Fog-based Data Fusion for Heterogeneous IoT Sensor Networks: A Real Implementation

Fredy João Valente
Department of Computing
Federal University of São Carlos (UFSCAR)
São Carlos – SP, Brazil
ORCID 0000-0002-2735-4787

João Paulo Morijo
Department of Computing
Federal University of São Carlos (UFSCAR)
São Carlos – SP, Brazil
email: jpmorijo@dc.ufscar.br

Kelen Cristiane T. Vivaldini
Department of Computing
Federal University of São Carlos (UFSCAR)
São Carlos – SP, Brazil
email: vivaldini@dc.ufscar.br

Luis Carlos Trevelin
Department of Computing
Federal University of São Carlos (UFSCAR)
São Carlos – SP, Brazil
email: trevelin@dc.ufscar.br

***Abstract*— The Internet of Things (IoT) is an environment that can be divided in three large layers: the sensor/actuator level where a wide variety of objects with different computing, sensors and communication capabilities resides, the communication layer with wireless technologies such as ZigBee, Bluetooth and emerging 6LoWPAN (e.g LoRa), and the intelligence layer, where computing analytics/decisions occur. IoT can be used for monitoring, inferring problems, decision making at a business level or actuating at the edge via IoT nodes. As the IoT sensor network grows, an enormous amount of data from multiple sources flows to the intelligence layer. In order to make decisions based on analytics over these data, the measurements need to be precise and accurate. Data fusion is an effective way to improve data quality, however, IoT environments are still evolving and the best way and location where data fusion should happen is an open problem. This paper presents one potential strategy for IoT sensor data fusion by implementing multi-sensor data fusion as microservices using a container platform built into an opensource IoT middleware based in a fog computing infrastructure which is can scale automatically as the influx of data from the IoT nodes grows. A number of data fusion tests were performed for different amounts of IoT nodes and sensor readings over ZigBee and LoRa using a specific data fusion algorithm. The results show that, the strategy can be effectively used in IoT heterogeneous environments.**

Keywords—IoT, data fusion, fog computing, IoT middleware, micro service container, sensor network

I. Introduction and Motivations

The Internet of Things (IoT) refers to an environment with a number of heterogeneous intelligent objects totally interconnected, capable of communicating through the Internet using many different transport protocols [7][9]. IoT is a part of the Future Internet [11] with billions of very low-cost tiny objects with low energy consumption, different processing and memory capabilities which resorts to a high-level intelligent layer whenever specific service is needed. These devices reside at the edge of the Internet and are known as "things", therefore IoT [9]. In IoT, "things" have capacities such as remote sensing and actuation based on IoT nodes wirelessly interconnected. The sensing/actuation capacity of a population of objects present in an environment have become a synonym to the concept of pervasive or ubiquitous computing, defining IoT capacity as an environment capable of sensing-computing-actuating [9]. IoT will be applied to most industries in innovative applications and in augmenting of existing applications, proving pervasive connections and sensing to machinery. However, the effort for adapting the infrastructure, communication, interfaces, protocols and standards among others, and the heterogeneity of such environments, might lead to a decision of replacing existing devices to more up-to-date IoT devices, which is not always possible when we consider the level of automation and the complexity of the existing solution. That explains why the full potential of the IoT will be difficult to realize. IoT will be a major producer of big data and become "a global infrastructure for the information society, which will rely on shared data from IoT, enabling advanced services by interconnecting (physical and virtual) things based on existing and evolving interoperable information and communication technologies" [1].

Data fusion is defined as the theory, techniques and tools which are used for combining sensor data, or data derived from sensory data, into a common representational format [1][5][8]. A timely fusion and analysis of big data (volume, velocity, variety, and veracity), acquired from IoT and other sources, to enable highly efficient, reliable and accurate decision making and management of ubiquitous environments would be a grand future challenge. Computational intelligence would play a key role in this challenge. In this context, this work makes a contribution by experimenting one potential strategy for IoT sensor data fusion in the fog instead of performing it at the IoT nodes which in a real scenario are based on a variety of hardware features and different computing capabilities, some very limited, making it difficult to implement and maintain a stable data fusion algorithm in every IoT node, which are likely to be by the thousands. To overcome such problem, we did implement a multi-sensor data fusion as microservices using a container platform built into an opensource IoT middleware based in a fog computing infrastructure. This paper is organized as follows: section II discusses the strategies for using different protocols, architecture features, standards and communication technologies, section III verses about data fusion, section IV, describes the implementation of data fusion strategy in the fog, followed by the discussion of the results achieved by performance tests in section V which shows that the strategy is feasible and finally a conclusion in section VI.

