Real-Time Big Data Analytics in Smart Cities from LoRa-Based IoT Networks

Antonio M. Fernández, David Gutiérrez-Avilés, Alicia Troncoso, and Francisco Martínez-Álvarez(✉)

Data Science & Big Data Lab, Pablo de Olavide University, ES-41013 Seville, Spain
fmaralv@upo.es

Abstract. The currently burst of the Internet of Things (IoT) tech-nologies implies the emergence of new lines of investigation regarding not only to hardware and protocols but also to new methods of pro-duced data analysis satisfying the IoT environment constraints: a real-time and a big data approach. The Real-time restriction is about the continuous generation of data provided by the endpoints connected to an IoT network; due to the connection and scaling capabilities of an IoT network, the amount of data to process is so high that Big data tech-niques become essential. In this article, we present a system consisting of two main modules. In one hand, the infrastructure, a complete LoRa based network designed, tested and deployment in the Pablo de Olavide University and, on the other side, the analytics, a big data streaming sys-tem that processes the inputs produced by the network to obtain useful, valid and hidden information.

Keywords: IoT · LoRaWAN · Real-time · Big data · Data streaming

1 Introduction

The current technological reality points to two main lines of research and development. First, the line of industry and services wherein the rise of the advanced technology to M2M communications or the Internet of Things (from now on, IoT) [21] is changing our means of production and our service management systems. This fact leads our society to a new industrial revolution, the 4.0 industry [12]. On the other side, the line of data science and big data [2] emerges as a consequence of the vast amount of it generating, day by day, in our society.

There is an intimal relation between IoT technology and the data science and big data. The IoT networks can potentially manage a massive amount of data depending on the number of endpoints connected to it. Although the management of the data traffic of an IoT network is a critical element, the useful and efficient treatment of this data is another crucial point to take into account. Due to the huge amount of data involved in this new framework, issues like data storage, data buffering appears implying the use of Big data solutions. Furthermore, Data science techniques are needed to analyze the real-time data of the IoT network and obtain useful, valid and hidden information [14].

In this article, we present the an IoT agent system consisting in an in-production LoRa based IoT network deployed, whose usability has been tested in the Pablo Olavide University (Seville, Spain) and a big data streaming system, based on HDFS and Spark, that analyzes in real time the data provided by the earlier mentioned IoT network.

The rest of the article is structured as follows. A summary of the previous researches related to the paper's topic is presented in Sect. 2. The architecture of the proposed system and the methods used in the development can be found in Sect. 3. The experimental setup carried out, and the yielded results are reported in Sect. 4. Finally, the conclusions and future work are provided in Sect. 5.

2 Related Works

Nowadays, GPRS, Sigfox, Narrowband Internet of Things (NB-IoT), and LoRa are four widely used IoT technologies providing the best coverage for IoT devices. These technologies have been deeply studied in terms of coverage in [13]. The authors simulated the coverage of the previously mentioned IoT networks comparing them in an area of 7800 km^2. This study aimed to obtain the technology that provides better coverage for the connected IoT devices concluding that NB-IoT provided a better coverage but having a maximum signal coupling and a signal loss of 164 dB.

By contrast, the authors in [17] demonstrated the low-consumption devices connected to LoRaWAN based IoT networks could transmit the data more efficiently compared with other devices and network servers. The authors carried out field testing with line of sight and no line of sight in an Indonesian University campus.

An architectural study was carried out in [18]. Here, the authors analyzed the LoRa technology and demonstrated the LoRaWAN based network architectures shows a good match with the measurement systems. Furthermore, their experimental results confirmed the capability of a low-cost transceiver to schedule the transmission of frames with a standard uncertainty less than 3 µs.

Related to the data, we can distinguish between two concepts. On the one hand, we find the discovery or extraction of valid, useful and hidden information from data sets; that discipline is known as data mining, the main step within the knowledge discovering in databases process. The Data Mining covers the data source selection processes, pre-processing, and the application of machine learning algorithms providing us of descriptor, predictor or classifying models of the data. These models extract those mentioned above valid, useful and hidden information that implies a huge number of applications like variable predictions, client segmentation or fraud detection [8], among others.

