

# Influence of on-Device Measurement Analysis on Energy Efficiency in Machine-to-Machine Systems

Pavle Skocir, Mario Kusek and Gordan Jezic  
University of Zagreb  
Faculty of Electrical Engineering and Computing  
Department of Telecommunications  
Unska 3, HR-10000 Zagreb, Croatia  
{pavle.skocir, mario.kusek, gordan.jezic}@fer.hr

**Abstract**—Machine-to-Machine Communication (M2M) enables communication between heterogeneous devices without human intervention. It is considered to be a key enabler technology for the concept of Internet of Things (IoT) and Cyber Physical Systems (CPS). With M2M's integration with Wireless Sensor Networks (WSN), information from different kinds sensors can be obtained. In order to discover useful knowledge from sensor data, various data mining techniques need to be applied. Due to the development of microprocessors on end devices in M2M system which collect data from sensors, data processing can also be executed on those devices. However, since end devices are often battery powered, energy consumption when running those algorithms needs to be taken into account. In this paper we implement an algorithm in M2M system, on Libelium Waspnote devices, which detects temperature plummeting in an indoor space. Afterwards, energy consumption of Waspnote devices is analyzed for two cases: when algorithm is executed on-device and when algorithm is executed on gateway or on back-end system.

## I. INTRODUCTION

MACHINE-to-Machine Communication (M2M), a concept which enables connection of heterogeneous devices with limited human intervention, is considered to be one of the enablers for the process of provisioning advanced applications and services, such as smart cities and hospitals, automated vehicular and industrial operation, along with others [1]. Through integration with Wireless Sensor Networks (WSN), M2M systems can obtain wide range of information [2]. By analyzing that information, useful knowledge can be discovered, and appropriate actions can be initiated. M2M is considered to be one of the fundamental technologies for enabling the concept of Internet-of-Things (IoT) and Cyber Physical Systems (CPS) [3]. The IoT concept includes connecting sensors and other devices to the broader Internet by using general Internet technologies [4]. CPS is considered as evolution of M2M which supports more intelligent and interactive operations, under the architecture of IoT [3]. Although interlaced with the aforementioned areas, in this paper we use the term of M2M systems for a sensing systems in which end devices collect measurements from sensors, and send them via gateway to the back-end system.

M2M systems generate massive data sets which are considered to be of high business value [5]. To extract hidden knowledge from data, data mining algorithms can be applied. For instance, by analyzing sensor measurements collected

within a smart home, the system can detect actions of the inhabitants and predict their future behavior. Additionally, it can recognize outliers, events which are not within usual patterns, and indicate towards a potential problem [6].

Most of the existing M2M solutions incorporate a central point for collecting and analyzing information [7]. However, with the development of hardware technologies which enabled miniaturizing wireless devices, smart sensors or actuators and micro-controllers, and enhancing their processing power, new schemes for refining software for embedded systems started to evolve [6]. The example of those new schemes is the possibility for certain algorithms to be executed on end devices or on gateways, instead of only on back-end system [8].

In this paper we analyze data mining techniques in M2M networks. Furthermore, we want to disclose how the functionalities of nodes within M2M system architecture influence energy efficiency of the system with limited energy resources. Particularly, we compare energy consumption of the battery-powered end devices when end device has measurement analysis functionality (i.e. analysis is performed on-device) and when measurement analysis functionality is performed elsewhere (i.e. when analysis is performed on gateway or on back-end system).

Section 2 presents research activities within the area of interest of this paper. Section 3 describes the network architecture compliant with existing M2M standards in which we conduct our measurements. In section 4 we introduce an algorithm which was deployed on end devices of our system and which monitors temperature fluctuations in an indoor space. Section 5 presents energy consumption comparison of end devices for two cases - when algorithm was executed on end devices, and when it was executed elsewhere. Section 6 concludes the paper and gives an outline for future work.