II. IoT Heterogeneity Challenge

The development of the IoT has enabled a variety of different applications in different markets and sectors and new business models such as PSS – Product Service Systems [13]. Intelligent things are part of end-to-end processes which make decisions based on context information provided by IoT [9]. IoT enables digital business by merging the physical and virtual worlds in a continuous effort over existing processes and solutions [11].

978-3-903176-24-9

IoT can be viewed as a distributed computing environment composed by three large layers: 1. Sensor/actuator network layer composed of a wide variety of sensor objects (the IoT nodes), computational capacities, architectural features, and different communication interfaces and standards for interconnection; 2. Communication layer which make use of different wireless communication technologies including WiFi – IEEE 802.11abgn, ZigBee – IEEE 802.15.4, Bluetooth 4.2/5 – IEEE 802.15 and emerging technologies such as LoRa based on LoRaWAN open standard, SigFox and NB-Weightless-P among others [9]. 3. Intelligence layer located in a fog/cloud computing supported by an IoT middleware which manages IoT nodes by means of Virtual Objects (VO) including security, sensor data collection and actuation messages. The IoT middleware also provides connection to NoSQL for storing fused sensor data.

In IoT, message transport protocols need to be agnostic to the communication technology used at link level. MQTT (Message Queue Telemetry Transport) – ISO/IEC PRF 20922, a publish/subscribe lightweight messaging protocol designed for constrained environments such as M2M (Machine to Machine) communications and the IoT middleware, is becoming the most adopted standard. MQTT is a many-to-many communication protocol which support three semantics for persistence: at most once, at least once and exactly once, very adequate for requirements of continuously event reporting from IoT sensor reads [9]. MQTT allows a continuous transportation of event sensor data with a very low overhead. Wireless technologies such as LoRa, provides long range communication, up to 25-30km with very low bandwidth of around 9600-48000 bps. Usually LoRa implementations drop completely the TCP/IP stack resorting to LoRa frames routing only up to the gateway in order to minimize the effect in the latency due to the stack processing overhead of the protocol. MQTT data is unstructured, meaning that endpoints at IoT nodes and endpoints at IoT middleware must know in advance how to encode and decode exchanged messages. CoAP – Constrained Application Protocol [9], is another protocol specifically designed for IoT and M2M – directed to simple low power devices at the IoT node level. It uses the request/reply model using a RESTful paradigm and a built-in service and resource discovery. CoAP has the advantage of providing inbuild support for metadata and content negotiation making it very suited for transferring state information between IoT nodes and the middleware making it suitable for solutions with central intelligence, which is not the case in this project. This paper focus on the provision of data fusion built into the IoT middleware in order to deal with the heterogeneity encountered on the sensor and communication layers.

III. Data Fusion for IoT

By 2020, it is expected that the number of IoT sensor objects might reach 50 billion becoming the main generator of Big Data. A major challenge will be to provide data fusion capability to enhance the quality of data for analysis in proper and flexible manners in the IoT environment in different applications such as smart cities and agricultural farms. In IoT, large amount of data is produced in small periods of time – reliable data and accurate information are critical in IoT – the amount of errors in data collected grows with the volume. The basis for planning, decision-making and control of intelligent autonomous machines [15] is data accuracy which can be achieved through the use of data fusion, an efficient way of optimally utilizing large volumes of data from multiple sources [3][6][10]. Multi-sensor data fusion seeks to combine information from multiple sensors to achieve inferences that would not be possible from a single IoT sensor data source. The core of the fusion and data is not in the architecture but in the data fusion methods [4]. However, depending on the method, the computational power required may be higher or lower – in this way the execution of data fusion at IoT nodes is dependent on available computing power. On the other hand, the implementation of data fusion in IoT middleware at the fog can be crucial for applications that require the reading of data from numerous sensor sources and data analysis through the use of artificial intelligence for predicting breaks based on inference and decision making of actuation in different IoT nodes. This explains why we have taken decision to implement the sensor data fusion in the fog. Artificial intelligence (AI) also has a crucial role in IoT especially in IoT applications where prediction and inference are used in the process of decision making. In this way, implementing the data fusion in the fog enables the evolving of different algorithms including those for Machine Learning (ML) or Deep Learning (DL) over fused data, in a more straightforward manner. Application areas of IoT such as autonomous vehicles can use data fusion strategies such as DL which is commonly used in complex multimodal learning procedure, as an example in the extraction of correlated audio and video characteristics. DL model can also be applied for deep multimodal fusion of discrete events [2]. DL-based data fusion methods domains include data fusion for activity recognition, data fusion for network traffic and pedestrian detection [1] – all of them IoT enabled applications. The use of DL techniques in data fusion demand computational resource consumption and a very large amount of training data, another reason to locate the IoT data fusion at the fog.

