International Journal of Wireless Networks and Broadband Technologies
Volume 10 • Issue 2 • July-December 2021

A Comparative Study of Machine Learning Models for Spreading Factor Selection in LoRa Networks

Christos John Bouras, University of Patras, Greece
https://orcid.org/0000-0001-9160-2274

Apostolos Gkamas, University Ecclesiastical Academy of Vella, Ioannina, Greece
https://orcid.org/0000-0003-0966-5140

Spyridon Aniceto Katsampiris Salgado, University of Patras, Greece
https://orcid.org/0000-0002-7486-5934

Nikolaos Papachristos, University of Patras, Greece
https://orcid.org/0000-0002-0564-6850

ABSTRACT

Low power wide area networks (LPWAN) technologies offer reasonably priced connectivity to a large number of low-power devices spread over great geographical ranges. Long range (LoRa) is a LPWAN technology that empowers energy-efficient communication. In LoRaWAN networks, collisions are strongly correlated with spreading factor (SF) assignment of end-nodes which affects network performance. In this work, SF assignment using machine learning models in simulation environment is presented. This work examines three approaches for the selection of the SF during LoRa transmissions: 1) random SF assignment, 2) adaptive data rate (ADR), and 3) SF selection through machine learning (ML). The main target is to study and determine the most efficient approach as well as to investigate the benefits of using ML techniques in the context of LoRa networks. In this research, a library that enables the communication between ML libraries and OMNeT++ simulator was created. The performance of the approaches is evaluated for different scenarios using the delivery ratio and energy consumption metrics.

KEYWORDS

LoRa, LoRaWan, Machine-Learning, Spreading Factor Assignment

INTRODUCTION

The accessible distribution of Internet of Things (IoT) devices has introduced industries, organizations, and individuals to the development of significant worth including IoT applications in section of rescue monitoring. It's a fact that IoT can certainly help in developing better solutions and real time added value to IoT devices and applications suitable to improve our lives and operational processes. In search and rescue (Bouras, Gkamas, & Katsampiris Salgado, 2021). and general healthcare area, there are several occasions such as rescue monitoring and tracking where sensors can play an important role. Even in COVID 19 era, many works have been made in order to tackle the pandemic, using

DOI: 10.4018/IJWNBT.2021070106

IoT solutions like Vedaei et al., 2020. Vedaei et al., (2020) have implemented an IoT based system for healthcare and physical distance monitoring system. The main components are the Machine Learning (ML) fog-computing tools, biometric sensors and wireless communication technologies. One of these wireless communication technologies is Low Power Wide Area Networks (LPWAN). LPWAN comes to solve the problem of transmitting data to long distances, with very small energy consumption. Some examples of LPWAN technologies are Long Range (LoRa), Narrowband IoT (NB-IoT) (Routray, S. K., & Mohanty, S. (Eds.). (2021)), SigFox and Weightless. Each technology has its advantages and disadvantages, trying to provide energy-efficient, long-distance, low-cost solutions, sacrificing high throughput, and low latency similar to what cellular technologies provide (Buurman, Kamruzzaman, Karmakar, & Islam, 2020).

As mentioned before, IoT tries to cope with different parameters in the context of the application. In this paper, the authors study the LoRa technology for a number of reasons. Firstly, the Medium Access Control (MAC) layer of the LoRa stack is open, and in order to transmit it is not necessary to pay for a paid subscription, in contrast to the NB-IoT technology, making LoRa a more appealing solution. In a LoRaWAN network, the nodes are not related with an explicit gateway. Instead, data broadcasted by a node is usually received by many gateways. Each gateway will forward the received packet from the end-node to the cloud-based Network Server (NS) via some backhaul. NS perform complex operations including management of the network and filtering redundant received packets, performing security checks, scheduling acknowledgments through the optimal gateway, performing adaptive data rate etc.