On the other hand, we find the processing and management of vast volumes of data form a new state-of-the-art discipline called big data [14]. The big data applies to an extensive collection of fields. In [5] a big data environment is developed to electricity market field, managing volumes of 1 TB of data with the aim of detect fraudulent clients. The authors in [6] present a software tool for

behavior pattern discovery in vast amounts of biological data and they develop in [7] a methodology to process and evaluate the results of the previous tool that they are equally big.

In this paper we present a new aspect into the big data and the data mining: the analysis of significant flows of data, coming from IoT sensors of a LoRaWAN network, in real time with auto-incremental machine learning algorithms. This field is called big data streaming. There are a few works related with this new approach; however, in [11], the authors propose a comparison between the main frameworks to work in streaming context these are Storm, Flink y Spark Streaming; the parameters of the study are performance and failure tolerance. The authors in [1] conduct an analysis of the performance of the linear regression algorithm with the Spark MLlib library and the Massive Online Analysis platform (MOA). Finally, in [16], a real-time methodology to detect cybercrimes and credit card fraud based on Spark Streaming is presented.

3 System Architecture

The proposed architecture is illustrated in Fig. 1. We can observe that the system is composed of several modules. Each of them is responsible for a task in the system and are connected by input/output links. In a general way, the LoRa based infrastructure is in charge of obtaining and manage the data form several IoT devices. The preprocessing module takes these data as input and decodes and prepares to the following subsystems. The real-time or experimental environment manages the data streams and feed to the big data streaming engine. This engine performs the training of an auto-incremental machine learning algorithm and makes predictions of the measured variables by the sensors.

Next, each module is separately analyzed in the following sections: the LoRa-based infrastructure in Sect. 3.1, the JSON payload buffering and preprocessing in Sect. 3.2 and the real-time environment in Sect. 3.3.

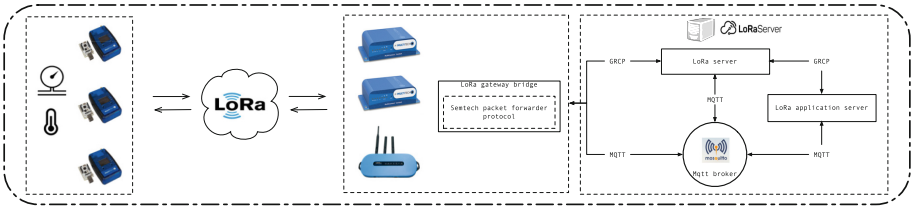

Fig. 1. Proposed IoT architecture.

3.1 LoRa Based Infrastructure

We present our LoRa based infrastructure whose function is to collect the data that will be processed and analyzed by the real-time big data analytics system.

The inputs of the LoRa infrastructure will be de data picked by IoT sensors; specifically, these devices will measure values of two variables, this is pressure and temperature. The outputs of this subsystem will be the raw metrics of pressure and temperature registered in the *LoRaServer* software.

In Fig. 1 we can observe the elements of this module. Firstly, we have three IoT devices with two sensors measuring temperature (in Celsius, C) and pressure (in kilopascals, kPa).

The next element is the LoRa network based on the LoRaWAN standard. LoRaWAN defines an IoT communication protocol and a system architecture for the deployment of a network. The protocols and the architecture of a network are the most influencer elements to determine the life of the batteries, the net capability, the quality of the service and the security [17]. We have chosen the LoRaWAN standard for two main reasons: LoRaWAN is an open standard, it offers excellent performance for our purpose, and there are several open-source software solutions to net managing (*LoRaServer* software).

Then, we can see the group of three LoRa gateways. These devices carry out the management of the access points to the network and, thanks to a piece of software called LoRa Gateway Bridge, transmit the collected data to the controller software. As a part of this transmission, the binary packets from the sensors are transformed into a JSON that can managed by for LoRaServer system. The protocol between LoRa gateways and LoraServer is called Semtech packet forwarder protocol and is included in the LoRa Gateway Bridge.