IV. Fog based IoT Data Fusion Implementation

Scenario: The work described in this section is a part of a research project which focus on precision agriculture. At present, data collection and analysis for precision agriculture in Brazil is either done locally using sensors connected to onboard computers in tractors and trucks or data is entered manually in handheld devices and sent to a central database using mobile communication when available or satellite communication in remote areas [14]. Sample of soil, crops, pests and so on, are collected in many different locations and then taken a laboratory for analysis. It is an expensive and inefficient methodology, not to mention that it is nearly impossible to monitor and make actuations remotely due to data latency. Based on this scenario, the main project aims to develop a low cost multisensory IoT solution architecture to replace and also incorporate existing solutions which are based on a variety of devices, software, proprietary data formats and interfaces as well as several communication technologies such as 3G, 4G, WiFi, ZigBee and satellite. The requirements for the project corroborate with the discussion made here so far: the solution should be able to deal with a variety of existing solutions such as micro weather stations disconnected to the internet, be able to incorporate existing solutions, be able to collect sensor data over very long distances in sugar cane fields, from a variety of sensors. To attend that, we decided to use two different IoT modules for interfacing with existing sensors and for connecting weather stations to the Internet: one based on Arduino equipped with ZigBee radio and the other based on ESP32 with a LoRa and WiFi radios onboard. For the gateway side, the ZigBee networks uses a Raspberry Pi 3 B+ and the LoRa network uses the same ESP32. The initial solution was developed for collecting sensor data from micro meteorological stations positioned at distances varying from 10 to 2000 meters using ZigBee (up to 500 meters) and LoRa.

The proposed solution architecture in Fig.1 depicts the IoT nodes connecting with a fog-based IoT middleware through LoRa and ZigBee gateways. The IoT middleware is an opensource construction from Kaa IoT [16] version 0.9 which features an architecture based on a cluster of processing nodes. Each processing node run a combination of services for control, operations and bootstrap. Control services manages system data, processes web APIs and external system calls. Apache Zookeeper maintains nodes running with high availability.

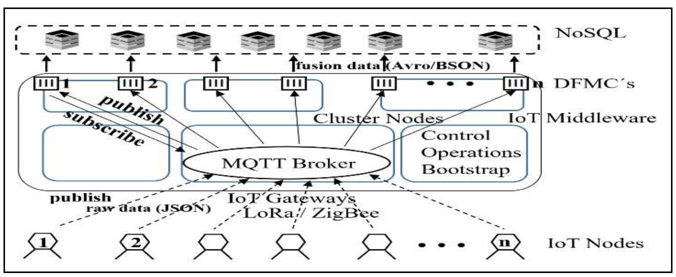

Fig. 1. Fog-based data fusion architecture for heterogeneous IoT sensors

The operations service is in charge of communicating with multiple IoT nodes concurrently. When all the processing nodes are setup with operations services enabled, the cluster scales horizontally automatically also providing load balancing in the processing nodes. The bootstrap service is in charge of managing IoT nodes and connection parameters which might include IP address, TCP port, security credentials and so on. It maintains the list of services operations available which can be retrieved by the IoT nodes. IoT nodes are maintained as virtual objects (VO´s) inside the IoT middleware. In this project, messages are handled by a MQTT broker built on top of the IoT middleware as shown in Fig.1, for routing incoming messages from IoT nodes to topic subscribers using JSON format. The topics in this case are sensor data reads from IoT nodes and the subscribers are DFMC - data fusion microservice containers. Each VO, an IoT node metadata structure, has at least one correspondent DFMC. In the present implementation, JSON formatted messages from IoT nodes are made available for consuming at the corresponding microservice through the IoT cluster nodes operations services which is running the MQTT broker. The microservice containers platform used to implement DFMC in this project is the Docker implementation running on the IoT middleware cluster nodes on the same server machine supported by Ubuntu 18.04 fog infrastructure. When a JSON message coming from an IoT node arrives at its corresponding DFMC in the platform, the data fusion is performed over the data and the result is written in a MongoDB database (NoSQL) using Avro (a remote procedure call which implements a pipeline over a source-sink channel framework which compacts the source data into a binary format called BSON) – a binary JSON [17]. MongoDB is a document-oriented database organized in individual rows of BSON documents with no particular schema, residing at the fog infrastructure in this project.