In this paper, as far as the LoRa SF assignment is concerned, ML techniques are used, in order to export the appropriate SF that could be used by the network server for data. Authors intent to show their findings regarding the possibility of using ML techniques for SF assignment in LoRa networks. Firstly, the authors explored the data created in the process of LoRa transmissions, and then analyzed and compared four classifications algorithms for the SF assignment using the most used metrics: accuracy, precision, recall, and F1 score. After the evaluation of the models, the authors implemented the ML based system in LoRa and can be used and extended as a separate library to research or university projects. Specifically, a library was created in order to enable the communication between two very important tools, the OMNeT++ based framework called FLoRa, and one of the most well- known libraries for ML called scikit learn. The aforementioned tool uses the FLoRa simulator and python for the ML operations (namely for the training and testing phase of the classification models and for the SF prediction/assignment). Also, we formulated the process of SF assignment as a classification problem. Using the above-mentioned library, two mechanism were created based on the k-NN algorithm and Naïve Bayes classifier. Finally, we present a comparative evaluation of the two proposed mechanisms against two variants of the Adaptive Data Rate (ADR) and the random initialization of the SF. The comparative evaluation was based on delivery ratio and the energy consumption metrics, to study the energy consumption, and the trade-off with the delivery ratio.

The rest of this paper is organized as follows: The next section presents related work. In section “LoRaWAN” important aspects of LoRa are presented in order to better understand our contribution. In section “Background” related works are presented. In Section “Machine Learning Approach” the problem formulation as a ML problem is presented and our approach is presented. In Section “Simulations” the results of our approach and the comparison among other de facto approaches are presented. Finally, the last two Sections the conclusion and future work are presented respectively.

LORAWAN

The LoRa is a physical modulation technique and derives of Chirp Spread Spectrum (CSS). LoRa constitutes a technique designed to operate in 433 MHz, 868 MHz, and 915 MHz. One of the most notable characteristics of LoRa modulation is its resistance against the Doppler Effect and multipath fading.

In a typical LoRa deployment there are four main devices: a) LoRa end-nodes, which acquire data from sensors and then these data are transmitted, b) LoRaWAN which is the communication network. c) One or more LoRa Gateways (GWs) that receive the LoRa frames and forward them through a wired network. d) One or more Network Servers, usually in the cloud, which are responsible to process the received and are likely in charge of decision-making.

LoRa's physical layer uses CSS modulation over a variety of frequency bands in Europe and USA. The value of 868MHz is one of the common values in most regions (mainly in Europe). There are multiple factors that characterize the LoRa communication between the end-nodes and the GWs such as SF, TP, Carrier Frequency (CF), Coding Rate (CR) and of course the Bandwidth (BW). The SF is defined as the ratio between the symbol rate and chip rate. The number of chips per symbol is defined as 2^{SF}. The SF values vary from SF7 to SF12, where higher SF values achieve higher ranges. CSS modulates the data symbols into chirp signals whose frequency is constantly changing. A LoRa frame has a preamble chirp, that helps the signal detection by the receiver. Because the preamble of LoRa is the same for each transmitter, the end of the preamble is separated by two sync words.

The parameters used by LoRa are SF, coding rate (CR), and bandwidth (BW). SF is given by Equation 1:

$$SF = \log_2 \frac{R_c}{R_s} \tag{1}$$

where Rc and Rs are chip rate and symbol rate, respectively.

The relation between the data rate and the SF is defined by Equation 2, where R_b signifies the bit rate:

$$R_b = \frac{2^{SF}}{BW} \tag{2}$$

On the other hand, TP usually ranges from -4dBm to 20dBm. This parameter sets the intensity in which LoRa end-nodes transmit the LoRa data frames to the GW. Theoretically, as SF and TP increases, the LoRa coverage area is larger. CR provides security against interferences, where higher values provide higher protection (4/5, 4/6, 4/7, and 4/8). BW is the frequency width in the transmission band.