The last part of the LoRa infrastructure is the LoRaServer system that manages and controls every payload that travels for the IoT net. Three subsystems compose the LoRaServer system: the *LoRa server* module responsible for the management of the gateways and end points. Next, we found the *LoRa App Server* that manage the applications, users, services and devices. Finally, the core of the system is the MQTT broker Eclipse Mosquito that connects the LoRa server with the LoRa Gateway Bridge and the LoRa App Server with any application.

3.2 JSON Payload Buffering and Preprocessing

The inputs of this module will be the LoRa JSON payloads acquired by a MQTT consumer. The outputs will be Kafka producers. Apache Kafka is an open-source software developed and maintained by Apache Software Foundation. The main objective of this software is to provide a unified platform with high performance and low latency for the manipulation of data sources in real-time environments. It can be interpreted like a message queue, developed as a register of distributed transactions using the publisher-subscription pattern. Apache Kafka is a large scale scalable, partitioned and replicated platform [3,10,15].

The general function of this module is to receive the JSON payloads from the LoRa network, then performing a JSON parsing to extract the *data* field. This information is encrypted by the Base64 method; then, the next step will be decrypting the data for, afterward, carry out device-driven decoding. After these

steps, we will have the measures of pressure and temperature in a legible form (in kPa and C respectively).

Finally, the module checks the environment of the system. For this particular proposal, a real-time environment is used. Each W-dependent instance is sent to its corresponding module via Kafka producer.

3.3 Real-Time Environment

This module carries out the set up of the environment needed for training and testing the auto-incremental machine learning algorithm. In this module a real-time environment is settled; therefore, every time that a new instance with a W learning window is building, it will arrive at it using a Kafka consumer channel.

The elements of this module can be observed in Fig. 2. There, we can see how the Kafka consumer channel forwards the instances to a software layer called Kappa architecture [14]. It is a software architecture pattern whose purpose is to process streams of data in real-time and to store the results as mentioned earlier.

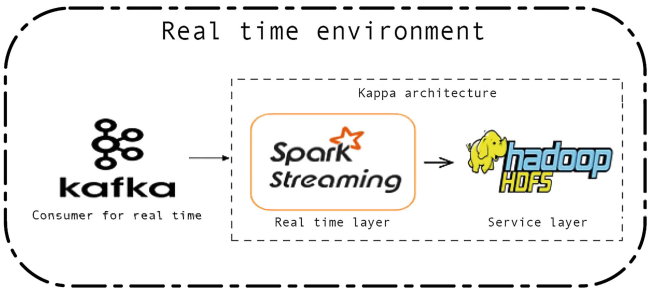

Fig. 2. Real-time big data analytics environment.

The Kappa architecture has two different elements. On the one hand, the real-time layer, implemented by Apache Spark Streaming is in charge of processing the instances coming from the Kafka channel and adapts the data for machine learning algorithm training and testing. On the other hand, we found the service layer, implemented by Apache Hadoop Distributed File System (HDFS) file system [19] which is in charge of storing the results of machine learning processing.

At this point, a brief explanation of Spark Streaming and Apache Hadoop HDFS is given:

– *Apache Spark Streaming*: It is an API of Apache Spark whose purpose is enabling scalable, high-throughput, fault-tolerant stream processing of live data streams. The streams of data can come from multiple sources (Apache Kafka, in our case). The processing results can be stored, likewise, in multiples systems like databases, dashboards, and, (in our case) HDFS.

– *Apache Hadoop HDFS*: It is one of the modules of Hadoop. HDFS is a distributed file system designed to run on commodity hardware. Furthermore, it is highly fault-tolerant and is designed to be deployed on low-cost hardware and provides high throughput access to application data and is suitable for applications that have large datasets.

3.4 Big Data Streaming Engine

This section describes the engine used in order to forecast big data streaming.

Data are received continuously, and therefore the prediction of a given time horizon, h, must be made in real time. The Apache Spark Streaming machine learning library (MLlib) is proposed for this purpose. In particular, the linear regression algorithm [9] is used to forecast future values for the target variables (temperature and pressure in our case).

