT Press: Cambridge, MA, USA, 2001; pp. 841–848.
69. Joo, R.; Bertrand, S.; Tam, J.; Fablet, R. Hidden Markov Models: The Best Models for Forager Movements? *PLoS ONE* **2013**, *8*, 1–12. [[CrossRef](#)]
70. Zhang, G.P. Neural networks for classification: A survey. *IEEE Trans. Syst. Man, Cybern. Part Appl. Rev.* **2000**, *30*, 451–462. [[CrossRef](#)]
71. Krenker, A.; Bešter, J.; Kos, A. Introduction to the Artificial Neural Networks. In *Artificial Neural Networks*; Suzuki, K., Ed.; IntechOpen: Rijeka, Croatia, 2011; Chapter 1. [[CrossRef](#)]
72. Abiodun, O.I.; Kiru, M.U.; Jantan, A.; Omolara, A.E.; Dada, K.V.; Umar, A.M.; Linus, O.U.; Arshad, H.; Kazaure, A.A.; Gana, U. Comprehensive Review of Artificial Neural Network Applications to Pattern Recognition. *IEEE Access* **2019**, *7*, 158820–158846. [[CrossRef](#)]
73. Rawat, W.; Wang, Z. Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review. *Neural Comput.* **2017**, *29*, 2352–2449. [[CrossRef](#)]
74. Goldberg, Y. *Neural Network Methods for Natural Language Processing*; Synthesis Lectures on Human Language Technologies; Morgan & Claypool Publishers: San Rafael, CA, USA, 2017. [[CrossRef](#)]
75. Zhou, Y.T.; Chellappa, R. *Artificial Neural Networks for Computer Vision*/Yi-Tong Zhou, Rama Chellappa; Springer: New York, NY, USA, 1992; 170p.
76. Hewamalage, H.; Bergmeir, C.; Bandara, K. Recurrent Neural Networks for Time Series Forecasting: Current Status and Future Directions. *arXiv* **2019**, arXiv:1909.00590.
77. Errouso, H.; Malhéné, N.; Benhadou, S.; Medromi, H. Predicting car park availability for a better delivery bay management. *Procedia Comput. Sci.* **2020**, *170*, 203–210. [[CrossRef](#)]

78. Hayou, S.; Doucet, A.; Rousseau, J. On the Impact of the Activation function on Deep Neural Networks Training. In Proceedings of the 36th International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019; PMLR, Proceedings of Machine Learning Research, 2019; Volume 97, pp. 2672–2680.
79. Dongare, A.D.; Kharde, R.R.; Kachare, A.D. Introduction to Artificial Neural Network. *Int. J. Eng. Innov. Technol.* **2012**, *2*, 189–194.
80. Chicco, D.; Jurman, G. The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genom.* **2020**, *21*. [[CrossRef](#)]
81. Larsen, J.R.; Martin, M.R.; Martin, J.D.; Kuhn, P.; Hicks, J.B. Modeling the Onset of Symptoms of COVID-19. *Front. Public Health* **2020**, *8*, 473. [[CrossRef](#)]
82. Martin-Abadal, M.; Ruiz-Frau, A.; Hinz, H.; Gonzalez-Cid, Y. Jellytoring: Real-Time Jellyfish Monitoring Based on Deep Learning Object Detection. *Sensors* **2020**, *20*, 1708. [[CrossRef](#)] [[PubMed](#)]
83. Johnson, J.; Khoshgoftaar, T. Survey on deep learning with class imbalance. *J. Big Data* **2019**, *6*, 27. [[CrossRef](#)]
84. Zeng, X.; Martinez, T. Distribution-Balanced Stratified Cross-Validation for Accuracy Estimation. *J. Exp. Theor. Artif. Intell.* **2001**, *12*. [[CrossRef](#)]

**Publisher’s Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).

## **APPENDIX F**

**Paper Title:**

Machine Learning and Soil Humidity Sensing: Signal Strength Approach

**Abstract:**

The Internet of Things vision of ubiquitous and pervasive computing gives rise to future smart irrigation systems comprising the physical and digital worlds. A smart irrigation ecosystem combined with Machine Learning can provide solutions that successfully solve the soil humidity sensing task in order to ensure optimal water usage. Existing solutions are based on data received from the power hungry/expensive sensors that are transmitting the sensed data over the wireless channel. Over time, the systems become difficult to maintain, especially in remote areas due to the battery replacement issues with a large number of devices. Therefore, a novel solution must provide an alternative, cost- and energy-effective device that has unique advantage over the existing solutions. This work explores the concept of a novel, low-power, LoRa-based, cost-effective system that achieves humidity sensing using Deep Learning techniques that can be employed to sense soil humidity with high accuracy simply by measuring the signal strength of the given underground beacon device.

**Cite as:**

Lea Dujić Rodić, Tomislav Županović, Toni Perković, Petar Šolić, and Joel J. P. C. Rodrigues. 2021. Machine Learning and Soil Humidity Sensing: Signal Strength Approach. *ACM Trans. Internet Technol.* 22, 2, Article 39 (May 2022), 21 pages.

