2021 28th International Conference on Telecommunications (ICT)

Spreading Factor Selection Mechanism for Transmission over LoRa Networks

1st Christos Bouras
Computer Engineering and Informatics
University of Patras
Patras, Greece
bouras@cti.gr

2nd Apostolos Gkamas
University Ecclesiastical Academy of Vella
Ioannina, Greece
gkamas@aeavellas.gr

3th Spyridon Aniceto Katsampiris Salgado
Computer Engineering and Informatics
University of Patras
Patras, Greece
ksalgado@ceid.upatras.gr

4th Nikolaos Papachristos
Computer Engineering and Informatics
University of Patras
Patras, Greece
papachristosn@upatras.gr

***Abstract*— This paper presents a mechanism for Spreading Factor (SF) prediction in LoRa networks for more optimized data transmissions. The proposed mechanism is based on Machine Learning (ML) algorithms and assigns the node's SF value based on prior transmission data. This paper examines three different approaches for the selection of the SF during LoRa transmissions a) Random SF assignment b) Adaptive Data Rate (ADR) and c) ML based SF selection. The main target is to study and determine the most efficient approach, as well as to investigate the exploitation of ML techniques in the context of LoRa networks. We created a simple library based on ML libraries, such as Scikit Learn that can be used with the FLoRa an OMNeT++ based LoRa simulator. With the use of this library, it is possible to predict the node's SF using ML techniques. Two classification algorithms were tested, the k Nearest Neighbors (k-NN) and Naïve Bayes classifier. Finally, we compared the ML mechanisms with two variants of the ADR mechanism. The approaches performance is evaluated for different scenarios, using the delivery ratio and energy consumption metrics.**

Keywords*— *LoRa; Machine Learning; Python; LPWAN;

I. INTRODUCTION

Low Power Wide Area Networks (LPWAN) come to solve the problem of long-distance transmission, with very low energy consumption. Some examples of LPWAN networks are LoRa, Narrowband IoT (NB-IoT), and Sigfox [1]. Among these technologies, LoRa has gained wide popularity with many applications. However, LoRa has many challenges in terms of network management, as the number of nodes is increasing. One of these challenges that have been studied by researchers is energy consumption.

In LoRa networks many parameters affect energy consumption such as the Transmission Power (TP), the Coding Rate (CR), and the Spreading Factor (SF). The SF parameter is one crucial parameter that affects both the energy consumption and the delivery ratio of the network, as the signal collisions are highly correlated with the SF assignment. SF determines on how many chirps, the carrier of the data, are sent per second. The relationship between transmission rate and SF assignment has been thoroughly studied in previous research papers ([2], [3] and [4]). Particularly, research work [2] studies the different SF assignment to verify the theoretical limits obtaining the practical performance profile of the LoRa radio. On the other hand, paper [3] studies SF assignment in rural areas to determine the effect on the coverage of the mobile network. In a LoRa network, initially, a node is not aware of how far away it is from a gateway, in order to adjust the SF according to the distance. Authors in [4] attempt to off-load the data traffic into several subnets by utilizing multiple-access dimension based on multi hop LoRa network. This is achieved by enabling packet transmissions in parallel with multiple SFs to become feasible.

Also, some papers are based on Machine Learning (ML). Energy efficiency is the key requirement to maximize sensor device lifetime in terms of location estimation on LoRa networks [5] [6]. In order to estimate nodes' location, a Global Positioning System (GPS) module can be incorporated, but such a solution can be energy-consuming as GPS requirements demand large amounts of energy. An example can be found in [7] where the authors propose a search and rescue application based on LoRa in which the localization process is based only on LoRa. The proposed system in [7] without using the GPS module, the battery life of the wearable can last significantly longer than the LoRa-GPS system one described in [8]. On the other hand, Received Signal Strength Indicator (RSSI) in LoRa localization algorithms suffer from the fact that it cannot be used for different places. This is because RSSI values are correlated with the topology where the measurements were conducted, and in the case of Time Difference of Arrival (TDoA), multipath can deteriorate the localization accuracy. Moreover, gateways that have this feature also have high cost.