## II. RELATED WORK

This section presents current research efforts in the area of applying data processing techniques within M2M network. Stojmenovic [7] considers M2M as a key enabling technology for the CPSs. The author identifies the problem that existing work in the area of M2M communication is based on small-scale M2M models and centralized solutions, while a few existing distributed solutions do not scale well. A paradigm

shift is suggested where end nodes should also make decisions based on local knowledge, instead of only forwarding collected messages to back-end system. By using this new paradigm, M2M solutions could scale to a significantly larger number of M2M devices. In the use-case example, the author considers a smart building control application in which sensors and actuators exchange information directly, without communication with servers, and coordinate by using distributed decision making to react to data. This reactions include opening of windows or injecting fresh air when needed.

Chen et al. [5] introduce an overview of data mining techniques in the area of Internet of Things. They present five data mining functionalities: classification, clustering, association analysis, time series analysis, and outlier analysis. The application areas of data mining are also presented: e-commerce, banking, retail, health care, and city governance. The main research issues are identified in the area of finding erroneous data, analysis of data streams and developing a framework to support big data mining. As for the nodes where data processing is performed, the authors suggest servers in the cloud where open source solutions like Hadoop, HDFS, Storm or Oozie can be used. According to their perception, end devices are used only for data gathering and forwarding towards cloud.

Bruns et al. [9] state that in traditional M2M systems, data processing is usually hard-coded and scattered all over source code, which makes it difficult to maintain. Therefore, they propose a complex event processing system (CEP) which separates event processing from source code. In CEP, event processing is encapsulated in rules which can be efficiently adapted and maintained. The authors discuss the application of the proposed system in solar power plants and printer supply and maintenance service. CEP is implemented on M2M server, it processes data streams just as new data arrives to the server.

M2M systems where end devices are serving as nodes which collect measurements from sensors are referred to as Wireless Sensor Networks (WSN). Analytics in WSN is also an ongoing research topic closely connected with the concept of M2M systems. Mahmood et al. [10] propose a taxonomy of data mining techniques for WSN. First level of classification is connected with general data mining classes: frequent pattern mining, sequential pattern mining, clustering and classification. Second level of classification is based upon the ability of the approach to process data in a centralized or distributed way. The third level of classification is determined according to the focus on two different aspects - WSN performance issues and application issues. The approaches which focus on WSN performance issues try to take into account resource constraints like energy, memory, and communication bandwidth. On the other hand, approaches which focus on application issues try to satisfy application requirements without much consideration for WSN performance. The authors group existing data mining techniques for WSN according to the presented taxonomy. Moreover, they also discuss what is the focus of this paper - on which nodes to execute different data mining techniques. Sensor nodes, referred to as M2M devices in M2M system

architecture, perform single pass algorithms and forward only the required and partially processed data to the network. In network, referred to as M2M gateway, data from various end devices is collected, and activities as network pattern identification are carried out. On sinks, referred to as M2M servers, computationally demanding tasks are executed.

Alsheikh et al. [11] present an overview of existing machine learning techniques used in WSNs. They group them into the categories of supervised, unsupervised and reinforcement learning. As important aspects which need to be taken into account when deploying machine learning techniques in WSNs, the authors emphasize power and memory constraints of sensor nodes, topology changes, communication link failures and decentralized management. The main functional challenges for which machine learning techniques were adopted include routing in WSN, clustering and data aggregation, event detection and query processing, localization and objects targeting, and medium access control. As for the nodes where processing is taking place, the authors also consider in-network processing of data since it enables the nodes to rapidly adapt their future behavior and predictions in correspondence with current environmental conditions. However, when executing learning algorithms on-device, special attention needs to be paid not to exhaust the nodes with complex and resource demanding computational tasks.

Suryadevara et al. [6] developed a smart home solution which enables identification of the Activities of Daily Living (ADL) in order to determine the wellness of elderly people. This was done by processing time series of data collected by the sensors deployed at users' homes. Among other things, the authors discuss different storage mechanisms for WSN data. They identify two approaches for storing data in WSNs - centralized and decentralized way. In centralized approach, data is stored and can be analyzed on a node which generates it. In decentralized approach, data is stored on different nodes. The most common decentralized storage approach is data centric storage, where data is stored on a node called sink. Centralized approach is identified as not appropriate for a setup with recurrent bursts of activities since it quickly overfills memory resource. Moreover, when sequences gathered from different sensors need to be processed, as is the case in the solution developed by the authors, data-centric storage appears to be a better solution. Therefore, the authors developed a system which stores sensor data in the form of event activities on a central system which then analyses that data and makes assumptions of ADLs.

