SVEUČILIŠTE U SPLITU
FAKULTET ELEKTROTEHNIKE, STROJARSTVA I BRODOGRADNJE

POSLIJEDIPLOMSKI DOKTORSKI STUDIJ ELEKTROTEHNIKE I INFORMACIJSKE TEHNOLOGIJE

KVALIFIKACIJSKI ISPIT

Big Data Challenges and Trade-offs in Energy Efficient Internet of Things systems

Jelena Čulić Gambiroža

Split, ožujak, 2021.

Contents

1. Introduction

The Internet of Things (IoT) concept has grown in the last few years and its topics are highly researched. It is predicted that there will be up to 75 billion devices connected in IoT by 2025 [1]. Basic IoT system consists of sensors collecting data, gateways as midpoint devices and cloud where data is stored, streamed, analyzed, and presented. Most of IoT systems have numerous sensors distributed within some area, each of which has its own battery with limited lifetime, which represents one of the main bottlenecks of this technology. Consequently, IoT sensor tasks have to be carefully planned as they can not handle large energy consumption requirements. Numerous sensors collect data in short time intervals, which will result with up to 79.4ZB of data generated by IoT in 2025 [2]. Finally, all the collected data is sent towards cloud where it must be processed and stored in large data centers. Furthermore, it is expected that 175ZB of data will be stored by 2025 [3]. However, more data is not always better than less data as collecting redundant data simply takes storage capacity without providing new information.

As the volume of generated data increases, storing and processing it becomes significantly challenging. This is already recognized as the Big Data concept, commonly described with five Vs which stands for volume, velocity, variety, veracity and value [4]. *Volume* represents the amount of data, while *velocity* is the speed at which new data is generated and the speed at which data moves around. *Variety* stands for different types of data. *Veracity* is the measure of data uncertainty, while *value* is the information obtained from the collected data.

Today, IoT requirements increase data volume and velocity, as well as variety. Finding the optimal balance between the volume reduction and the information loss (value) requires the utilization of data variety and thus reducing data velocity. The expected growth in the number of IoT data sources gives rise to network-edge computing [5]. Edge-mining stands for data processing on battery-powered devices placed at the edges of an IoT network. Such a solution would achieve reduction of data volume at the network edge and thus reduce energy consumption, bandwidth, as well as storage capacity and processing power at the cloud backend systems. The focus of our research is data velocity and volume reduction, while taking into a consideration data variety and preserving its value in all parts of an IoT system.

Processing data on the IoT edge can be achieved in several ways; data collected by sensors can be compressed, aggregated and/or correlated on IoT gateways and only then transferred to Cloud. To move even closer to the edge, data can be filtered on sensors themselves in order to forward only relevant information to an IoT gateway [5]. However, although the amount of data is decreased, sensors are still constantly turned on and consume energy.

The paper is divided into 5 sections. Section I introduces the paper and provides motivation. It is followed by an IoT system design and overview of existing solutions in Section II. Section III provides structural approaches for combining IoT and Big data concepts more efficiently in terms of reducing data volume and energy consumption. Sections IV overviews open challenges, while Section V summarizes current research. The last, Section VI, concludes this document.

2. IoT System overview

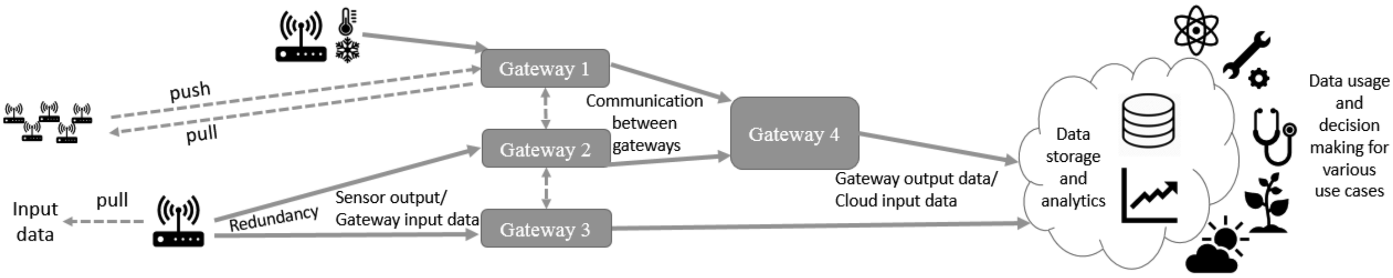

Figure 1 IoT system design

IoT systems consist of three main components; *sensors* that collect data from the environment and send it through *gateway* towards *cloud*, where data is analyzed and used for decision making. Each component receives the *input data*, performs internal data processing, and forwards *output data* (Figure 1). *Input* and *output data* can be received/forwarded using either *push* or *pull* mechanisms, or as their combination. The initiator for *push* mechanism is the southbound component, e.g., a sensor pushes the data towards a gateway at any moment, which requires the gateway to asynchronously wait for data transfer. For *pull* mechanism the initiator is the northbound component, e.g., a gateway requests data from sensor, which requires the sensor to be always on and waiting for data transfer.

2.1. Sensors

Sensors are endpoint devices that read physical measurements from the environment. As endpoint devices, sensors are commonly battery powered and have low data processing power. Due to growth of IoT, their application is increasing. Due to drastically reduced production costs, sensors are usually collecting more than one measurement type, so data collected by a single sensor can contain heterogenous values.

- *input data* – a measurement collected by the sensor
- *output data* – data processed on sensor and transmitted towards gateway

2.1.1. Energy Efficient Approaches

While collecting data from the environment, sensors are dealing with large amount of data in short time intervals and consequently, consume a lot of battery power, but with a lot of similar values collected. Sensors should implement algorithm that dynamically adapts reading interval based on previously collected values as well as filter collected data, classify them and self-control data transmission according to their trends.

- **Input data**

In [6] Mastelic et al. introduce a dynamic monitoring frequency algorithm for collecting monitoring data from ultrascale systems. The algorithm deterministically reduces data velocity by dynamically self-adapting the monitoring frequency to the volatility of collected data. It also reduces data volume because it reads data less often but still keeps the same data value as the equivalent static monitoring frequency. The algorithm requires human (administrator) input of maximum and minimum periods as well as factor for calculating window size.