Following the above study, we present the development of a library in order to enable the communication between the FLoRa simulator and the sci-kit-learn library, in order to use ML techniques for the SF assignment. Also, we formulated the process of SF assignment as a classification problem, with features of the energy consumption per packet and the TP. The choice of these features has been made in order to avoid using the nodes' location. Using the above-mentioned library, two variants of the proposed mechanism were created based on the k-NN algorithm and the Naïve Bayes classifier. Finally, we present a comparative evaluation against the Adaptive Data Rate (ADR) mechanism and the random initialization of the SF. The comparative evaluation was based on delivery ratio and the energy consumption metrics, so as to study the energy consumption, and the trade-off with the delivery ratio. Finally, the comparison conducted in paper [9] was important for our choice to use FLoRa simulator.

The rest of the paper is organized as follows: the next section describes the System Model. Section 3 presents the proposed mechanism and Section 4 presents the mechanisms by which the proposed mechanism will be compared. In Section 5 the results of the evaluation are described. Finally, Section 6 presents the conclusion and the future work.

2021 28th International Conference on Telecommunications (ICT) | 978-1-6654-1376-3/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICT52184.2021.9511509

978-1-6654-1376-3/21/$31.00 ©2021 IEEE

Authorized licensed use limited to: Southeast University. Downloaded on February 07,2023 at 03:03:55 UTC from IEEE Xplore. Restrictions apply.

II. System Model

A LoRa network uses SF to specifically set the data transfer rate relative to the range. Chirps are used to encode data in LoRa networks on the transmitter (Tx) side, while inverse chirps are used on the receiver (Rx) side for signal decoding. SF indicates how many chirps are used per second, and define bit rate, per symbol radiated power and achievable communication range. Specifically, the chips per symbol is defined as: 2^{SF}.

Higher SF provides increased processing gain and higher reception sensitivity. For example, SF9 is 4 times slower than SF7 in terms of bit rates. The data rate is lower at higher SF, but the communication range is higher. The scalability of LoRa is achieved by the rational SF assignment. The slower the bit rate, the higher the energy per data set and the higher the range [10]. LoRa supports a heuristic selection of the SF factors depending on the link budget of the latest 20 uplink frames, the so-called ADR.

The transmission of a packet is assumed successful when the power of the received signal is higher than the sensitivity of the receiver. The received power in the simulation is expressed in Eq.1 and derives from paper [15], where P_{rx} is the received power, the GL is the General Gains and losses and the PL(d) is the path loss model.

$$P_{rx} = P_{tx} + G_L - PL(d) \tag{1}$$

The path loss model follows Eq. 2 ([15]):

$$PL(d) = PL(d_0) + 10nlog_{10}\left(\frac{d}{d_0}\right) + X_\sigma \tag{2}$$

where $PL(d_0)$ is the mean loss for a reference distance d_0, the n is the path loss exponential and the $X\sigma$ is a random variable following a zero mean gaussian distribution, playing the role of noise. Also, the sensitivity threshold of the radio receiver is described in Eq. 3: ([15]):

$$S = -174 + 10\log_{10} BW + Y + SNR \tag{3}$$

BW refers to bandwidth, where Y is a constant value representing the receiver's noise figure and depends on the hardware implementation that may vary and SNR is related to the Signal-to-Noise ratio.

Apart from the above, orthogonality is taken into consideration. The orthogonality assumes that LoRa GWs successfully receive signals with different SFs from the end-nodes, transmitted in the same time. Furthermore, signals (with larger TP value) that have the same SF value in a simultaneous transmission, can be received by the GW. Finally, regarding the energy consumption, the values of the voltage and current depend on the SX 1272[1] transceiver made by Semtech. This transceiver has three states: a) transmit b) sleep c) receiving. The supply voltage used in our scenarios was 3.3 V. Finally, in Fig. 1 the architecture of the studied system is presented. The architecture consists of the nodes that are transmitting using LoRa technology and the gateway that is an intermediate node that relays the data transmitted by the nodes to the NS. The NS is a centralized entity that is responsible for the network parameters and allows the access of the nodes to Application Servers (AS). In our tested case, the AS is that it takes as an input the LoRa parameters from the NS and gives as an output the SF to be assigned to the respective node.