MLlib's linear regression algorithm does not support multi-step forecasting and, for this reason, the algorithm must be adapted as previously done in [4,20]. Therefore, the problem that must be solved is:

$$[x(t+1), x(t+2), \ldots, x(t+h)] = f(x(t), x(t-1), \ldots, x(t-w-1)) \quad (1)$$

where w represents the window of past values considered for predicting the h future values.

For each data stream received from any sensor at the instant t, a forecasting model M_t is generated at the training stage. This model is incrementally updated along with new coming data, thus generating new models at different time stamps M_{t+1}, M_{t+2},...,M_{t+n}. When a prediction is made, the last generated model is used.

4 Results

This section describes the experimentation that has been carried out for the extraction of information with the input data and application of a big data streaming algorithm.

4.1 Dataset Construction and Linear Regression Parametrization

The data used are those obtained by the sensors of the LoRaWAN network. The measurement of the data has been performed every ten seconds for a period of one month, having a raw data set of approximately seventy thousand records.

The dataset contained erroneous records, so a pre-processing of the data was applied, checking that the data had a correct measurement and eliminating the erroneous measurements. At the end of the cleaning pre-processing, five thousand seven hundred and twenty-six records are obtained.

The linear regression algorithm with gradient descent [9], used for experimental study in streaming, requires the optimization of two main parameters as indicated in the previous section:

– α. It is the size of the step that moves the gradient in the downward direction.
– σ. The number of iterations necessary for the method to converge.

To obtain the optimum value of these two parameters, an exhaustive search algorithm has been developed. Finally, the optimum parameters obtained are shown in Table 1, when the mean relative error (MRE) is minimized.

Table 1. Optimum parameters obtained for the linear regression algorithm.

Data	σ	α (stepSize)	MRE
Pressure	10	3.33E-11	1.27E-05
Temperature	15	3.61E-05	4.18E-05

4.2 Experimental Setup

After analyzing the input data, building the datasets, and obtaining the optimal execution parameters for the linear regression model, the following experimental design is proposed to test the effectiveness of the Apache Spark streaming linear regression model in a real streaming situation.

The aim of this study is to test the MRE variation when the self-incremental learning model is fed with new learning data through the streaming channel to which it is connected.

Thus, for each variable measured $V = \{P, T\}$ by the pressure and temperature sensor and for each time window $w_i = \{w_3, w_6, w_{12}, w_{24}, w_{90}, w_{180}\}$ the following process is performed:

1. Activate the streaming system.
2. Parametrize the incremental Linear regression algorithm with the optimal parameters $\langle \alpha, \sigma \rangle$ for V and w_i.
3. Inject training dataset into the training channel with a time lapse of 5 s, generating a M_i model for each dataset.
4. For each M_i model a prediction of the complete test set is made, measuring the associated MRE.

Once this process is executed, the MRE is obtained for each model generated $M_1, M_2, M_3, ..., M_{20}$. At the same time, the observed values will be compared with the predicted values, highlighting the models of the streaming channel that reaches lower MRE in the prediction.

4.3 Analysis

In this section different experiments are carried out in order to perform an exhaustive analysis of the incremental learning of the different online models that are generated to estimate the predictions in a streaming environment. First,

the section on errors evaluates the quality of the prediction, in terms of the average relative error, while adding LP to the training package used to obtain the prediction model. Finally, the predictions section presents the results of the prediction of the set of tests obtained with the best model for each time window. To obtain the online models, the training data set has been divided into 30 batches. Each of these subsets will be injected one at a time at 5 s intervals. In this way, 30 prediction models will be obtained in real time in an incremental way to predict the test set. These models are obtained using the optimal configuration of the parameters α and σ, necessary for linear regression, which is shown in Table 1.