The initial data fusion implementation uses the Chauvenet criterion which is very useful for a measurement series when some particular measurements strikingly differs from the majority and must be rejected [15]. However, the Chauvenet criterion has some limitations on its use as a fusion algorithm because it evaluates whether a point deviates from the average of the values which could cause the elimination of correct values in case the majority of the readings are erroneous. The theory of Dempster Shafer allows a combination of evidence from a variety of sources to achieve a degree of credibility, a more flexible strategy compared to the Bayesian approach which works with probabilities, i.e., the probability that a marked point is wrong or correct [18]. Another way to improve the accuracy of the data in the fusion process is to increase the number of measurements as a function of time. The choice of rejecting data is quite controversial for some cases, but in our case, the measurements of climate data such as air humidity, air temperature, wind vane and speed which normally do not change abruptly, this data fusion strategy can be satisfactory used. To apply the Chauvenet criterion, for every N measurements $x_i, \ldots, x_N$ of a single quantity x, we calculate $\overline{x}$ the sample mean and σ_x the sample standard deviation. If one of the measurements x_{sus} differs so much from $\overline{x}$ then we proceeded with the calculation of t_{sus} as shown in (1), which is the number of how much they differ.

$$t_{\text{sus}} = \frac{|x_{\text{sus}} - \bar{x}|}{\sigma_x} \qquad (1)$$

Next, we calculate the probability *Prob*(outside $t_{sus}\sigma$) by subtracting 100% minus the probability *Prob*(within $t_{sus}\sigma$). The probabilities are calculated using the Gauss distribution for a specific interval. The *Prob*(outside $t_{sus}\sigma$) multiplied by N, the number of measurements results in n, the expected number as deviant as x_{sus}. When n falls below 0.5 then the sample is discarded [15]. At present, data fusion is only available as DFMC for Chauvenet criterion. Other data fusion algorithms will soon be available. The combination of a publish / subscribe protocol with DFMC provides the possibility of using different data fusion algorithms over the same sensor data in different containers, i.e., DFMCs with different data fusion algorithms can be subscribers to the same topic - data readings from a specific IoT node, making it easy to switch from one data fusion strategy to another or even decide which data fusion result is better, based on the same set of sensor readings.

V. Results

The use of microservices containers paired with endpoints at IoT nodes make it possible to perform data fusion in parallel as it would have happened if the data fusion were performed at the IoT nodes but with the advantage of being in a centralized fog infrastructure compared to a distributed network of heterogeneous IoT nodes. Fig. 2 shows the raw data and fusion data in pairs distributed along a number of DFMC´s, showing wind speed measurements from a couple of real micro meteorological stations, combined with data fabricated in other 75 IoT nodes based on ESP32. The purpose of this experiment was to exercise and demonstrate the proper functioning of data fusion implemented as DFMC´s in the IoT platform. Data fusion stored in the NoSQL database are interpreted and displayed via Apache Zeppelin, a platform capable of data analytics and data interpreting from many different sources including MongoDB.

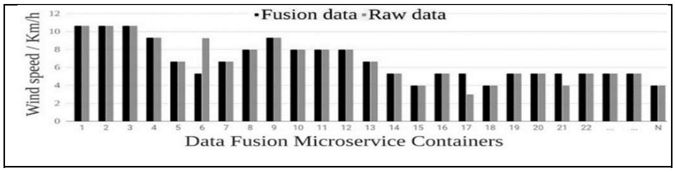

Fig. 2. DFMC - data fusion microservice containers working in parallel.

The resulting data fusion curve from the Chauvenet criterion is shown in Fig. 3 for air humidity over time produced by 4 sensor reads every 5 seconds at the IoT node at a micro meteorological station and sent to a DFCM using Chauvenet with $N = 4$. In this experiment both raw and fusion data were stored in order to compare the raw and fusion curves for air humidity every 5 seconds. We can notice that in nine occasions, air humidity measurements were strikingly out of the mean sensor data as the plotted dots of raw data measurements shows. That demonstrates the effectiveness of the Chauvenet criterion for this specific problem and the usability of the platform implemented.