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 Equation 3 and derives from paper Bor, Roedig, Voigt, T., & Alonso, (2016), where P_{rx} is the received power, the GL is the general gains and losses and the PL(d) is the path loss model:

$$P_{rx} = P_{tx} + G_L - PL\left(d\right) \tag{3}$$

The path loss model follows Equation 4 (See (Bor, et al., (2016))):

$$PL\left(d\right) = PL\left(d_0\right) + 10nlog_{10}\left(\frac{d}{d_0}\right) + X_\sigma \tag{4}$$

where $PL\left(d_0\right)$ is the mean loss for a reference distance do, the n is the path loss exponential and the Xσ 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 Equation 5: (See (Bor, et al., (2016))):

$$S = -174 + 10\log_{10} BW + Y + SNR \tag{5}$$

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. SNR is related to the Signal-to-Noise ratio. Apart from the above, orthogonality is taken into consideration. The orthogonality dictates that the LoRa signals that have the same SF and are transmitted simultaneously, are colliding. Furthermore, stronger signals (with larger TP value) that have the same SF in a simultaneous transmission, can be received by the GW.

LoRaWAN in a typical deployment assumes a star topology. It refers to the MAC layer and uses as a physical layer the LoRa modulation. It defines three device types: Class A, Class B, and Class C, according to the application requirements. The LoRaWAN consists of a gateway and multiple end-devices. To save battery life, a single-hop with simple protocols connects the gateway and end-devices. SF and the TP of a node can be assigned either by the gateway or the node itself, while the channel for is end-device communication is allocated by the gateway. A dedicated module (e.g., SX1301) is usually used in order to support multi-channel and multi-data rates.

When the end-devices are stationary, an adaptive data rate (ADR) can be used to achieve energy consumption of the end-device. Particularly, LoRaWAN optimizes data rate, airtime, and energy consumption. If ADR is enabled, the gateway divides the frequency band into eight 125 kHz channels and listens for the uplink frame in any channel simultaneously. In each channel, the LoRaWAN network server estimates the link budget of the channel as follows: it calculates the margin of SNR and then decides which value of the SF and the TP is the best suited for each node.

A huge number of end-devices can be supported by a gateway that supports multichannel and multi-data rates. In indoor environments with many obstacles, the communication coverage decreases; hence, it is difficult to expect high-density end-devices in the network. Besides, the single-hop star topology of LoRaWAN restricts the network scalability: if a new end-device exceeds the gateway communication radius, a new network with an additional gateway should be built for it. Studies such as (Adelantado et al., 2017), (Liao, Zhu, Kuwabara, Suzuki, & Morikawa, 2017) have already addressed the need for multi-hop LoRa networks. Last but not least, the communication between LoRa end-nodes and gateways can be unidirectional or bidirectional. LoRaWAN, on the other hand, specifies the architecture, layers, and protocols operating over LoRa. Mesh or stars are the two possible topologies supported in LoRa (Zhu et al., 2019).

One important parameter in LoRaWAN communication is Spreading Factor (SF). The SF parameter decides how many chirps (the carrier of the data) are sent per second. Higher SF indicates less chirps per second; hence, less data is processed per second. Transmitting the same amount of data with higher SF needs more transmission time, known as airtime. In order to achieve bigger airtime, the modem is operating and running longer and thus consuming more energy. The main advantage of high SF is that the extended airtime increases the possibility the gateway to receive the transmitted packet, thus increasing the gateway's sensitivity. Better sensitivity means that the network provides better coverage. Hence, the SF assignment is a crucial process for the network and SF value for the transmission of the data should be carefully selected. As result, SF assignment is a trade-off between performance and energy consumption which crucial for LoRa based IoT applications. In order to fully exploit the benefits of the LoRa technology and to improve its performance, the network decides the SF (graded between 7-12) based on the environmental conditions between the communication device and the gateway. The relationship between transmission and SF assignment has been thoroughly studied in previous research works, such as (Sagir, Kaya, Sisman, Baltaci, & Unal, 2019), (Turmudzi, Rakhmatsyah, & Wardana, 2019), and (Zhu et al., 2019).