Research efforts analyzed within this section focus on presenting current trends in the area of data processing in M2M/IoT/WSN domain. However, we have not identified solutions which would take into consideration optimal position in the network to perform data processing with regard to energy efficiency of the system with limited energy resources.

### III. M2M ARCHITECTURE

The architecture of our M2M system is shown in Figure 1, and is compliant with functional M2M specification from

oneM2M<sup>1</sup> [12] and high-level M2M system architecture by ETSI<sup>2</sup> [13].

Application Dedicated Node (ADN) contains sensors which collect measurements from their physical environment. It contains at least one Application Entity (AE), which is an entity in the application layer that implements M2M application service logic. Examples of these application entities are power metering application or remote blood sugar monitoring application. ADN does not contain Common Services Entity (CSE) which is situated on Middle Node (MN) or Infrastructure Node (IN). CSE is a set of common service functions of the M2M environment, like data management, device management, subscription management and location services. Middle node contains CSEs and can contain one or more AEs. When ADN contains CSE, it is called Application Service Node (ASN). ADN/ASN can communicate directly with IN, or can communicate with IN via MN. In our case scenarios, ADN and ASN communicate via MN. ADN/ASN, MN, IN, defined in oneM2M standard [12], can be referred to as M2M Device, M2M Gateway and M2M Server respectively in previous standards by ETSI [13].

In this paper we focus on M2M communication within smart homes which can include following sensors: temperature, humidity, luminosity, presence, hall effect, electricity consumption or water metering. Based on the data collected by these sensors, various events can be detected, for instance if residences are present in the house, if temperature fluctuates unnecessarily (e.g. a window is opened too long during cold winter or hot summer days) and should be controlled, if power is more affordable in certain times of day and when should specific consumers with variable operating times like washing machine be turned on etc. Moreover, the system can detect outliers and report them to the interested users, like water leakage, extreme temperatures or unusual power consumption. Our system is composed of ADNs/ASNs which

<sup>1</sup>oneM2M (<http://www.onem2m.org/>) - organization which develops standards for M2M and the Internet of Things

<sup>2</sup>ETSI (<http://www.etsi.org/>) - European Telecommunications Standards Institute

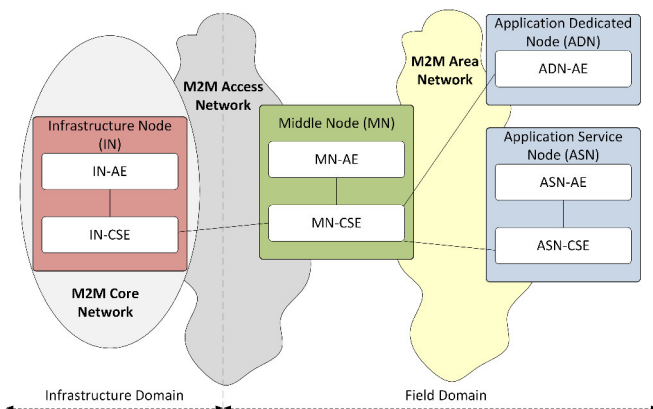


Fig. 1: M2M Architecture

monitor temperature fluctuations and an MN through which the devices communicate with IN.

The easiest way to perform the analysis of data collected by sensors is to transfer it to infrastructure domain where servers have plenty of memory and processing power [14]. However, due to rapid development of embedded devices which have more and more processing power and due to application needs for faster response, processing can also take place in field domain, on MNs and ADNs/ASNs [11], [15]. When executing in-network data processing, special attention needs to be paid not to exhaust the nodes with complex computational tasks. By analyzing data in networks, valuable information instead of raw data is delivered to infrastructure domain, from where it can be easily accessed by user applications.

In the first case scenario of our energy consumption measurement process, explained in detail in Section V, end device is according to oneM2M specification an ASN because it contains both AE and partially CSE - data management. In the same scenario, gateway is MN with only CSE functionality. In the second case scenario, end device is an ADN since it contains only AE, while CSE - data management and other functionalities, as well as a part of Application Entities (AE) is executed on MN (gateway).