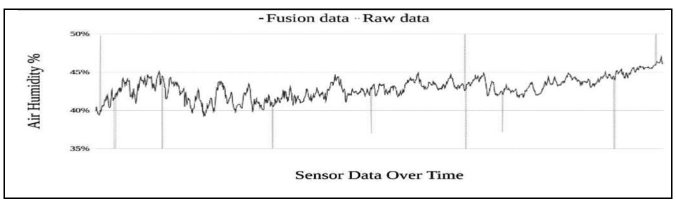

Fig. 3. Fusion data / raw data comparison curves – relative air humidity

Considering that the data fusion can be timely performed at the DFMC, we have then executed a performance test by comparing data fusions being executed in the IoT middleware with a single DFMC against the same load on a single thread data fusion implementing the same algorithm in C and set to run in a raspberry pi 3 b+, a common hardware module used in LoRa gateways, another potential location for performing data fusion in IoT environments. Fig. 4 shows that when the number of data fusion goes beyond 250, the fusion time tends to grow exponentially in the gateway contrasting to the quasi linear growth in the processing time at the DFMC implemented in the platform.

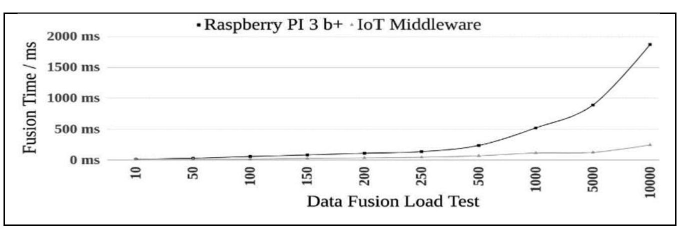

Fig. 4. Data fusion performance – IoT middleware versus IoT gateway

As the communication links play an important part in the message latency from IoT nodes to DFMC, we did compare sensor data messages over ZigBee and LoRa by means of round trip time (RTT), as depicted in Fig. 5. The measurements were done between an IoT node and their gateways, both using radios of 915Mhz and a bandwidth of 250 kbps and messages with 48 bytes long. From 10 up to 200 meters the RTT is roughly the same for both technologies. Going upwards, LoRa becomes 4 to 10 times faster than ZigBee from 350 to 500 meters distance, a zone where LoRa performs better than ZigBee for sensor data.

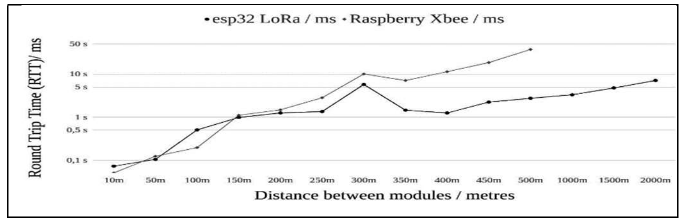

Fig. 5. Comparison of LoRa and ZigBee RTT´s (round trip time)

This experiment, shows that computing power is a hindrance to implement data fusion in gateways or at the edge in the IoT nodes – on the other hand the increment of computing power allocated by the IoT middleware, as the processing demand increases is a key feature for data fusion in large scale IoT environments. The better performance of LoRa can be explained by the absence of TCP/IP header processing time compared to ZigBee which causes an effect on the latency. Also, radio modulation of LoRa, which uses CSS – chirping spread spectrum, contributes to minimize the need for frame retransmission. The peak seen around 300 meters can be explained by obstacles, bushes in this case between the IoT nodes and the gateways which caused diffraction and retransmission. The IoT wireless network will be a matter of further work in the next phases of this project, which intend to investigate the use of AI algorithms for routing frames and stream processing for IoT data fusion using implementations in FPGA at gateways and routers.

VI. CONCLUSIONS

Data fusion is a compulsory task for sensor data in IoT environments. In our experiment the need for data fusion appearred in several sensor reads which clearly had wrong measurements read from the meteorological stations. The use of DFMC in the fog fed by a publish/subscribe mechanism have demonstrated to be a straightforward way for IoT data collection and data fusion. Although the platform with IoT middleware and DFMC were located in a fog computing platform in this project, it was possible to witness the dynamic increase computing power as the influx of IoT sensor data grew showing that the chosen strategy is feasible for heteregeneous IoT environments. Beyond that, the use of DFMC developed in this work provides a flexible IoT data fusion platform which allows the use of different algorithms in new DFMC by simply subscribing to the same topic with the MQTT broker. In the next phases of this work, additional stress tests using other data fusion algorithms and filters will be performed in combination with a fast low latency wireless IoT network in order to potentially offload intelligence from robots to the fog-based IoT platform described in this work.