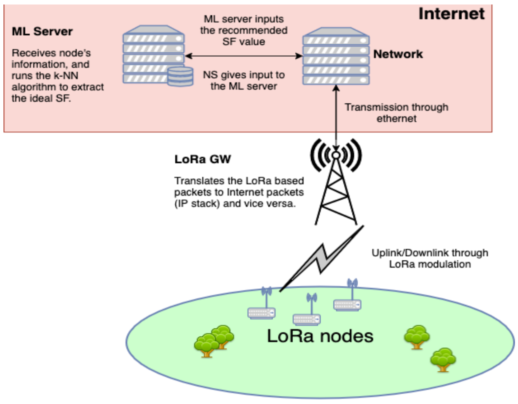

Fig. 1. SF selection architecture through ML

III. Proposed Approach

Before proceeding to the phases of the ML-based mechanism, it is important to formulate the problem of the SF assignment as ML problem and describe the ML algorithms used. Firstly, SF can be considered as a multiclass classification problem, and each value of the SF can be assumed as a different class. Thus, in this case, there are 6 classes as many as the SF values, with the label ranging from 7 to 12.

Secondly, the ML algorithms used are the k-NN and Naïve Bayes classifiers. K-NN algorithm takes into consideration the fact that similar data points are in proximity to each other. So, the way that the input points are classified is the following: for an input, point, let be x assign this point to the class that its k most similar points of the training dataset have. The distance can be the Euclidean, Minkwoski, Manhattan, Mahalanobis, and Chebysev distance. In this paper, the Euclidean distance was used. One of the main advantages of the k-NN algorithm is the fact that it is very easy to implement, but when the number of features is large, then the k-NN tends to perform poorly. As in this paper, the feature space is small and the k-NN algorithm is a good candidate.

The second ML algorithm examined is the Naïve Bayes classifier. Naïve Bayes classifiers refer to a class of probabilistic classifiers. The Naïve Bayes classification algorithm applies the Bayes theorem to the input point, with the "naïve" assumption of independence between the features of the input point. There are some variants of the Naïve Bayes classifier, such as the Gaussian Naïve Bayes and Bernoulli Naïve Bayes variant. In this work, the Gaussian variant was used as it was more accurate. The main advantage of this algorithm is that it can achieve high accuracy without the need of large datasets. This can be useful in real life deployments, as it can be used without accumulating a large amount of data. Especially, in LoRa applications that the nodes do not transmit frequently and the accumulation of a large amount of data can be time consuming, the use of Naïve Bayes can be beneficial. As explained, both k-NN and Naïve Bayes can be good candidates to be used in LoRa applications.

A. ML algorithms training phase

Firstly, the simulation executed with the ADR mechanism enabled. The data created were used for the training phase. Due to the fact that the SF allocation through ADR creates an

[1] [1]https://www.semtech.com/products/wireless-rf/lora-transceivers/sx1272

Authorized licensed use limited to: Southeast University. Downloaded on February 07,2023 at 03:03:55 UTC from IEEE Xplore. Restrictions apply.

imbalanced dataset, it was necessary to create synthetic data, according to the SMOTE-NC technique [11]. This helps us to reduce the bias against some SFs values. During the training phase it is very important to extract the key factors of the data in order to train the dataset. In the feature selection phase, using chi-squared analysis, the total energy divided by the total packets sent and the TP were selected as the features. It is worth noting that node's position was excluded and not considered as a potential feature as the localization in LoRa networks cannot always provide accurate positioning. In Fig. 2, the created dataset is presented. where the x axis represents the Energy/Packet and the y axis the TP value. Each color represents a value of the SF. In Fig. 2 the SF classes can be quite separatable, and these two features can classify the SF.

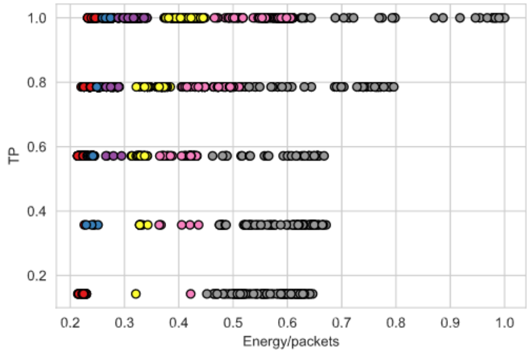

Fig. 2. Dataset Visualization

Moreover, the data were normalized and then the choice of the k was investigated. Using 10-fold cross validation, for the k-NN with k ranging from 2 to 50, it was concluded that with an average accuracy of 96% in the 10-fold cross validation, k=4 seems to be the most suitable value. Moreover, the Naïve Bayes classification algorithm was used, in order to compare the results of the ML mechanism with two classifiers. From the different variations of the Naïve Bayes algorithm, the Gaussian Naïve Bayes was chosen, as it gave better results in terms of accuracy. The difference between the Gaussian and the other variants of Naïve Bayes was huge, as for example the Multinomial variant achieved an accuracy as low as 30%.