#### IV. OUTLIER DETECTION ALGORITHM

Temperature function shown in Figure 2 recorded temperature values at our laboratory during one week in winter from February 23<sup>rd</sup> until March 1<sup>st</sup>. Vertical dotted lines separate measurements belonging to a certain day of the week. The measurements were obtained from sensors and forwarded to back-end system every 10 seconds. Three peaks can be identified, which are marked on Figure 2 and which occurred when the window was opened, and no-one was present in the room. In these cases, temperature fell to around 12°C. The algorithm that we propose has its main goal to raise the alarm when temperature plummets like in those 3 cases. The reason for that is because in such occasions the temperature in the room was too low to reside there. By raising the alarm, window could be closed sooner to prevent those unpleasant conditions.

In order to determine the algorithm which could detect rapid temperature fall, which we define as an outlier from normal temperature fluctuation during the day, we monitored the derivative of the temperature function. Since the temperature function is discrete, its derivative, often referred to as backward difference, is calculated by using the following expression [16]:

$$\Delta_n f(n) = f(n) - f(n-1) \quad (1)$$

where  $f(n)$  is a current temperature value measured by end device, while  $f(n-1)$  is the value measured in the last measurement. In our case, the difference between those two consecutive measurements is 10 seconds. By calculating the difference of the function, we wanted to analyze how does the difference of the function behave when the temperature plummets. When analyzing the difference between two consecutive

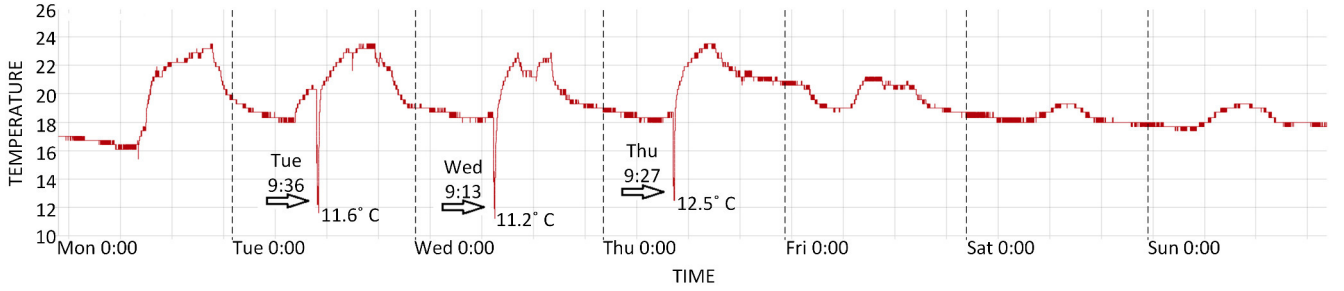


Fig. 2: Measurements from temperature sensor during 1 week in winter

values, extreme descends as the ones seen in Figure 2 cannot be detected since in those events the temperature does not fall in every interval. However, if we monitor the difference in larger time window, which is a good approach when dealing with data that arrives continuously in time [17], rapid decrease can be detected when comparing current measurement with the 5th historic measurement. The equation of such differences calculation is the following:

$$\Delta_n f(n) = f(n) - f(n - 5) \quad (2)$$

where  $f(n)$  is current measured value, and  $f(n - 5)$  is 5th historical measurement, measured 50 seconds ago. We have tried to use more recent historic measurements (1st-4th), but in those cases certain temperature decreases were captured which were not so steep as the ones we wanted to identify. When running an algorithm which analyzes when difference values calculated as in Equation 2 are falling in 10 more than consecutive intervals, extreme descends in temperature like the ones marked in Figure 2 can be detected. The number of falling intervals was set according to empirical evaluation. Table I shows the number of falling intervals for the cases when temperature plummeted. The times of these events were 9:36 on Tuesday, 9:13 on Wednesday, and 9:23 and 9:27 on Thursday. The reason for two captured intervals on Thursday is because temperature was falling for a longer period on that day. The temperature for that case did not fall in every interval, in some intervals it remained unchanged or even slightly grew. Since the algorithm was analyzing constant falls of differences from Equation 2, this particular outlier was captured twice. If the number of intervals in which descend is monitored is between 7 and 10, those rapid falls can always be detected. However, if the number of falling intervals is smaller than 7, then some other events can be captured, which are not interesting to us, like the fall of temperature during night when the heating was off. Since the algorithm in which 10 consecutive falling intervals are detected fits to our needs