One of the most important challenges that should be taken into consideration during the development of a system, is the appropriate resource allocation. Resource allocation can be focused on energy consumption, latency, throughput, packet loss etc. Many techniques have been proposed for resource allocation. One of the key factors is the use of ML for configuration parameters prediction. ML extracts meaningful information from raw data and provides accurate results. It is widely known that; this information helps in solving complex and data-rich problems like resource allocation.

BACKGROUND

Many papers have addressed the problem of parameter selection in LoRa networks. In this section, we highlight works that aim to energy reduction/resource allocation in LoRa deployments, using both traditional techniques and ML algorithms. Also, it is noted the significance of using ML in LoRa in other applications as well, such as localization, showing the importance of the integration of ML techniques in LoRa. ML and LoRa can boost the IoT acceptance in many applications in the future.

Firstly, Sagir et al., (2019) study the different SF assignment in order to verify the theoretical limits obtaining the practical performance profile of the LoRa radio. On the other hand, Turmudzi et al. (2019) study SF assignment in rural areas to determine the effect on the coverage of the mobile network. Zhu et al., (2019) attempt to off-load the data traffic into several subnets by utilizing multiple-access dimension based on multi hop LoRa network. This achieved by enabling packet transmissions in parallel with multiple SFs to become feasible. Li, Yang, & Wang, (2020), propose an energy-efficient mechanism that dynamically changes the SF and TP values, according to sparse LoRa packets. Their results improved the energy consumption as part of optimization, while having an acceptable trade-off in terms of delivery ratio. Moreover, apart from heuristic methods of setting the different LoRa parameters, Tehrani, Amini, & Atarodi, (2020) present a system that is opposite of the de facto LoRa deployments. The authors suggest a tree based LoRa topology showing that in a such topology the energy consumption of the nodes can be mitigated. In this context, the authors propose an energy efficient routing algorithm for multi hop LoRa deployments. The authors suggest that energy consumption is reduced in contrast to single hop topologies (Paul, 2020). Also, Mukherjee, Jain, & Yang, (2020) have extended the above method using neural networks for the clustering process in next generation network technologies.

Besides, Zourmand, Kun Hing, Wai Hung, & AbdulRehman (2019) present the performance and the actual coverage area of the LoRa network in both the indoor and outdoor condition using different configuration on topology and SF variables studying the behavior of the energy consumption and total system performance in case of LoRa networks. Moreover, the comparison conducted in paper (Bouras, Gkamas, Katsampiris Salgado, & Kokkinos, 2020) was important for our choice to use FLoRa simulator (“FLoRa simulator”, 2021) as it is a simulation framework for carrying out end-to-end simulations for LoRa networks.

ML can be used in various applications in LoRa networks, as well. For example, node localization is one application that ML can benefit LoRa networks e.g. (Daramouskas, Kapoulas, & Pegiazis, 2019a). The authors of the above paper have compared various localization techniques for LoRa networks, including ML based algorithms, such as clustering and Social Learning Particle Swarm Optimization (PSO). Following the above study, we use ML to tackle the problem of transmission. Daramouskas et al., (2019b) propose a Received Signal Strength Indication (RSSI) based monitoring algorithm that uses neural networks for localization in LoRa networks. The results showed that the neural networks perform well in these cases.