Finally, in the testing dataset, the accuracy, precision, recall and the F1 Score were used as metrics for the ML algorithm evaluation. The accuracy refers to the ratio of correct predictions to the total predictions. The precision metric refers to the ratio of the correctly predicted answers of a class to the total number of the answers that predicted this class. The recall metric is the ratio of the number of correctly predicted answers to the number of the actual instances of the class. F1 refers to the relative contribution of the precision and recall. As Table1 1 shows, the k-NN classifier scored highly in all metrics. In addition, the "curse of dimensionality" is reduced because the features used in this work are only two. The Gaussian Naïve Bayes classifier scored very well but lower than the k-NN classifier. Next, the steps that the ML-based mechanism follows are presented.

TABLE 1 METRIC SCORES

Metric	k-NN	Naïve Bayes
Accuracy	0.9692	0.8547
Precision	0.9694	0.8678
Recall	0.9696	0.8549
F1	0.9695	0.8571

B. Step 1: Export NS values to ML server

In order to run our ML algorithms a series of information related to our simulation is required. The above information in the simulation framework is collected and stored in an external file as test dataset, able to be used as input to the ML server. The stored information is the transmission power, the packets sent as well as the energy consumed.

C. Step 2: SF Selection

In order to extract the ideal SF for the transmission of the data to a single node, firstly the stored data must be retrieved and analyzed. For this reason, through ML we try to extract (based on training dataset) the ideal SF that could be used from NS for the transmission of the data using the k-NN and the Naïve Bayes algorithms. The SF that meets the conditions based on the input data of transmission power, packets and energy is chosen.

D. Step 3: SF Integration and Transmission

In this step, the AS receives the node's data from the NS. The AS extracts the energy consumption/packets value and the TP and feeds the ML model with the above data. Then, the ML model extracts the SF value. From this point, the AS sends a downlink message through LoRa, to update the node's SF.

Furthermore, a simple application layer mechanism has been created to keep track the lowest SF of the node. This is necessary for the cases where the ML model returns a SF value that falls below the minimum required SF in order to be received successfully by the GW. Also, the ADR part that runs in the nodes is used to deal with the cases where the initial SF assigned (before reaching to the NS) are lower than the minimum SF.

IV. COMPARISON MECHANISMS

A. Comparison Mechanism–RSF(RandomSF) selection

The 1st mechanism that the ML mechanism will be compared, relies on random SF selection for the data transmission. The algorithm chooses a random value between 7 and 12 to be used during the transmission. The SF values do not change in the whole simulation. This can be realistic because in the many cases it is unknown what SF should the user assign, so actually, the node's SF can be considered as random.

B. Comparison Mechanism–ADRSF(ADRSF) selection

The next mechanism that is examined in this paper is the ADR mechanism [12]. Given the time on air, nodes closer to the gateway do not need the high link budget that goes along with SF12; nor do they need to stay on air as long. So, the ADR can optimize the node's SF, and minimize the subsequent Time on Air, according to the link budget of each node. ADR is a very simple heuristic mechanism that consists of two parts, one running in the NS and the second in the node itself. The goal of ADR running at the node is to increase the SF (thereby reducing the data rate) if uplink transmissions cannot reach the gateway. If a downlink frame is not received within a configurable number of frames, the node increases the SF of the subsequent uplink frame.

In this paper, two variants of the ADR mechanism are tested. The first one, in the NS part, the link quality is estimated using the max SNR value from the latest 20 frames, while the second version of the ADR proposed in [13] uses the average of the latest received frames. The ADR variant using max operator we define it as MaxADR, while the variant using the average operator as AvgADR.

Authorized licensed use limited to: Southeast University. Downloaded on February 07,2023 at 03:03:55 UTC from IEEE Xplore. Restrictions apply.