TABLE I: Captured outliers when looking for negative difference in more than 10 consecutive measurements

time	number of falling intervals
Tue 24.2.2015 9:36:04	12
Wed 25.2.2015 9:13:01	10
Thu 26.2.2015 9:23:00	15
Thu 26.2.2015 9:27:08	10

(it identified rapid temperature fall caused by the opened window), it was implemented on our M2M system in order to monitor energy consumption of end devices in different case scenarios described in detail in the next section.

## V. ENERGY CONSUMPTION ANALYSIS

The proposed algorithm described in Section IV was implemented on Libelium Waspote devices v1.2 [18] with XBee communication modules to monitor energy consumption of end devices for two case scenarios: when the algorithm is executed on end devices and when algorithm is executed on gateway or on back-end system. Since device consumes different amounts of energy by executing different tasks during its operating cycle, it was necessary to identify those tasks, measure its duration and power consumed during the execution. Figure 3 shows the tasks and their order of execution for both scenarios.

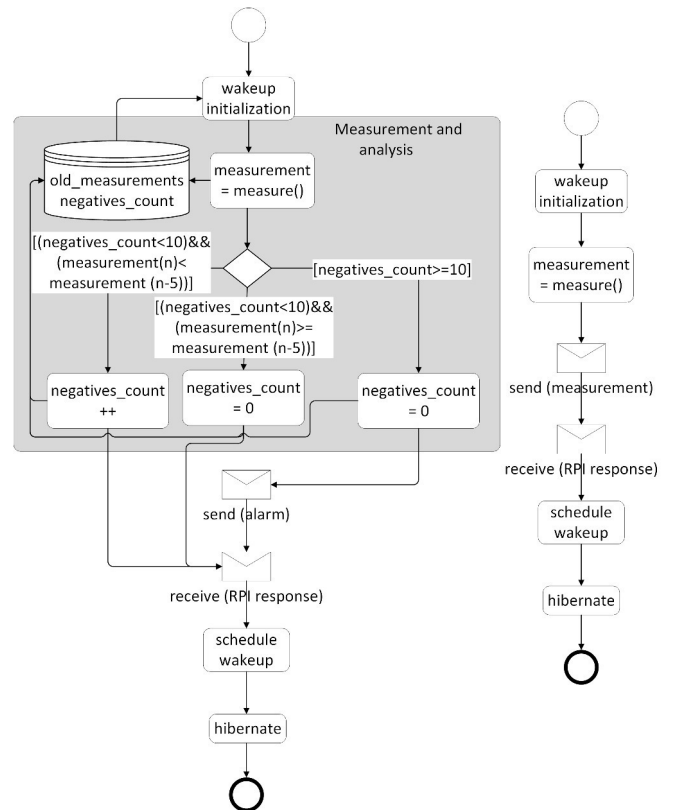


Fig. 3: Activity diagram of end device

In each operating cycle, device wakes up from hibernate mode, which is a state on Wasp mote device where lowest amount of energy are consumed. Afterwards, it obtains measurements from sensors. In the first case scenario, shown on the left-hand side of Figure 3, that task is followed by executing the algorithm to detect rapid temperature decrease. When the overseen decrease is identified, the device reports it by sending a message to the gateway. After reporting the alarm message, it waits for a certain amount of time to receive a new message from gateway which contains information for a new task, e.g. if measurements are to be sent in a specified interval instead of forwarding only alarms. If no alarm needs to be raised, the device only waits for a message from the gateway. After the time for receiving message has expired, the device is ready to return to hibernate mode. The process of exchanging messages between end devices and gateway in order to reach an agreement about operating times is described in our previous work [19].