Trihinas et al. [7] introduce AdaM, a lightweight adaptive monitoring framework for battery-powered IoT devices with limitations in processing capabilities. Framework consists of two algorithms, one for adaptive sampling and other for adaptive filtering. In this part of the system, adaptive sampling, where algorithm dynamically adapts monitoring intensity based on the current evolution and variability of the metric stream with minor processor usage, is used. Results show that it reduces volume and energy consumption while preserving relatively high accuracy.

In [8] Lujić et al. present a three layer (gathering, edge and cloud) architecture model for data storage management on the Edge. This includes an adaptive algorithm that is dynamically finding a trade-off between data accuracy and data volume, focusing on time series data. By using the proposed approach it is possible to reduce the amount of stored data by an average 73.02% and 80.27% in each cycle for the two datasets respectively, while satisfying demands for forecast accuracy and thereby showing potential for saving limited storage space.

Jia et al. in [35] propose a new, low-power, automatic, accurate, and wireless ammonia monitoring approach that uses metal oxide sensors. This approach does not wait for equilibrium as this consumes significant amount of energy, rather it tries to predict the resistance at equilibrium using the sensor's transient measurements in the short heating window using LSTM neural networks. Proposed model accurately predicts the equilibrium state resistance value with an average error rate of 0.12%.

Arendt et al. [41] present a model-predictive communication framework based on historical data analysis. Data collected in three previous days is used to predict fourth day values. Two algorithms, autoregression based ARIMA and a neural network with LSTM cells, are compared, in order to reduce communication effort. Results show a significant potential to reduce communication effort per day of at least 60% ut to 95%, depending on the required accuracy, significantly contributing to the achievement of mMTC performance targets.

In [42], Cecchinel et al. have investigated whether there is significant battery lifetime gain when using adaptive sampling and sending periods. Self-adaptive approach using machine learning and deep-sleep is used to provide an optimal configuration extending the battery lifetime of a sensor platform. Prediction model based on historical data is implemented on a middle layer, and it is used for the generation of the optimal configuration. Gain in the battery lifetime is compared to solution that uses fixed periods (with and without deep-sleep). It is shown that adaptive periods successfully lower the battery discharge. In experimental validation, authors have successfully scaled up the battery lifetime of a temperature sensor from a monthly to a yearly basis.

- **Output data**

The second part AdaM [7] algorithm, adaptive filtering, is useful for preparing sensor output. After sampling algorithm selects data to read, filtering algorithm analyses the dataset and decides for every input, is it relevant and should it be passed to the next stage.

In [9] Zordan et al. presents a lossy compression scenario which uses machine learning techniques for signal classification to improve energy efficiency while tolerating some distortion. The algorithm first collects timeseries data and extracts features. Next step is feature normalization and selection of relevant ones. Last step is signal classification. Experiments show that small number of features (< 20) achieve classification performance over 97%.

In [10] Baharudin et al. propose low-energy algorithm for sensor data transmission from sensor nodes where the sensors can self-control data transmission according to the trends of data. It uses Adaptive Duty Cycle for data transmission adjustment frequency and Compressive Sensing (CS) for data compression. Simulation results show that collective transmission with CS-based data compression significantly reduces transmission energy.

In [42], Cecchinel et al. have investigated whether there is significant battery lifetime gain when using adaptive sampling and sending periods. While adaptive sampling is optimization in input sensor data, adaptive sending periods optimize output of a sensor. As already written in previous section, experimental validation shows that this combination of adaptive parameters extends battery lifetime of a temperature sensor from a monthly to a yearly basis, while ensuring a proper level of data quality and freshness.

2.2. Gateways

Gateways are midpoint devices placed between sensors and a data analytics backend in the cloud [11]. A gateway allows sensors to communicate over shorter distances and abstracts the underlying communication layer such as LoRa [12] on its southbound, while on its northbound it forwards the data towards the cloud. Before forwarding data, it can perform additional processing for optimization or on-site analytics and decision making.

- *input data* – represents data coming from sensors. The data is received over a communication channel that sensors support, such as LoRa or even some proprietary solution. Consequently, the data format can also be arbitrary, which requires a gateway to speak the same language as the sensors. While the data is commonly pushed towards the gateway as previously explained, gateways can still support some type of push back communications towards the sensors as they are required to manage the sensor network, e.g., sending a configuration to sensors, load balancing different communication channels, etc.
- *output data* – data received from sensors that is processed by the gateway and forwarded to the cloud for further analysis and decision making. With the rise of Fog and Edge computing, more functionalities are pushed towards gateways [13][14]. Due to their low computational power and storage capacity compared to the cloud, the data is still pushed to cloud. This is commonly done over the Internet, while communication channels can vary from 3G/4G and tomorrow 5G, to WiFi or even cable connection if the gateway is located in (sub)urban area.

2.2.1. Energy Efficient Approaches

Data from sensors are coming to the gateway. Sensors may decide when to push, or gateway may send pull request to one of sensors in the group, or to all of them, depending on their position, battery level or Internet connection. When received, gateway should filter, compress and corelate data in order to forward only relevant values towards the Cloud.

- **Input data**

Mushunuri et al. in [14] incorporate optimization libraries within the Robot Operating System deployed on a robotic sensor – actuators. To overcome the limitations of IoT devices in terms of computation power, Fog computing is used. One robot acts as a master node with others as client nodes. Whenever there is data available for computation, the client robot shares its computation load in order to extend its battery lifetime. The Edge client robot having data for computation sends its entire data to Fog server robot that divides computation among peer robots depending on their communication path loss and power availability.

Samie et al. in [15] propose a technique for managing computation offloading in a local IoT network under bandwidth constraints. The gateway gives each device its minimum bandwidth demand. Then it calculates the battery life of each device under this configuration and remaining bandwidth. The remaining bandwidth must be allocated to the devices, while prioritizing the devices with lower battery life. Results show more than 40% improvement in utilization of gateway's bandwidth as well as up to 1.5 hour improvement in battery life.

Liyanage et al. [16] proposes a proactive IoT gateway service scheduling scheme in the opportunistic Internet sharing environment. The scheme is to be used as a background service of a device to continuously retrieve the information of available gateways and to schedule the connection among the available gateways. The scheduling scheme reduces the re-connection between the collaborative devices and minimizes the unnecessary energy consumption derived from re-connection processes.