V. Simulation Results and Evaluation

In this paragraph, the experimental results are presented. The parameters of the simulation setup are presented in Table 2 and is considered as urban ([13]). The nodes were randomly deployed in the area, according to the uniform distribution. Each node sends uplink packets in every period that follows the exponential distribution with λ 1000ms. The simulations abide by the 1% duty cycle restriction that is imposed in the Industrial, Scientific, and Medical (ISM) bands [14]. The simulation time was 10 days where experiments run multiple times in order to reduce the bias.

As far as the evaluation is concerned, we present the following: ADR mechanism using the max operator, ADR mechanism using the average operator, the case where the ADR is disabled (NoADR), the k-NN based ML mechanism, and the Naïve Bayes ML mechanism. We compared the ML-based mechanism with the ADR, as the ADR is the de-facto mechanism used in LoRa. About ML, we used two algorithms, in which both achieve high scores in 4 metrics as Table 1 presents.

Some of the characteristics of LoRa networks is the balance between energy requirements and its capabilities in terms of transmission and coverage. For this reason, delivery ratio and energy consumption were the main evaluation criteria that were used in our simulations. The delivery ratio is computed as the ratio of the total number of the messages received successfully by the NS divided by the total number of the messages sent by the nodes. The energy consumption is calculated by the energy consumed by all nodes divided by the number of successfully received messages by the NS.

Table 2. Simulation Parameters

Parameter	Value
Number of Nodes	50 – 250, 50 step
σ	0
Spreading Factors (SF)	7-12
Transmission Power (TP)	2 to 14 dBm
Code Rate	4
Carrier Frequency	868 MHz
Bandwidth	125KHz
Number of GWs	1
d0	40 m
PL(d0)	127.41 dBm
Packet size	20 bytes
λ	1000 ms

The Fig. 3 depicts the energy consumption in NoADR case, with ADR and ML mechanisms. Specifically, in Fig. 3, the energy consumption of all examined mechanisms is presented. According to Fig. 3 the random assignment of the SF is the worse method. Comparing the 4 remaining methods, the ADR has the least energy consumption the AvgADR consumes more energy than the ML based mechanisms. The Naïve Bayes based ML mechanism consumes a little more than the k-NN, but in the case of 100 nodes the energy consumption is almost identical. The reason why the ML mechanisms are seemed to perform a little worse is that in contrast to the ADR, only the SF is optimized, while in ADR the TP is also changing accordingly. But even though there is no policy to change the TP, the ML based algorithms can get close to the ADR algorithm.

As we can see, Fig. 4 depicts the delivery ratio of the NoADR case, the case with ADR enabled and the one with ML mechanisms enable. Also, there is no significant difference between the optimized cases, especially as the number of nodes increases, while the randomly assigned SF method yields the worst results. The ML based mechanisms perform better than the AvgADR mechanism.

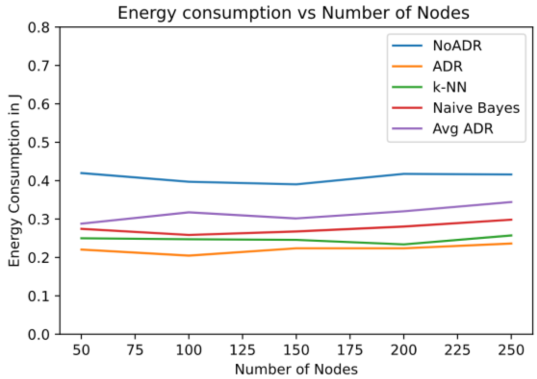

Fig. 3. Energy consumption of the examined mechanisms, as the number of the nodes is increasing

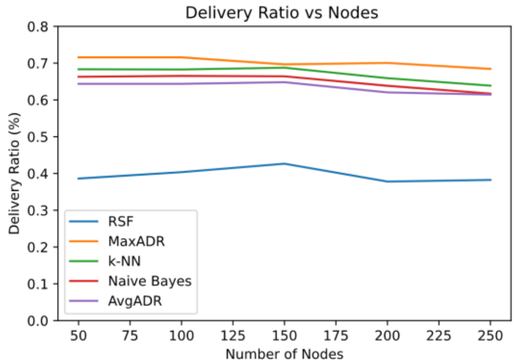

Fig. 4. Delivery Ratio of the examined mechanisms as the number of nodes increases