In the second case scenario, shown on the right-hand side of Figure 3, after wake up from hibernate mode and obtaining a measurement from sensor, the end device forwards that data to gateway. Since end device does not execute algorithm for detecting temperature decrease, acquired measurement needs to be sent to back-end system for further analysis. As in the first case scenario, the end device then waits for messages from gateway. After the eventual processing of the received message from gateway, end device goes back to hibernate mode.

Table II shows power consumption and duration of tasks executed on end devices when analysis is performed on-device (ASN functionality), while Table III shows power consumption and duration of tasks executed on end devices when analysis is performed on gateway or back-end system (ADN functionality of devices). Current consumption and voltage levels are measured by using Rigol DS1102D oscilloscope<sup>3</sup>.

TABLE II: Power consumption and duration of tasks in the case of on-device analysis

state/task	wake-up initialization	measure	analyze	send	receive	handle response
power consumption (mW)	70,13	130,83		258,95	262,68	141,12
duration (ms)	80	70		255,3	208,8	1

TABLE III: Power consumption and duration of tasks when analysis is not performed on-device

state/task	wake-up initialization	measure	send	receive	handle response
power consumption (mW)	70,13	77,7	258,95	262,68	141,12
duration (ms)	80	10	255,3	208,8	1

Figure 4 shows the energy consumption for one operating cycle. The left-hand and central columns represent energy consumption for the first case scenario when analysis is performed on-device. Since the end device in that scenario needs to raise an alarm, it sends a message to the gateway

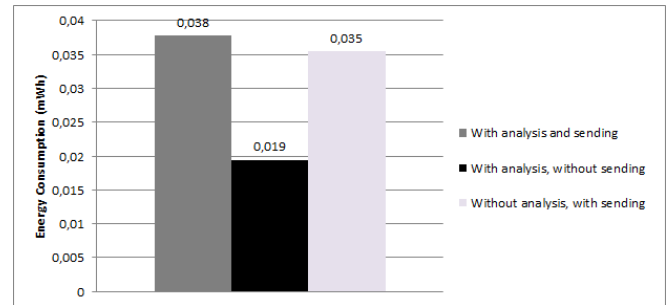


Fig. 4: Energy consumption of end devices during one operating cycle

only when temperature value plummets. The consumption of that operating cycle is shown in the left-hand column. When rapid temperature decrease is not detected, the device does not send any data to gateway. The consumption of such operating cycle is shown in the central column. The consumption of the operating cycle in second case scenario, when end device reads the value from sensor and forwards it to the gateway, is shown in the right-hand column of Figure 4.

It can be observed that energy consumption presented in the left-hand and right-hand columns is similar. It appears that on-device analysis does not add much to total energy consumption of the device in one operating cycle. However, when comparing the power consumption for measurement and analysis task in Table II - 130.8 mW and only measurement task in Table III - 77.7 mW, it can be observed that for the scenario when data is analyzed, the power consumption is around 70% higher. But since both actions last for quite a short time (0,07 s and 0,01 s), they do not have a large influence on the overall energy consumption. On the other hand, energy consumption shown in the central column is about 50% lower than in two aforementioned cases. The reason for that lies in the fact that sending data has a high influence on overall energy consumption. Although it consumes only twice as much power compared to measuring and processing, it lasts around 3.5 times longer.

Total energy consumption during one week for the two cases when analyzing data presented in Figure 2 is shown in Figure 5. Left-hand column shows consumption for the first case scenario when analysis is performed on-device. In that scenario, in most of the operating cycles the device was only performing measurements and analysis without sending alarm message to the gateway due to the fact that on the

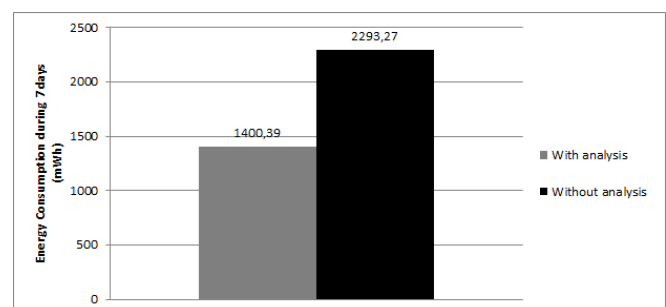


Fig. 5: Energy consumption of end devices during 1 week

<sup>3</sup><http://www.rigolna.com/products/digital-oscilloscopes/ds1000d/ds1102d/>



analyzed data set alarms were needed to be raised only on four occasions. Since operating cycle which includes analysis without sending consumes around 45% less energy than the operating cycle in which the measurement data is forwarded to back-end system, the total consumption of the first case scenario is around 40% less than the consumption of the second case scenario which is shown in the right-hand column in Figure 5 and in which data was sent during each operating cycle.