As far as the network optimization is concerned, Sandoval, Garcia-Sanchez, & Garcia-Haro, (2019) expressed the update process of LoRa parameters, such as SF, as a reinforcement learning problem. The parameter configuration is made by neural networks. The results yielded by their policies show a 147% increase in throughput. Moreover, in paper (Yu, Mroueh, Li, & Terre, 2020) a multi agent Q-Learning algorithm is proposed in order to achieve better resource allocation in LoRa networks. Particularly, the SF is dynamically changing in order to reduce the collisions that can be occurred due to SF transmission. The results yielded were robust, but the input of their mechanism is the location of the nodes, something not realistic for several applications. For example, if the network operator is not aware of the node's location, then a GPS module is necessary, leading to an increase of energy consumption mitigating the benefits of the mechanism. Cuomo, Garlisi, Martino, & Martino (2020) investigate the possibility of integrating ML for LoRa network optimization. Their study concluded in proposing a system that uses different ML tools, such as clustering, Long Short-Term Memory Neural Networks and decision trees in order to predict the period of inter-arrival time of the packets, showing promising results. Park, Lee, & Joe (2020) propose a reinforcement learning based system that assigns the optimal LoRa parameters such as the SF, transmission power (TP) and channel bandwidth. Cui, & Joe (2020) applied ML algorithms to tackle with the collisions that occur in LoRa networks, with dynamic parameter allocation. Specifically, they handle the collision avoidance problem as a time series problem, and they apply Long Short-Term memory extended Kalman filter to predict the collisions.

MACHINE LEARNING APPROACH

Machine Learning Algorithms

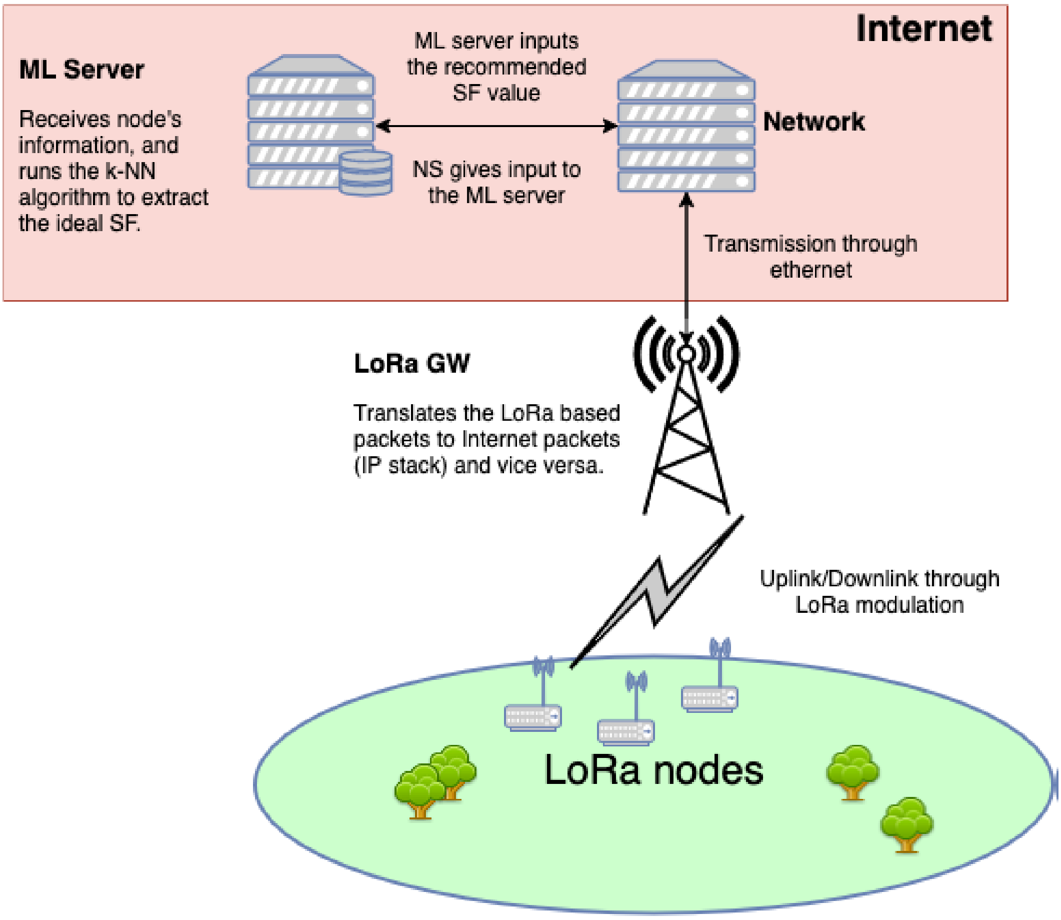
ML consists of different approaches that aim to help people in making decisions, and the approaches can be categorized into three main categories a) supervised learning b) unsupervised learning c) reinforcement learning. Supervised learning is the process in which the learning is occurred with the use of data that we know exactly their class, in other words, labeled data. The procedure in which the learning occurs is called training and the dataset used is called training dataset. After the training, to test the performance of the algorithm, we must test an unknown part of the dataset known as the testing dataset. Supervised learning can be used for classification and regression problems. Classification problems are the problems in which the prediction is about discrete finite labels, while the regression problems aim to predict continuous target labels. When the learning process does not involve any target labels, then the unsupervised learning is discussed. The most common use of unsupervised learning is clustering in which the user tries to find some groupings in the data. Finally, reinforcement learning refers to the learning and predicting of the next action that helps to the maximizing the benefit of minimizing the cost in the future (Figure 1).