In order to evaluate the delivery ratio results, a thorough insight of the created data was investigated. After the research, the authors concluded that the main reason for the slightly worse performance compared to the ADR algorithm can be understood from the Fig. 5., in which the Random SF selection case, yields the worst results, due to the largest number of packets that could not be received by the GW. This derives from the signal power that falls below the GW's sensitivity threshold. Among the 4 mechanisms, namely the two variants of the ADR and the two ML mechanisms, the MaxADR has the least number of packets that fall below the GW's sensitivity threshold. The ML mechanisms fall between the AvgADR and the MaxADR, as the number of packets that could not be received by the GW is between the MaxADR and AvgADR. This is the reason that the ML based mechanisms perform slightly worse than the MaxADR, but better than the AvgADR.

In this phase, it is important to understand the part of the ADR algorithm that runs in the nodes. After 64 uplink transmissions the node requests from the NS to send a downlink packet, within the next 32 uplink packets. In the scenario where the node's SF is below the lowest necessary value accepted by the GW successfully, 96 uplink transmissions need to be sent in order the node to increase the SF value. Thus, when the ML models make one false prediction that forces the node to have a SF value that falls below the sensitivity threshold, more than 96 uplink transmissions need to be sent, in order to reach the lowest SF value. In order the simulations to be more realistic no

Authorized licensed use limited to: Southeast University. Downloaded on February 07,2023 at 03:03:55 UTC from IEEE Xplore. Restrictions apply.

assumption about the lowest SF was made in contrast to work [5].

To extract the lowest SF, we made a simple application layer mechanism that keeps track the lowest SF. Despite this, not a prior knowledge was assumed, in order to find the lowest SF, unsuccessful uplink transmissions occurred in some cases. Finally, in contrast to [5], in this paper the nodes had the ability to transmit in the range of the accepted TP values. The authors assumed that all the nodes transmitted in the highest value of TP, something that is not common in real life scenarios and deployments. Also, the nodes could transmit with different TP values, but the ML mechanisms did not change dynamically the TP values. (but used the first randomly assigned in the beginning of the simulation) On the contrary ADR mechanisms can change the TP values dynamically. Despite this, the ADR mechanisms yield slightly better results, thus making the adoption of ML mechanism a promising candidate for SF selection.

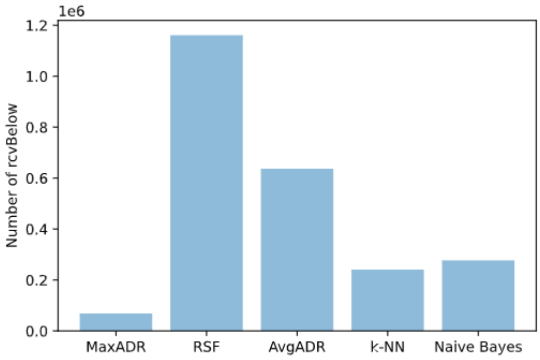

Fig. 5. Number of packets that GW did not receive, due to signal was weaker than the GW's sensitivity

VI. Conclusion and Future Work

LoRa technology is one promising wireless technology that copes to deal with applications that need long range and low energy communication. In this paper, we created a library that enables the communication between OMNeT++ simulator called FLoRa and the sci-kit learn library. Moreover, the authors investigated the possibility of using ML based mechanism for SF prediction. In this framework, a thorough study has been conducted and a comparison in terms of delivery ratio and energy consumption among 5 cases is presented. The studied ML mechanisms allow predicting a SF that could be used from the NS in order to transmit the data. Based on a trained dataset as exported in ADR case, we use it in our model to calculate the most suitable SF based on our input data. We studied the cases of k-NN and Naïve Bayes classifier for the ML mechanism. Finally, a comprehensive presentation of the limitations has been made, in order to expose the vulnerabilities and open areas of the integration of ML in LoRa SF predictions. The problem is that classification error can lead to retransmissions, that in case of LPWAN can be costly, as to increase the SF value by one unit, 96 unsuccessful uplink transmissions should be made. To deal with this issue, we created an application layer mechanism to keep track the lowest SF. This could be beneficial to the scientific community. As far as the future work is concerned, an investigation of the ML for LoRa network optimization will be conducted through TP dynamically change, based on the results of this paper.