As a power source for end devices used in this experiment we had at our disposal batteries with capacity of 6600 mAh. According to energy consumption shown in Figure 5, and by taking into account that the average voltage of the battery is 3.7 V, by using the first case scenario, device can operate for 17.4 weeks. By using the second case scenario, the device can remain operational for 10.6 weeks. For this particular application and data set, on-device analysis extends end-device lifetime for about 65%.

## VI. CONCLUSION

This work presented the analysis of the influence of data measurement analysis on energy efficiency in Machine-to-Machine system with Libelium Waspote v1.2 devices. By implementing a specific outlier detection algorithm on end devices, it was shown that on-device data analysis when using this algorithm does not consume as much energy as communication. More energy can be saved by reducing communication. As a result, it can be advisable to perform local data analysis on end devices in those cases when the duration of the measurement and analysis tasks is shorter than the duration of communication tasks.

If the data set were different and temperature plummeting occurred more often, the difference in total energy consumption during one week would have been smaller. However, since temperature outliers do not occur often, the scenario in which the measurement analysis is performed on-device will usually spend less energy than the scenario where measurement analysis is performed on gateway or on back-end system.

In future work, we plan to compare energy efficiency for more sophisticated algorithms in smart home environment which would enable event detection. We also plan to extend this research to different types of applications, not only the applications which should raise alarms, but applications which require continuous or periodical streaming.

## ACKNOWLEDGMENT

This work was supported by strategic project "Energy Efficient M2M Device Communication" in cooperation with FTW - Forschungszentrum Telekommunikation Wien GmbH, Austria and by Croatian Science Foundation under the project 8065.