In [17] Natarajan et al. propose an end-to-end system prototype for power-efficiently compression of continuous bio-signals in sensor network. The gateway dynamically tunes compression parameters to adapt to signal changes during continuous monitoring and transmits them to the sensor. Gateway dynamically adapts and tunes compression parameters based on current sparsity levels. Real-time parameter tuning is achieved by periodical transmission of raw measurements from the node to the gateway. This results with ~100X faster solution with energy consumption less than ~3% of standard encoder.

In [18] Wang et al. propose sleep scheduling and wake up protocol. In this solution gateway calculates sleep interval for sensor in order to save their battery and achieve energy efficiency.

Khandel et al. in [19] aim to reduce the communication overhead and propose a method that is able to determine which sensors should send their data to the central node and which one should drop data. Some sensors have high data correlation or they may have data that are not essential at the central node for current operation, the should drop their data. They designed a controller that selects a subset of sensors to send data to the cloud, while keeping the accuracy at an acceptable level. Considering the dynamics of the data collected at the sensors, Advantage Actor-Critic based reinforcement learning (RL) scheme is implemented to train the controller. They also demonstrated how the parameters of the RL reward function can be tuned to make an appropriate tradeoff between the accuracy and communication overhead.

- **Output data**

In [20] NECtar agent, a solution for IoT network-edge data reduction is presented by Papageorgiou et al. This solution presents three main elements; streamification, one-click data handler instantiation and IoT-specific analysis. Streamification includes new data reduction algorithms that work upon data streams and take decisions per incoming data item. One-click data handler instantiation enables network-edge devices to select one of many possible algorithms and instantiate a handler that enforces the respective data reduction logic. Finally, IoT-specific analysis stands for one of the first network-edge data reduction algorithms evaluated specifically for IoT devices and datasets. NECtar achieves accuracies 76.1% to 93.8% while forwarding 1/3 of data items, without adding significant forwarding delays.

Razafimandimby et al. [21] present Bayesian Inference Approach. Sensors are collecting multiple values, including temperature and humidity, every 30 seconds. Gateways compute the probability of making an inference error in the cloud given the temperature and the humidity, before sending their data. If there is a strong chance that the error magnitude exceeds a predefined threshold, the gateway sends both humidity and temperature, else the gateway sends only the temperature data, and the humidity value will be inferred in the cloud. Evaluation results show that this approach drastically reduces number of transmitted data and the energy consumption, while maintaining an acceptable level of data prediction accuracy.

2.3. Clouds

Data collected by sensors and processed by gateways ends up in the cloud where they are analyzed and archived. Due to distance from sensors, real-time systems may suffer of high latency, but latency decreases with introduction of 5G and high speed Internet [22]. Utilization of machine learning and prediction algorithms enables filling the gap between real-time systems and distant clouds [23]. Finally, distribution of cloud towards Fog and Edge reduces this lag even more [24].

- *input data* – represents data that is received by the cloud services, commonly in a push manner, i.e., gateways push the data towards the cloud.
- *output data* – once received by the cloud services the data is then processed, visualised and stored. Data format on the output depends on its purposes, as well as the use case. Due to high data velocity, more processing is performed using streaming analytics while storing only relevant data. Even visualization is done only after preprocessing the data due to its high volume

2.3.1. Energy Efficient Approaches

After data is collected, it should be stored somewhere. Storing every single data that is collected results in large amount of similar data with great need for storage capacity. Before storing, data should be filtered, compressed and normalized in a smart way so only really relevant data is stored.

- **Input data**

In [25] Soultanopoulos et al. describe the data collection from Bluetooth Low Energy devices to identify and track data in real time and allow processing in the cloud to improve analysis. One of the main characteristics of this solution is smart mode of data transmission including the time interval, on demand-violation and rules. A gateway enables the processing of sensor data before they are transferred to the cloud. The experimental results demonstrate that the system supports real time communication and fast data collection with data transmission in less than 130 ms.

In [26] Ge et al. proposes energy efficient solutions such as putting idle servers in Data Center (DC) to sleep as well as shutting down idle DCs during off-peak hours.

- **Output data**

Meng and Liu [27] introduce the concept of monitoring-as-aservice (MaaS), its main components and requirements. There are three enhanced MaaS capabilities: window-based violation detection which can save 50% to 90% communication cost because of its parameter tuning ability; violation-likelihood based state monitoring which can adjust monitoring intensity based on the probability to detect important events which results with significant gain in monitoring service consolidation; and multitenant state monitoring techniques. Monitoring topology planning technique minimizes monitoring data delivery overhead that leads to improvements in the scalability of the service.

Moon et al. in [28] compare lossy compression algorithms Discrete Cosine Transform (DCT), Fast Walsh-Hadamard Transform (FWHT), Discrete Wavelet Transform (DWT), and Lossy Delta Encoding (LDE) on weather sensor data. Results indicate that DCT and FWHT generate higher compression ratios than others. Regarding information loss, LDE outperforms others. Compression error is much severe in DCT and FWHT while LDE is able to maintain lower error rate than others.

In [29] Eliseev et al. give an overview of modern methods, models and technologies to process Big data in large-scale system such as Map-Reduce. It also shows that Big data processing takes much less time on a cluster, compared to a single computer.

In [38] Kosović et al. estimate value of daily solar radiation using Machine Learning (ML) algorithms. Input to ML algorithm are other collected parameters that are correlated to solar radiation. Using such approach, the amount of data that is being sent through whole IoT network is reduced. Cost is additionally lowered since one expensive sensor is removed from IoT network.

3. Energy Efficient Approaches for Data Velocity and Volume Reduction

In order to build an energy efficient IoT system that is able to tackle Big Data challenges it is important to understand where reduction in data volume and velocity can be made and what are the trade-offs. This Chapter provides overview of the approaches with their applications on specific components of an IoT system and highlights their trade-offs.