Table 2. MRE for pressure sensor data stream.

	w_3	w_6	w_{12}	w_{24}	w_{90}	w_{180}
M_1	5.08E-03	9.25E-03	5.76E-03	5.74E-03	5.80E-03	**4.48E-03**
M_2	4.95E-03	4.54E-03	4.55E-03	4.49E-03	4.44E-03	4.52E-03
M_3	4.89E-03	4.66E-03	4.65E-03	4.31E-03	4.70E-03	4.52E-03
M_4	4.99E-03	4.59E-03	4.40E-03	4.56E-03	4.44E-03	6.29E-03
M_5	4.89E-03	**4.51E-03**	4.63E-03	**4.31E-03**	4.65E-03	4.92E-03
M_6	4.96E-03	4.52E-03	4.38E-03	4.60E-03	**4.42E-03**	5.13E-03
M_7	4.98E-03	4.52E-03	4.58E-03	4.32E-03	4.49E-03	4.73E-03
M_8	4.89E-03	4.53E-03	4.40E-03	4.48E-03	4.81E-03	5.31E-03
M_9	4.96E-03	4.54E-03	4.36E-03	4.61E-03	4.42E-03	5.40E-03
M_{10}	4.90E-03	4.69E-03	4.56E-03	4.32E-03	4.53E-03	4.72E-03
M_{11}	**4.88E-03**	4.68E-03	4.65E-03	4.52E-03	5.41E-03	5.13E-03
M_{12}	4.98E-03	4.67E-03	4.39E-03	4.33E-03	4.59E-03	5.21E-03
M_{13}	4.90E-03	4.64E-03	**4.36E-03**	4.31E-03	5.17E-03	4.73E-03
M_{14}	4.89E-03	4.64E-03	4.60E-03	4.55E-03	4.42E-03	5.25E-03
M_{15}	4.99E-03	4.54E-03	4.70E-03	4.34E-03	4.46E-03	4.76E-03
M_{16}	4.89E-03	4.54E-03	4.37E-03	4.31E-03	4.50E-03	5.44E-03
M_{17}	4.96E-03	4.55E-03	4.59E-03	4.49E-03	5.23E-03	4.71E-03
M_{18}	5.00E-03	4.62E-03	4.37E-03	4.32E-03	4.59E-03	5.88E-03
M_{19}	4.90E-03	4.74E-03	4.56E-03	4.49E-03	4.98E-03	4.75E-03
M_{20}	4.89E-03	4.53E-03	4.63E-03	4.32E-03	4.50E-03	5.26E-03

The errors made in the prediction of the test set when using the different models that are generated in an incremental way are now discussed. Table 2 shows the MRE made when predicting the test set for the pressure sensor, using each of the models obtained online for different lengths of historical data. In bold type, the models that have obtained the minimum error for each window are highlighted. Similar results are reported for both pressure and temperature but, due to space limitations, they are not shown here.

5 Conclusions

In this article we propose a complete system to collect information from the environment through the use of a IoT architecture using LoRa technology and the LoRaWAN standard. The system has been successfully deployed at Pablo de Olavide University (Seville, Spain). The implementation of an architecture for data analysis, called Kappa architecture, for real-time analysis and development of experimentation is also described, and it is based on the underlying pache Spark Streaming and HDFS technologies. Experiments carried out using the physical IoT network and real sensors are reported in order to evaluate the incremental models and the quality of the predictions made.

Acknowledgments. We would like to thank the Spanish Ministry of Economy and Competitiveness for the support under project TIN2017-88209-C2-1-R. Additionally, we want to express our gratitude to Enrique Parrilla, Lantia IoT's CEO, since all the equipment has been provided by him. The T-Systems Iberia company is also acknowledged since all experiments have been carried out on its Open Telekom Cloud Platform based on the OpenStack open source.