## REFERENCES

- [1] R. Ratasuk, A. Prasad, Z. Li, A. Ghosh, and M. Uusitalo, "Recent advancements in M2M communications in 4G networks and evolution towards 5G," in *Intelligence in Next Generation Networks (ICIN)*, 2015 18th International Conference on, 2015, pp. 52–57. doi: <http://dx.doi.org/10.1109/ICIN.2015.7073806>
- [2] J. Zhang, L. Shan, H. Hu, and Y. Yang, "Mobile cellular networks and wireless sensor networks: toward convergence," *Communications Magazine, IEEE*, vol. 50, no. 3, pp. 164–169, 2012. doi: <http://dx.doi.org/10.1109/MCOM.2012.6163597>
- [3] J. Wan, H. Yan, Q. Liu, K. Zhou, R. Lu, and D. Li, "Enabling cyber-physical systems with machine-to-machine technologies," *Int. J. Ad Hoc Ubiquitous Comput.*, vol. 13, no. 3/4, pp. 187–196, 2013. doi: <http://dx.doi.org/10.1504/IJAHUC.2013.055454>
- [4] J. Hiller, V. Tsiatsis, C. Mulligan, S. Karnouskos, S. Avesand, and D. Boyle, *From Machine-to-Machine to the Internet of Things: Introduction to a New Age of Intelligence*. Academic Press, 2014. ISBN 978-0-12-407684-6
- [5] F. Chen, P. Deng, J. Wan, D. Zhang, A. V. Vasilakos, and X. Rong, "Data mining for the internet of things: Literature review and challenges," *International Journal of Distributed Sensor Networks*, in press.
- [6] N. K. Suryadevara and S. C. Mukhopadhyay, *Smart Homes: Design, Implementation and Issues*. Springer, 2015. ISBN 978-3-319-13556-4
- [7] I. Stojmenovic, "Machine-to-machine communications with in-network data aggregation, processing, and actuation for large-scale cyber-physical systems," *Internet of Things Journal, IEEE*, vol. 1, no. 2, pp. 122–128, 2014. doi: <http://dx.doi.org/10.1109/IJOT.2014.2311693>
- [8] I. Stojmenovic and S. Wen, "The fog computing paradigm: Scenarios and security issues," in *Proceedings of the 2014 Federated Conference on Computer Science and Information Systems*, vol. 2. IEEE, 2014, pp. 1–8. doi: <http://dx.doi.org/10.15439/2014F503>
- [9] R. Bruns, J. Dunkel, H. Masbruch, and S. Stipkovic, "Intelligent M2M: Complex event processing for machine-to-machine communication," *Expert Systems with Applications*, vol. 42, no. 3, pp. 1235 – 1246, 2015. doi: <http://dx.doi.org/10.1016/j.eswa.2014.09.005>
- [10] A. Mahmood, K. Shi, S. Khatoun, and M. Xiao, "Data mining techniques for wireless sensor networks: A survey," *International Journal of Distributed Sensor Networks*, vol. 2013, pp. 1–24, 2013. doi: <http://dx.doi.org/10.1155/2013/406316>
- [11] M. Abu Alsheikh, S. Lin, D. Niyato, and H.-P. Tan, "Machine learning in wireless sensor networks: Algorithms, strategies, and applications," *Communications Surveys Tutorials, IEEE*, vol. 16, no. 4, pp. 1996–2018, 2014. doi: <http://dx.doi.org/10.1109/COMST.2014.2320099>
- [12] oneM2M, "M2M Functional Architecture," Technical Specification, draft, 2015. [Online]. Available: [http://www.onem2m.org/images/files/deliverables/TS-0001-Functional\\_Architecture-V1\\_6\\_1.pdf](http://www.onem2m.org/images/files/deliverables/TS-0001-Functional_Architecture-V1_6_1.pdf)
- [13] ETSI, "Machine-to-Machine communications (M2M); Functional architecture," Technical Specification ETSI TS 102690 V2.1.1, 2013. [Online]. Available: [http://www.etsi.org/deliver/etsi\\_ts/102600\\_102699/102690/02.01.01\\_60/ts\\_102690v020101p.pdf](http://www.etsi.org/deliver/etsi_ts/102600_102699/102690/02.01.01_60/ts_102690v020101p.pdf)
- [14] S. Kitagami, M. Yamamoto, H. Koizumi, and T. Sukanuma, "An M2M Data Analysis Service System Based on Open Source Software Environments," in *Advanced Information Networking and Applications Workshops (WAINA)*, 2013 27th International Conference on, 2013, pp. 953–958. doi: <http://dx.doi.org/10.1109/WAINA.2013.124>
- [15] Cisco Systems, "Proposed Computation and Analytics Use Case," Input contribution, 2013. [Online]. Available: [ftp.onem2m.org/Meetings/REQ/2013%20meetings/20130225\\_REQ-ARC19\\_REQ20\\_San%20Francisco/oneM2M-REQ-2012-0102R01-Analytics\\_for\\_oneM2M.DOC](ftp.onem2m.org/Meetings/REQ/2013%20meetings/20130225_REQ-ARC19_REQ20_San%20Francisco/oneM2M-REQ-2012-0102R01-Analytics_for_oneM2M.DOC)
- [16] B. Hamrick, "Discrete calculus." [Online]. Available: <http://homepages.math.uic.edu/kauffman/DCalc.pdf>
- [17] N. C. Krishnan and D. J. Cook, "Activity recognition on streaming sensor data," *Pervasive Mob. Comput.*, vol. 10, pp. 138–154, 2014. doi: <http://dx.doi.org/10.1016/j.pmcj.2012.07.003>
- [18] Libelium Comunicaciones Distribuidas S.L., "Waspote," Technical Guide, 2013. [Online]. Available: [http://www.libelium.com/uploads/2013/02/waspote-technical\\_guide\\_eng.pdf](http://www.libelium.com/uploads/2013/02/waspote-technical_guide_eng.pdf)
- [19] M. Kusek, I. Lovrek, and H. Maracic, "Rich presence information in agent based machine-to-machine communication," *Procedia Computer Science*, vol. 22, no. 0, pp. 321 – 329, 2013. doi: <http://dx.doi.org/10.1016/j.procs.2013.09.109>