3.1. Dynamic monitoring frequency

When reading data in short intervals, change in value is not necessarily detected. Thus, instead of collecting data on a specific interval, data should be collected when some event occurs. While this is possible only when system itself creates event, it is possible to emulate such behavior in systems where monitoring value is sensed from the environment by dynamically changing the monitoring frequency to detect and store only changes. The sensor should use lightweight algorithm [7][8] to collect data only when the significant delta is expected and sleep in the meanwhile. Consequently, if the current delta is lower than significant delta, monitoring interval increases and vice versa. This way, if data is changing slowly, sensor may be in sleep mode for hours instead for minutes. Otherwise, if data is sampled after it is collected, sensor still has to spend energy for data collection and analysis. Despite changing frequency on sensor, gateway still receives large amount of data from multiple sensors in small intervals, so it can also orchestrate data collection on sensor by sending pull requests, e.g. depending on the delta of multiple sensors. Since the gateway is a bigger and usually plugged in, orchestration can lower sensor battery consumption. However, such approach results in centralized architecture with two-way communication, which may increase sensor battery consumption due to constantly being online. Finally, either changing the monitoring frequency on a sensor or a gateway, deltas that occur between two readings is skipped and thus lost forever. This makes this technique hard to evaluate online and use machine learning to improve it as there is no data to evaluate its performance.

3.2. Data filtering

Sensors may filter data and transmit only relevant data, while ignoring the rest. On one hand, they save on transmission as less data is transmitted. On the other hand, sensors still have to collect data and consume energy while doing so. Furthermore, a complex filtering algorithm would require both processing power and as well as more memory. Therefore, filtering is commonly performed on gateways as more powerful device. Either way, filtering requires high level knowledge of collected data to work in optimal manner. Consequently, the entire sensor network has to be managed in a distributed manner, where each gateway or a sensor has to be configured properly. Besides doing filtering on sensors and gateways, it can be done in the cloud as well. Filtering in the cloud results with significant reduction of volume while storing data and velocity when sending data to analysis. Besides doing filtering on sensors and gateways, it can be done in the cloud as well. Since the cloud is running a backend system it has all the knowledge of the data being collected, and thus can perform very efficient data filtering [28]. Filtering in the cloud results with significant reduction of volume while storing data and velocity when sending data to analysis. However, all the received data had to pass through the whole network, which may have produced high energy usage even before reaching the cloud.

3.3. Data consolidation

There is commonly a group of sensors within a certain area. Sensors may or may not communicate with each other. They can also decide when to send data and push them toward a gateway or decision can be made on the gateway, so the sensor has to wait for a pull request. They can agree on transferring bulk of data from only one sensor, or a single sensor that collects different data types can bulk transfer its readings. Problem is that all types must be read at the same interval, excluding dynamic frequency approach. For long intervals, sensors may not have enough memory or transmission might take too long. Using the consolidation scheme for gateways can be more feasible due to larger memory and better connection with the cloud than the one between sensors and gateways is. Gateways can also orchestrate Internet connection sharing [16] or share data computation load among sensors towards saving energy [14] based on bulk transfer principles. In the cloud, consolidation can also take place when storing and partitioning data, e.g., related data can be stored together for quick and easy access.

3.4. Data aggregation

Data aggregation presents data in a summarized form, e.g., storing average values or a trend instead of raw data. Using this approach, sensor device can aggregate data in some predefined interval and send only aggregated values. It may also compare collected value with aggregated one and send both values if they vary significantly. Same procedure can be implemented on gateways; they can also aggregate values from multiple sensors at the same moment in time. Cloud may also implement similar approach and aggregate data to reduce required storage capacity.

3.5. Data correlation

Data correlation refers to a process of combining the data in some manner, either by correlating it with some other data and thus applying the soft(ware) sensor concept [21][30][38], namely correlating different sensor values to obtain additional one instead of measuring it with a hardware sensor, or by correlating the same type of sensor data between different sources. A sensor device can collect one or multiple data types. If it collects multiple values, e.g., temperature and humidity, it can corelate those metrics by some rule and read and/or transmit only one of them, e.g. temperature, while humidity can be inferred on a gateway or a server using some algorithm, e.g., BP [21]. To achieve this, sensors require high-level knowledge of temperature humidity correlation. Dynamic frequency discussed earlier can be defined using this correlation, e.g., by reading temperature values constantly while humidity is read only if the temperature changes. However, this complicates the sensor algorithm and thus consumes more energy on an already battery powered device. Gateway receives large volume and variety of data, so it can correlate data form different sensors which are close to each other, or correlate different sensor types. It can also orchestrate data collection on the sensor side based on correlation by scheduling them for optimal performance, such as the sensor battery level [15]. For instance, a group of sensors collect temperature. If two sensors are close to each other and one collects value of 5°C, other will probably collect the same or very similar value. If difference between two sensor values is less than the significant delta (e.g. 1°C) gateway could learn that those two sensors can be read interchangeably. Similar approach can be taken in the cloud, which like filtering requires transferring all the data into cloud. However, implementing soft sensors in the cloud also keeps the raw data as well, so any other analysis can be performed later on.

3.6. Data compression

Lossless compression on sensors can be achieved in several ways i.e., only changes of values (deltas) can be sent instead of full values. If sensors are collecting data in predefined period, they could exclude timestamp and it can be calculated later from a known period. Furthermore, if sensors send a predefined structure, they can exclude variable names. Instead, gateways could be implemented to read the predefined structure. While all these compressions would save only bytes of transmission data, in radio communication each byte counts. Typically, lossy compression is aimed at saving transmission energy, yet affects the quality of transmitted data over lossy channels. Accordingly, using error correction coding along with compression is required to guarantee both energy efficiency and high-fidelity reconstruction [39]. Similar compressions can be performed on inter-gateway communication and clouds as well, where due to huge amount of data few bytes become gigabytes of data.

3.7. On-site data analytics

On-site data analytics performed on a gateway represents Edge [14] – a concept for collecting and analyzing data on the spot. In such scenarios, data is only forwarded to the cloud for storing or post-analysis, while all the decision making is moved closer to the Edge, namely gateways and sensors. Furthermore, on-site data analytics applied in the cloud would refer to the streaming analytics, i.e., the ability to constantly, in real-time measure and analyze data while moving within the large data streams. Afterwards, data can be stored (in compressed, aggregated, consolidated form, or as raw data) for archive.