**DOI:**

<https://doi.org/10.1145/3418207>



# Curriculum Vitae

## Lea Dujić Rodić

Lea Dujić Rodić is a dedicated PhD student pursuing doctoral studies in electrical engineering and information technology at Faculty of Electrical Engineering, Mechanical Engineering, and Naval Architecture, University of Split. With a strong research background, she has made significant contributions as a researcher within the Croatian Science Foundation project: "Internet of Things: Research and Applications," conducted at FESB.

Lea Dujić Rodić, a native of Zadar, Croatia, completed her primary and high school education in her hometown. She pursued mathematics and achieved a Master's degree in Mathematics and Computer Science with distinction from the Faculty of Science at the University of Split, which she obtained in 2010.

Since April 2015, she has been employed at the Faculty of Electrical Engineering, Mechanical Engineering, and Naval Architecture in Split, where she aspired to deepen her understanding of computer science. In pursuit of this goal, she enrolled in the Postgraduate Study of Electrical Engineering and Information Technology at FESB in 2017. As a Data Scientist, Lea Dujić Rodić is actively involved in the project "Internet of Things: Research and Applications" since 2019. She specializes in data analysis and processing of sensor data collected through the LoRaWAN network protocol. She has extensive expertise in developing effective Machine Learning algorithms using the Python programming language, catering to a wide range of applications in the field of IoT and smart environments.

Lea's research interests primarily revolve around IoT applications, Machine Learning approaches, and Deep Learning techniques for extracting high-level information and resolving challenges presented by IoT data. Her research interests also extend to Human-Computer Interaction (HCI) and technologies for young children's education.

Throughout her scientific career, Lea's work has resulted in the development of algorithms and device prototypes for real-life applications in Smart Parking/Smart City and Smart Agriculture solutions. She is the author of over 15 cited scientific articles published in prestigious international journals and has presented her research at renowned scientific conferences.

In addition to her research achievements, Lea has gained valuable experience in teamwork through collaborative projects. She excels in preparing scientific achievement reports, financial reports, and project work plans while consistently meeting deadlines. Lea possesses excellent proficiency in spoken and written English, which she has honed over the years.

# Životopis

## Lea Dujić Rodić

Lea Dujić Rodić predana je doktorandica na doktorskom studiju elektrotehnike i informacijskih tehnologija na Fakultetu elektrotehnike, strojarstva i brodogradnje Sveučilišta u Splitu. Svojim istraživanjem dala je značajan znanstveni doprinos u okviru projekta Hrvatske zaklade za znanost: „Internet stvari: istraživanje i primjena“, koji se provodi na FESB-u.

Lea Dujić Rodić, rođena je u Zadru, Hrvatska, gdje je završila osnovnu i srednju školu. Zbog izraženog afiniteta prema matematici upisuje studij matematike na Prirodoslovno-matematičkom fakultetu Sveučilišta u Splitu 2005. godine, gdje 2008. stječe diplomu prvostupnice matematike. Na navedenom studiju je stekla diplomu Magistra inženjerske matematike računalnog smjera 2010. godine.

Od travnja 2015. zaposlena je na Fakultetu elektrotehnike, strojarstva i brodogradnje u Splitu, gdje je težila produbiti svoje razumijevanje računarstva. Slijedeći taj cilj, 2017. godine upisuje Poslijediplomski studij elektrotehnike i informacijskih tehnologija na FESB-u. Kao Data Scientist, Lea Dujić Rodić aktivno uključena u projekt „Internet of Things: Research and Applications“ od 2019. godine. Specijalizirana je za analizu podataka i obradu senzorskih podataka prikupljenih sa LoRaWAN senzorskih uređaja. Lea posjeduje stručno znanje u razvoju učinkovitih algoritama strojnog učenja pomoću programskog jezika Python, koji obuhvaćaju širok raspon primjena u području IoT-a i pametnih okruženja.

Lein istraživački interes prvenstveno je fokusiran na IoT aplikacije, tehnike strojnog učenja za ekstrakciju informacija iz okoline te rješavanje izazova koje navadni podaci predstavljaju unutar IoT paradigme. Njezin istraživački interes također uključuje interakciju čovjeka i računala (HCI) i edukacijske tehnologije.

Tijekom njezine znanstvene karijere, Lein znanstveno-istraživački rad rezultirao je razvojem algoritama i prototipova uređaja za stvarne primjene u Smart Parking/Smart City i Smart Agriculture rješenjima. Autorica je više od 15 citiranih znanstvenih članaka objavljenih u prestižnim međunarodnim časopisima, a svoja je istraživanja prezentirala na renomiranim znanstvenim skupovima.

Osim svojih istraživačkih postignuća, Lea je stekla dragocjeno iskustvo u timskom radu kroz suradničke projekte. Izvrсна je u pripremi izvješća o znanstvenim postignućima, financijskih izvješća i planova rada na projektima uz dosljedno poštivanje rokova. Lea posjeduje izvrsno znanje govornog i pisanog engleskog jezika, koje je usavršavala tijekom godina.