References

1. Akgün, B., Öğüdücü, S.G.: Streaming linear regression on Spark MLlib and MOA. In: Proceedings of the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, pp. 1244–1247 (2015)
2. Chen, M., Mao, S., Liu, Y.: Big data: a survey. Mob. Netw. Appl. **19**(2), 171–209 (2014)
3. D'Silva, G.M., Khan, A., Gaurav, Bari, S.: Real-time processing of IoT events with historic data using Apache Kafka and Apache Spark with dashing framework. In: Proceedings of the IEEE International Conference on Recent Trends in Electronics, Information Communication Technology, pp. 1804–1809 (2017)
4. Galicia, A., Talavera-Llames, R., Troncoso, A., Koprinska, I., Martínez-Álvarez, F.: Multi-step forecasting for big data time series based on ensemble learning. Knowl. Based-Syst. **163**, 830–841 (2019)
5. Gutiérrez-Avilés, D., Fábregas, J.A., Tejedor, J., Martínez-Álvarez, F., Troncoso, A., Arcos, A., Riquelme, J.C.: SmartFD: a real big data application for electrical fraud detection. Lect. Notes Artif. Intell. **10870**, 120–130 (2018)
6. Gutiérrez-Avilés, D., Rubio-Escudero, C., Martínez-Álvarez, F., Riquelme, J.: Trigen: a genetic algorithm to mine triclusters in temporal gene expression data. Neurocomputing **132**, 42–53 (2014)
7. Gutiérrez-Avilés, D., Giráldez, R., Gil-Cumbreras, F.J., Rubio-Escudero, C.: TRIQ: a new method to evaluate triclusters. BioData Min. **11**, id15 (2018)
8. Han, J., Pei, J., Kamber, M.: Data Mining: Concepts and Techniques. Elsevier, Amsterdam (2011)
9. Hu, T., Wu, Q., Zhou, D.X.: Convergence of gradient descent for minimum error entropy principle in linear regression. IEEE Trans. Signal Process. **64**(24), 6571–6579 (2016)
10. Ichinose, A., Takefusa, A., Nakada, H., Oguchi, M.: A study of a video analysis framework using Kafka and Spark Streaming. In: Proceedings of the IEEE International Conference on Big Data, pp. 2396–2401 (2017)

11. Karakaya, Z., Yazici, A., Alayyoub, M.: A comparison of stream processing frameworks. In: Proceedings of the International Conference on Computer and Applications, pp. 1–12 (2017)
12. Lasi, H., Fettke, P., Kemper, H.G., Feld, T., Hoffmann, M.: Industry 4.0. Bus. Inf. Syst. Eng. **6**(4), 239–242 (2014)
13. Lauridsen, M., Nguyen, H., Vejlgaard, B., Kovacs, I.Z., Mogensen, P., Sorensen, M.: Coverage Comparison of GPRS, NB-IoT, LoRa, and SigFox in a 7800 km^2 Area. In: Proceedings of the IEEE Vehicular Technology Conference, pp. 1–5 (2017)
14. Marz, N., Warren, J.: Big Data: Principles and Best Practices of Scalable Real-time Data Systems. Manning Publications Co., Shelter Island (2015)
15. Noac'h, P.L., Costan, A., Bougé, L.: A performance evaluation of Apache Kafka in support of big data streaming applications. In: Proceedings of the IEEE International Conference on Big Data, pp. 4803–4806 (2017)
16. Pallaprolu, S.C., Sankineni, R., Thevar, M., Karabatis, G., Wang, J.: Zero-day attack identification in streaming data using semantics and Spark. In: Proceedings of the IEEE International Congress on Big Data, pp. 121–128 (2017)
17. Rahman, A., Suryanegara, M.: The development of IoT LoRa: a performance evaluation on LoS and Non-LoS environment at 915 MHz ISM frequency. In: Proceedings of the International Conference on Signals and Systems, pp. 163–167 (2017)
18. Rizzi, M., Ferrari, P., Flammini, A., Sisinni, E.: Evaluation of the IoT LoRaWAN solution for distributed measurement applications. IEEE Trans. Instrum. Meas. **66**(12), 3340–3349 (2017)
19. Shvachko, K., Kuang, H., Radia, S., Chansler, R.: The Hadoop distributed file system. In: Proceedings of the IEEE Symposium on Mass Storage Systems and Technologies, pp. 1–10 (2010)
20. Torres, J.F., Galicia, A., Troncoso, A., Martínez-Álvarez, F.: A scalable approach based on deep learning for big data time series forecasting. Integr. Comput.-Aided Eng. **25**(4), 335–348 (2018)
21. Wortmann, F., Flüchter, K.: Internet of things. Bus. Inf. Syst. Eng. **57**(3), 221–224 (2015)