4. Open Challenges

There are still many open challenges that have to be taken into consideration when building IoT system for handling Big Data:

- *Remote configuration* – as previously mentioned, there are many sensors and/or gateways distributed in some area and they must be configured, either at deployment or during runtime. This requires standardized approach for remote configuration, e.g. all sensors as well as gateways should be configurable from a remote device, either a server or one of the gateways. Consequently, this required standardized interfaces and communication protocols, as well as reuse of existing solutions in order to achieve industry standards.
- *Data reduction trade-offs* – Reducing data volume and/or velocity comes with trade-offs, namely loss of data, additional processing at the network edge, as well as in the cloud for recreating missing data for analytic purposes or more complex analytic algorithms that can handle missing data. That said, data reductions should be balanced not to create even more energy consumption in backend, or even reduce data value and usage. In order to tackle this challenge new metrics are required, namely those that are able to measure concepts such as data value and data loss, as well as energy savings and trade-offs in such systems.
- *Signal type recognition* – As number of IoT devices is increasing, large number of devices, monitoring different environmental phenomena, are connected to a single IoT network. To reduce energy consumption, sensors may send only timeseries data. Thus, knowledge of data type is linked with sensor generating it only in the cloud. In case of unexpected problems, including network, database, etc., sensors will continue to send data, but knowledge of data type is lost. Machine learning can be used to learn parameters that describe specific signal from chunks of signal, in order to correctly classify data type and restore the network.
- *Online evaluation* – When implementing reduction algorithms that skip or in some way omit data, there is no straightforward approach to evaluate and thus improve those algorithms online. Therefore, use of machine learning (ML) might be limiting. However, creating simulators that are able to virtually mimic the real world use cases would allow writing and testing these algorithms in near-realistic situations. This is possible due to the nature of real world sensor data as it can be modelled as a continues signal sampled in discrete manner.

- *General purpose algorithms* – As IoT equipment evolves, more devices will serve general purpose. However, creating generic data reduction algorithms represents a challenging task due to high data variety as well as variety of use cases. Therefore, the first step towards a solution is creating a variety of common datasets used for evaluating the algorithms. This way, a methodical research can be applied in order to tackle this challenge.

5. Current and future work

This chapter will give an overview of current research, as well as future plans. In order to perform Big data optimization in IoT, the main focus is put on data reduction on the edges of IoT. Reduction on sensor side results with the highest energy savings since data is eliminated before it is even collected. Therefore, two approaches will be resarched; predicting final value from chunk of data and dynamic monitoring frequency approach.

5.1. Predicting value from chunck of data

There are sensors that have to heat up before being able to collect some metric from the environment. Example of such sensor is gas sensor, e.g. MQ-2 gas sensor [40]. In order to collect the actual value representing the gas concentration, a gas sensor needs to heat up. During this preheat time, voltages and currents take time to stabilize to obtain readings of the actual gas concentration. A momentary variation in the current or the voltage during this preheating transition is called a transient, only after which an actual value can be read. However, transients in low cost sensors can take minutes, which consumes significant amount of energy from a battery powered sensors. Therefore, instead of waiting for the sensors to fully preheat, only parts of the transient can be collected. As Jia et al. show in their research [35] energy can be saved if gas values are predicted from a part of the transient using LSTM neural network [31].

Once sensor readings are stabilized, the sensor can periodically read data while continuously being online as shown in Figure 2a. In case the sensor is put to sleep to save energy, it has to preheat after every sleep period and then read the last value of the transient, i.e., the actual value of gas concentration. This scenario is shown in Figure 2b. Approach we propose, where a sensor reads multiple values from the first part of the transient, while the actual value is estimated from that part using machine learning, is depicted in Figure 2c. This way, the sensor will have a longer sleep period, while more values are read in its short online period.

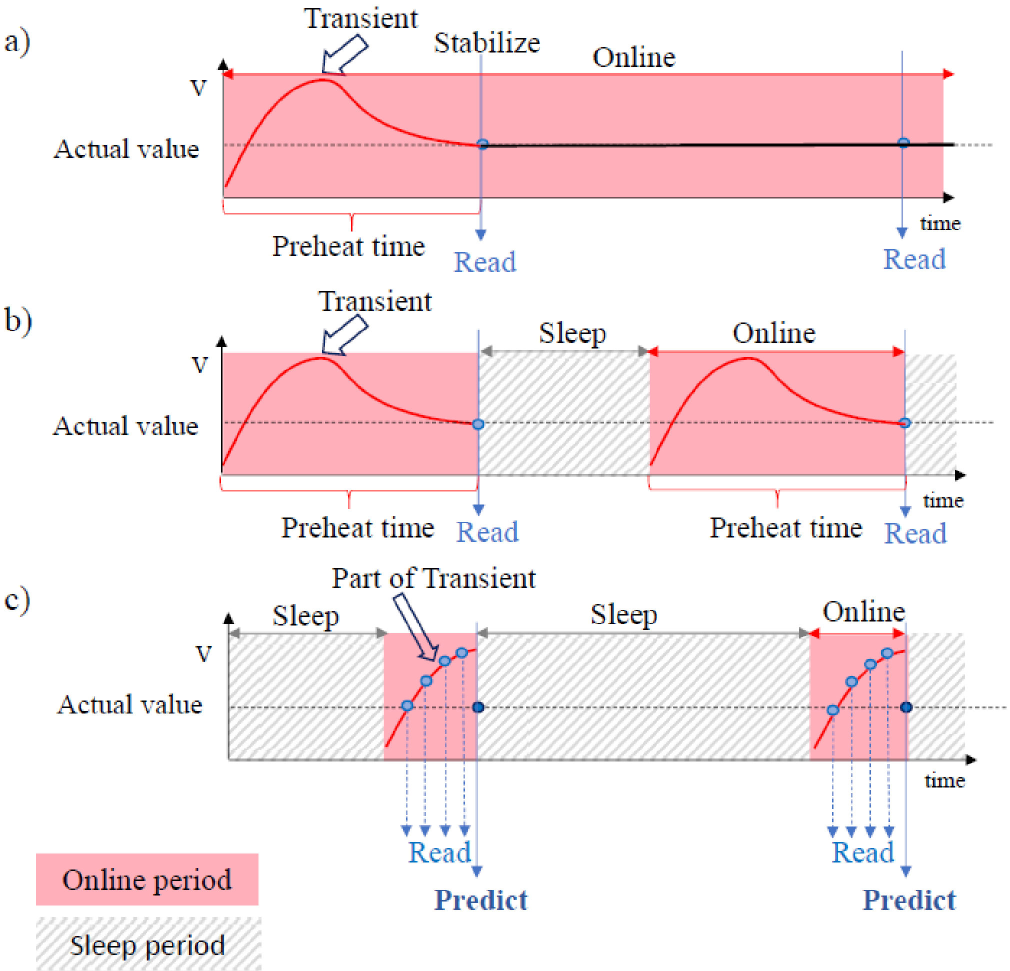

Figure 2 Three scenarios for sensors setup, namely a) always online, b) with full preheat and sleep period and c) with prediction based on a partial preheat and sleep period

5.2. Dynamic monitoring frequency

Dynamic monitoring frequency approach is already described in Chapter 3.1. Future plan is to implement such algorithm in order to predict when expected change will happen. For algorithm implementation it is planned to use machine learning methods as well as statistical methods, since statistical method have provide good results in [6] and [7].