Acknowledgment

This work has been supported by FAPESP research funding agency under grant number 2017/17226-2.

References

[1] F. Alam et al, “Data fusion and IoT for smart ubiquitous environments: a survey” in IEEE Access, June 2017, pp 9533-9554.

[2] H. Martínez and G. Yannakakis, “Deep multimodal fusion: combining discrete events and continuous signals”, in *Proc. 16th Int. Conf. Multimodal Interact.*, 2014, pp. 34-41.

[3] N. Audebert, B. Le Saux, and S. Lefèvre. (Sep. 2016). “Semantic segmentation of earth observation data using multimodal and multi-scale deep networks”, Available: https://arxiv.org/abs/1609.06846

[4] R. Salustiano and C. Reis Filho, “Signal-level sensor fusion applied to monitoring environment conditions”, Proceedings of the 6th International Caribbean Conference on Devices, Circuits and Systems, Mexico, April 2006, pp 26-28.

[5] M. Kumar, D. P. Garg, and R. A. Zachery, “A method for judicious fusion of inconsistent multiple sensor data”, *IEEE Sensors J.*, vol. 7, no. 5, pp. 723-733, May 2007.

[6] F. Alam, R. Mehmood, I. Katib, and A. Albeshri, “Analysis of eight data mining algorithms for smarter Internet of Things (IoT)”, *Procedia Comput. Sci.*, vol. 98, pp. 437-442, Dec. 2016.

[7] “Overview of the Internet of Things”, Rec. ITU-T Y.2060, 2012.

[8] T.W.Martin and K.Chang, “A data fusion formulation for decentralized estimation predictions under communications uncertainty”, in *Proc. 9th Int. Conf. Inf. Fusion (FUSION)*, 2006, pp. 1-7.

[9] F. Valente and A. C. Neto, “Intelligent steel inventory tracking with IoT/RFID”, in IEEE International conference on RFID technology and application – RFID-TA, Warsaw, Poland, 20-22 Sep 2017, pp. 158-163, http://ieeexplore.ieee.org/document/8098639.

[10] J. Dong, D. Zhuang, Y. Huang, and J. Fu, “Advances in multi-sensor data fusion: algorithms and applications”, *Sensors*, vol. 9, no. 10, pp. 7771_7784, Sep. 2009.

[11] O. Vermesan, P. Friess, P. Guillemin, S. Gusmeroli, H. Sundmaeker, A. Bassi, I. Soler Jubert, M. Mazura, M. Harrison, M. Eisenhauer, and P. Doody, “Internet of things strategic research roadmap”, 2009.

[12] K. Chang and C. Mori, “Analytical and computational evaluation fo scalable distributed fusion algorithms”, IEEE Transactions on Aerospace and Electronic Systems, vol. 46, n.4, Oct. 2010.

[13] R. Carrião, H. Rozenfeld and F. Valente, “Requirements for a meta-model to represent product-service systems (PSS) that incorporate Internet of Things (IoT) solutions”, 11º Brazilian Congress for Innovation and Product Development Management, 2017, São Paulo. Blucher Design Proceedings, 2017. p. 866.

[14] A. Torres, J. Adriano Filho, A. Rocha, R. Gondim and J. Souza, “Outlier detection methods and sensor data fusion for precision agriculture”, 9º SBCUP – Brazilian Symposium of Ubiquitous and Pervasive Computing, São Paulo, SP, Brazil, 2017.

[15] J. R. Taylor, “An introduction to error analysis: the study of uncertainties in physical measurements”, University Science Books, 2nd Auflage , ISBN 0935702423.9780935702422.

[16] “Kaa Iot middleware”, www.kaaproject.org

[17] http://avro.apache.org

[18] N. Nesa and I. Banerjee, “IoT-Based Sensor Data Fusion for Occupancy Sensing Using Dempster-Shafer Evidence Theory for Smart Buildings” in IEEE Internet of Things Journal, vol. 4, no. 5, October 2017.