Acknowledgment

This research has been co-financed by the European Union and Greek national funds through the Operational Program Competitiveness, Entrepreneurship and Innovation, under the call RESEARCH - CREATE - INNOVATE (project code: T1EDK-01520).

References

[1] Buurman B., Kamruzzaman J., Karmakar G., and Islam S., Low-Power Wide-Area Networks: Design Goals, Architecture, Suitability to Use Cases and Research Challenges, IEEE Access, vol. 8, 17179–17220 (2020).

[2] S. Sağır, İ. Kaya, C. Şişman, Y. Baltacı and S. Ünal, "Evaluation of Low-Power Long Distance Radio Communication in Urban Areas: LoRa and Impact of Spreading Factor," 2019 Seventh International Conference on Digital Information Processing and Communications (ICDIPC), Trabzon, Turkey, 2019, pp. 68-71.

[3] M. Turmudzi, A. Rakhmatsyah and A. A. Wardana, "Analysis of Spreading Factor Variations on LoRa in Rural Areas," 2019 International Conference on ICT for Smart Society (ICISS), Bandung, Indonesia, 2019, pp. 1-4.

[4] G. Zhu, C. Liao, T. Sakdejayont, I. Lai, Y. Narusue and H. Morikawa, "Improving the Capacity of a Mesh LoRa Network by Spreading-Factor-Based Network Clustering," in IEEE Access, vol. 7, pp. 21584-21596, 2019.

[5] T. Yatagan and S. Oktug, "Smart Spreading Factor Assignment for LoRaWANs," 2019 IEEE Symposium on Computers and Communications (ISCC), Barcelona, Spain, 2019, pp. 1-7, doi: 10.1109/ISCC47284.2019.8969608.

[6] B. Reynders, W. Meert and S. Pollin, "Power and spreading factor control in low power wide area networks," 2017 IEEE International Conference on Communications (ICC), Paris, 2017, pp. 1-6, doi: 10.1109/ICC.2017.7996380.

[7] C. Bouras, A. Gkamas, and S. A. K. Salgado, "Energy efficient mechanism for LoRa networks," Internet of Things, vol. 13, p. 100360, Mar. 2021, doi: 10.1016/j.iot.2021.100360.

[8] T. Hadwen, V. Smallbon, Q. Zhang, and M. D'Souza, "Energy efficient LoRa GPS tracker for dementia patients," in 2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Seogwipo, Jul. 2017, pp. 771–774, doi: 10.1109/EMBC.2017.8036938.

[9] C. Bouras, A. Gkamas, S. A. Katsampiris Salgado, and V. Kokkinos, "Comparison of LoRa Simulation Environments," in Lecture Notes in Networks and Systems, Springer International Publishing, 2019, pp. 374–385.

[10] C. Bouras, Kokkinos, V., and Papachristos, N., ?Performance evaluation of LoraWan physical layer integration on IoT devices?, in Global Information Infrastructure and Networking Symposium (GIIS 2018), Thessaloniki, Greece, 2018.

[11] N. V. Chawla, K. W. Bowyer, L. O.Hall, W. P. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique," Journal of artificial intelligence research, 321-357, 2002.

[12] "Understanding the LoRA: Adaptive Data Rate": https://lora-developers.semtech.com/uploads/documents/files/Understanding_LoRa_Adaptive_Data_Rate_Downloadable.pdf

[13] M. Slabicki, G. Premsankar and M. Di Francesco, "Adaptive configuration of lora networks for dense IoT deployments," NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium, Taipei, 2018, pp. 1-9, doi: 10.1109/NOMS.2018.8406255.

[14] "The Things Network: https://www.thethingsnetwork.org/docs/lorawan/duty-cycle.html

[15] M. C. Bor, U. Roedig, T. Voigt, and J. M. Alonso, "Do LoRa Low-Power Wide-Area Networks Scale?," presented at the MSWiM '16: 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems, Nov. 2016, doi: 10.1145/2988287.2989163.

Authorized licensed use limited to: Southeast University. Downloaded on February 07,2023 at 03:03:55 UTC from IEEE Xplore. Restrictions apply.