Alternative to dynamic monitoring frequency is data collection with fixed interval. Figure 3 and Figure 4 show one day of temperature data. Figure 3 depicts ideal data collection algorithm (an optimistic baseline) on one day data, where data is collected only when it is changed for 0.5°C. There would be only 37 values collected that day, instead of 1440 actually collected values, with one minute period. As this algorithm collects data dynamically, amount of collected data in known only after data collection. Thus, this solution should be compared to algorithm that collects the same amount data, while collecting with fixed period. Figure 4 depicts values that would be collected by algorithm that collects the same amount of readings with fixed interval of 39 minutes. In this case, minimal collected delta is 0.1°C and maximal delta is 1.3°C, on the first day of dataset.

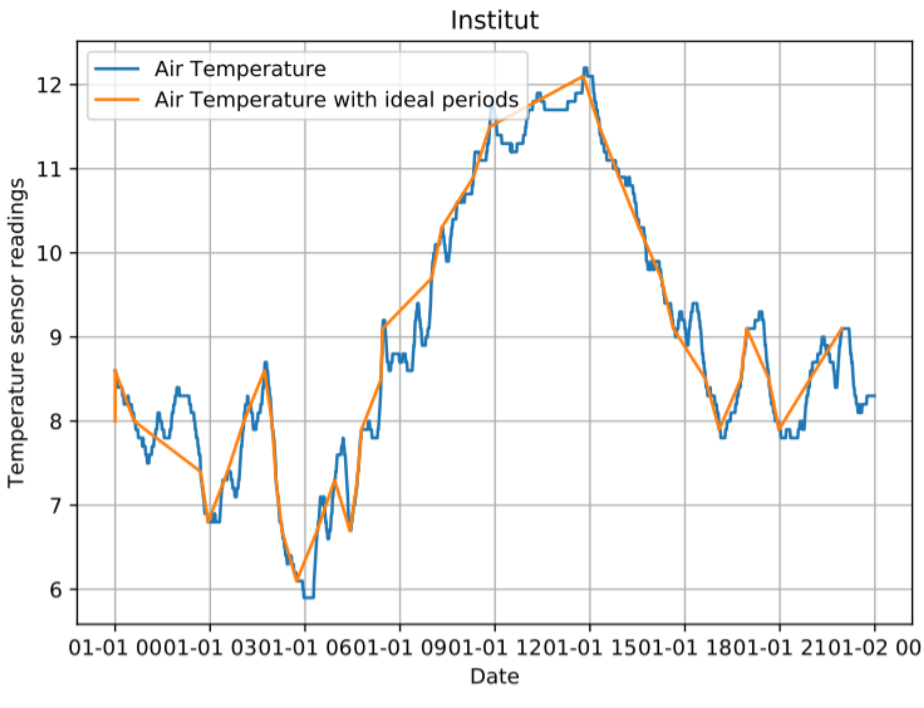

Figure 3 One day data collected with ideal algorithm that collects temperature only when it is changed for 0.5°C

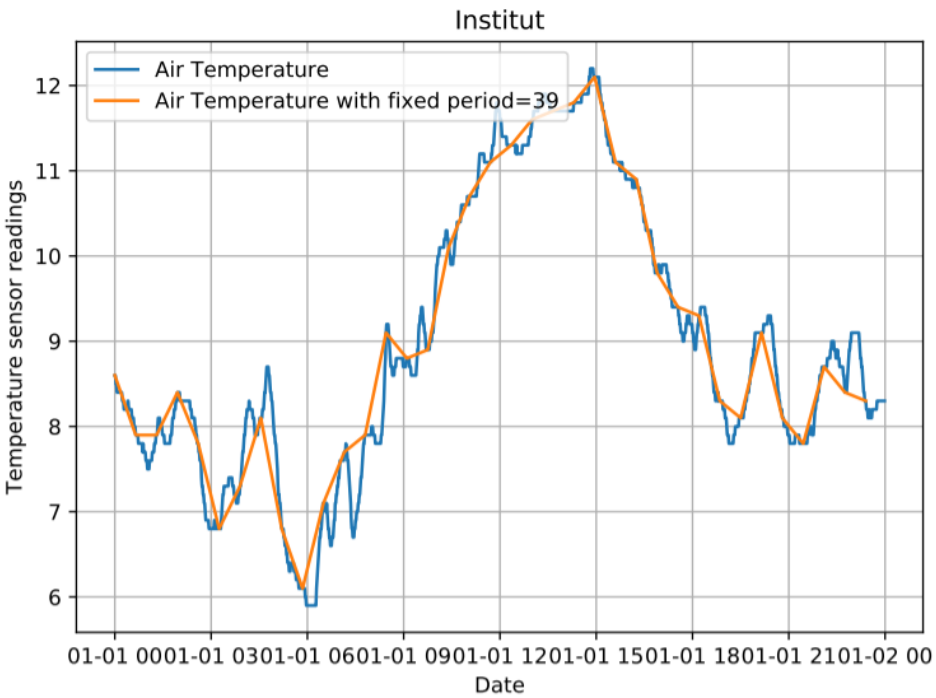

Figure 4 One day data collected with fixed interval = 39 minutes

6. Conclusion

IoT usage is growing in last few years and number of IoT devices is increasing rapidly. Therefore, amount of collected data is increasing as well, which leads to Big data problems. IoT sensors are commonly battery powered so growth of data volume also results with high energy consumption and short-lasting batteries. This paper provides an overview of the existing solutions related to IoT and Big data concepts focusing on data velocity and volume reduction, while taking into a consideration data variety and preserving its value. Velocity and volume reduction is considered in all parts of IoT system, namely sensors, gateways and cloud. The work is structured to differentiate relevant research fields and key points within the IoT system where Big data optimization can be done.

Data reduction on sensor side can be made on input side, i.e. collecting data with dynamic monitoring frequency or on sensor output where sensor can process data and forward only relevant data towards gateway. Input data on gateway side can be optimized i.e. by sleep scheduling and wake up protocol in order to collect data only from one of correlated sensors. On output side gateways can i.e. aggregate values from multiple sensors at the same moment in time. Just as on gateways input side, similar approach can be applied for clouds input, i.e. outputs from correlated gateways can be collected interchangeably. At the end, as cloud output, data can be stored (in compressed, aggregated, consolidated form, or as raw data) for archive. Additionally, related data can be stored together for quick and easy access.

It is shown that knowledge of future helps reducing energy consumption; either by predicting final value of sensor with long heat-up period [40] or by predicting when will significant change occur in the future i.e. dynamic monitoring frequency. Thus, machine learning techniques can be used to reduce energy consumption. In the future, focus will be on finding light-weight algorithm for sensors that will enable them to collect data only when the significant delta is expected and sleep in the meanwhile to reduce battery consumption.

7. References

[1] Internet of Things (IoT) Connected Devices Installed Base Worldwide From 2015 to 2025 (in Billions). Accessed: July 26, 2020. Available: https://www.statista.com/statistics/471264/iotnumber-of-connected-devices-worldwide/

[2] "Worldwide Global DataSphere IoT Device and Data Forecast, 2020–2024 ", IDC Accessed: August 26, 2020. Available: https://www.idc.com/getdoc.jsp?containerId=prUS45213219

[3] D. Reinsel, J. Gantz, J. Rydning „The Digitization of the World From Edge to Core", IDC Whitepaper, 2018.

[4] J. Č. Gambiroža, T. Mastelić, M. Čagalj, „Data Analytics in Internet of Things", PhD Forum, SoftCOM, 2019.

[5] J. Č. Gambiroža, T. Mastelić, "Big Data Challenges and Trade-offs in Energy Efficient Internet of Things systems", SoftCOM, 2018.

[6] T. Mastelic and I. Brandic, "Data velocity scaling via dynamic monitoring frequency on ultrascale infrastructures", IEEE 7th International Conference on Cloud Computing Technology and Science, pp. 422 – 425, 2015.

[7] D. Trihinas, G. Pallis and M.D. Dikaiakos, "AdaM: an Adaptive Monitoring Framework for Sampling and Filtering on IoT Devices", IEEE International Conference on Big Data , pp. 717-726, 2015.

[8] I. Lujic, V. De Maio, I. Brandic, "Efficient Edge Storage Management Based on Near Real-Time Forecasts," 2017 IEEE 1st International Conference on Fog and Edge Computing (ICFEC), Madrid, 2017, pp. 21-30, doi: 10.1109/ICFEC.2017.9.

[9] D. Zordan, M. Rossi and M. Zorzi, "Rate-Distortion Classification for Self- Tuning IoT Networks", IEEE International Conference on Communications Workshops (ICC Workshops), pp. 857 – 863, 2017.

[10] A.M. bin Baharudin, M. Saari, P. Sillberg, P. Rantanen, J. Soini, T. Kuroda, "Low-Energy Algorithm for Self-Controlled Wireless Sensor Nodes", International Conference on Wireless Networks and Mobile Communications, pp. 42-46, 2016.

[11] S.A. Al-Qaseemi, H.A. Almulhim and M.F. Almulhim, "IoT architecture challenges and issues: Lack of standardization", Future Technologies Conference, pp.731 – 738, 2016.

[12] J. Č. Gambiroža, T. Mastelić, P. Šolić, M. Čagalj, "Capacity in LoRaWAN Networks: Challenges and Opportunities," 2019 4th International Conference on Smart and Sustainable Technologies (SpliTech), Split, Croatia, 2019, pp. 1-6, doi: 10.23919/SpliTech.2019.8783184.

[13] E.I. Gaura, J. Brusey, M. Allen, R. Wilkins, D. Goldsmith, R. Rednic, "Edge Mining the Internet of Things", IEEE Sensors Journal, vol:13, pp.3816-3825, Oct 2013

[14] V. Mushunuri, A. Kattepur, H.K. Rath and A. Simha, "Resource Optimization in Fog Enabled IoT Deployments", Second International Conference on Fog and Mobile Edge Computing, pp. 6-13, 2017.

[15] F. Samie, V. Tsoutsouras, L. Bauer, S. Xydis, D. Soudris and J. Henkel, "Computation Offloading and Resource Allocation for Low-power IoT Edge Devices", IEEE 3rd World Forum on Internet of Things , pp. 7-12, 2016.

[16] M. Liyanage, C. Chang and S.N. Srirama, "Energy-Efficient Mobile Data Acquisition using Opportunistic Internet of Things Gateway Services", IEEE International Conference on Internet of Things, pp. 217 – 222, 2016.

[17] V. Natarajan and A. Vyas, "Power Efficient Compressive Sensing for Continuous Monitoring of ECG and PPG in a Wearable System", IEEE 3rd World Forum on Internet of Things (WF-IoT), pp. 336 - 341, 2016.

[18] K. Wang, Y. Wang, Y. Sun, S. Guo and J. Wu, "Green Industrial Internet of Things Architecture: An Energy-Efficient Perspective", IEEE Communications Magazine, vol. 54, no. 12, pp. 48-54, Dec. 2016.

[19] Khandel, A. H. Rassafi, V. Pourahmadi, S. Sharifian, and R. Zheng,"Sensordrop: A reinforcement learning framework for communicationoverhead reduction on the edge,"CoRR, vol. abs/1910.01601, 2019.

[20] A. Papageorgiou, B. Cheng, and E. Kovacs, "Real-time data reduction at the network edge of Internet-of-Things systems", 11th International Conference on Network and Service Management, pp .284-291, 2015.

[21] C. Razafimandimby, V. Loscri, A. M. Vegni and A. Neri, "A Bayesian and Smart Gateway Based Communication For Noisy IoT Scenario", International Conference on Computing, Networking and Communications, pp.481-485, 2017.

[22] A. Mukherjee, "Energy Efficiency and Delay in 5G Ultra-Reliable Low- Latency Communications System Architectures", IEEE Network, vol:32, issue: 2, pp. 55 - 61, April 2018.

[23] R. Han, F. Zhang, L. Y. Chen and J. Zhan, "Work-in-Progress: Maximizing Model Accuracy in Real-time and Iterative Machine Learning," 2017 IEEE Real-Time Systems Symposium (RTSS), Paris, 2017, pp. 351-353.

[24] M. Aazam, S. Zeadally and K. A. Harras, "Fog Computing Architecture, Evaluation, and Future Research Directions," in IEEE Communications Magazine, vol.56, no.5, pp.46-52, 2018

[25] T. Soultanopoulos, S. Sotiriadis, E.G.M. Petrakis and C. Amza, "Data Management of Sensor Signals for High Bandwidth Data Streaming to the Cloud", IEEE 37th Sarnoff Symposium, pp. 53 – 58, 2016.

[26] C. Ge, N. Wang, K. Xu and J. Wu, "Energy management in cross-domain content delivery networks: a theoretical perspective", IEEE Transactions on Network and Service Management, vol. 11, no. 3, pp. 264 – 277, Sept. 2014.

[27] S. Meng and L. Liu, "Enhanced Monitoring-as-a-Service for Effective Cloud Management", IEEE Transactions on Computers, pp.1705-1720, 2013.

[28] A. Moon, J. Kim, J. Zhang and S.W. Son, "Lossy Compression on IoT Big Data by Exploiting Spatiotemporal Correlation", IEEE HPEC, pp. 1-7, 2017.

[29] A. Eliseev, D. Kachalov and M. Farkhadov, "Modern methods to collect, store, and process big data in large-scale systems", 5th International Conference on Control, Instrumentation, and Automation, pp.179-182, 2017.

[30] P. Kadlec and B. Gabrys, "Soft sensors: where are we and what are the current and future challenges?", 2016.

[31] A. Geron, ”Hands-On Machine Learning with Scikit-Learn and TensorFlow", 1st ed. 1005 Gravenstein Highway North, Sebastopol, CA 95472: O'Reilly, Media, Inc., 2017.

[32] N. McClure, ”TensorFlow Machine Learning Cookbook", 1st ed. Livery Place, 35 Livery Street, Birmingham B3 2PB, UK: Packt Publishing Ltd., 2017.

[33] I. Goodfellow, Y. Bengio, A. Courville, „Deep Learning“, MIT Press, 2016.

[34] R. Atienza, „Advanced Deep Learning with Keras“, Packt Publishing, 2018.

[35] Z. Jia, X. Lyu, W. Zhang, R. P. Martin, R. E. Howard, and Y. Zhang, "Continuous Low-Power Ammonia Monitoring Using Long Short-Term Memory Neural Networks," in The 16th ACM Conference on Embedded Networked Sensor Systems (SenSys), 2018, pp. 1–6.

[36] F. Qian and X. Chen, “Stock Prediction Based on LSTM under Different Stability,” in 2019 IEEE 4th International Conference on Cloud Computing and Big Data Analysis (ICCCBDA), April 2019, pp. 483–486.

[37] TensorFlow. (2019) TensorFlow Lite for Microcontrollers. [Online]. Available: https://www.tensorflow.org/lite/microcontrollers/overview

[38] I. N. Kosović, A. Božić, T. Mastelić and D. Ivanković, "Building Soft Sensors using Artificial Intelligence: Use Case on Daily Solar Radiation," 2019 4th International Conference on Smart and Sustainable Technologies (SpliTech), Split, Croatia, 2019, pp. 1-6, doi: 10.23919/SpliTech.2019.8783137.

[39] Ez-Zazi, M. Arioua, and A. El Oualkadi, “Adaptive joint lossy source-channel coding for multihop iot networks,”Wireless Communicationsand Mobile Computing, vol. 2020, pp. 1–15, 05 2020.

[40] J. Č. Gambiroža, T. Mastelić, T. Kovačević and M. Čagalj, "Predicting Low-Cost Gas Sensor Readings From Transients Using Long Short-Term Memory Neural Networks," in *IEEE Internet of Things Journal*, doi: 10.1109/JIOT.2020.2990526.

[41] C. Arendt, S. Böcker and C. Wietfeld, "Data-Driven Model-Predictive Communication for Resource-Efficient IoT Networks," *2020 IEEE 6th World Forum on Internet of Things (WF-IoT)*, New Orleans, LA, USA, 2020, pp. 1-6, doi: 10.1109/WF-IoT48130.2020.9221019.

[42] Cyril Cecchinel, François Fouquet, Sébastien Mosser, Philippe Collet, „Leveraging live machine learning and deep sleep to support a self-adaptive efficient configuration of battery powered sensors“, Future Generation Computer Systems, Volume 92, 2019, Pages 225-240, ISSN 0167-739X, https://doi.org/10.1016/j.future.2018.09.053.

8. Labels

IoT – Internet of Things

5Vs – volume, velocity, variety, veracity and value

LoRa - Long Range spread spectrum modulation technique

ML – Machine Learning

LSTM – Long Short-Term Memory

9. Abstract

With everyday growth of Internet of Things (IoT), number of connected sensor devices increases as well, where each sensor consumes energy while being constantly online. During that time, they collect large amounts of data in short intervals leading to the collection of redundant and perhaps irrelevant data. Moreover, being commonly battery powered, sensor batteries need to be frequently replaced or recharged. Former requires smarter and less frequent data collection, while latter being complementary to the former requires putting them to sleep while not being used in order to save energy.

In this document, existing research work related to IoT and Big data concepts is surveyed and presented, with the focus on data velocity and volume reduction, while preserving value and variety of data. The work is categorized and structured to differentiate relevant research fields and key points within the IoT system where Big data optimization can be done. The system includes a complete data path from end-point sensors, through network of gateways, to the backend cloud and its users. The paper covers different approaches to Big data optimization in IoT.

It is shown that machine learning can be used as a tool for predicting future values. Predicted values are used to gain knowledge about future what helps reducing energy consumption; either by predicting final value of sensor with long pre-heat period and thus reducing energy, or by predicting future value and deciding if significant change will occur i.e. dynamic monitoring frequency.

Keywords — Internet of Things; Big data; Energy efficiency; Data volume; Data velocity;