In this work, we use supervised learning in a classification context. Specifically, the problem in which the authors use ML is the following: The goal is to assign a value to the SF. The values of SF vary from 7 to 12. So, the problem of SF assignment can be considered as a classification problem. Because the target values range from 7 to 12, the classes are numbered to 6. Hence, the problem of the SF assignment is suggested to be as a multi-class classification problem. After the formulation of the learning task the authors describe the algorithms used in this work, which are the k-NN, Naïve Bayes, and Support Vector Machines.

k-NN

The k-NN algorithm is a classification algorithm whose basic assumption is the fact that the data points in the dataset that have similar behavior exist in a small proximity. This assumption leads to the formulation of the learning goal as the classification of the new unseen data points by calculating the distance of the K data points in the training set that has the smaller distance in the feature space. The distance is a function that is used to express how similar or not is the new unseen data point with the data points in the training dataset. The distance can be the Euclidean, Hamming, or Mahalanobis distance. The distance is expressed in Equation 6:

Figure 1. SF selection architecture through ML



$$l = \sum_{1}^{n} \left(\left| x_i - y_i \right|^p \right)^{\frac{1}{p}} \tag{6}$$

Naïve Bayes

Naïve Bayes in reality is not one classifier, but a class of probabilistic classifiers. The basic idea is that, given an input vector, that represents the unseen data point, a Naïve Bayes classifier, applies the Naïve Bayes theorem, assuming independence between the features of the given input vector. The probabilities in the Equation 4 $\Pr\left\{t = c\right\}$ and $\Pr\{x_i \mid t = c\}$ can be assumed to follow a distribution. Thus, the Naïve Bayes classifier can use the Gaussian or Bernoulli distribution, etc. In this work, the Gaussian variant of the Naïve Bayes classifier is used. The main advantage of the Naïve Bayes classifier is that can achieve high accuracy with small data, in contrast to more complex models, such as neural networks.

The Bayes' theorem is applied (t is the class variable and $X = \left[x_1,\ x_2, \ldots, x_k\right]$ is the input unseen data point) the classification problem is expressed in Equation 7:

$$y = \arg_{c} \max Pr\left\{t = c\right\} \prod_{i=1}^{n} Pr\{x_{i} \mid t = c\} \tag{7}$$

Support Vector Machines

The Support Vector Machines (SVMs) are contrary to the Naïve Bayes, a non-probabilistic family of classifiers. The main idea behind the SVMs, is that the objective is to find a hyperplane that splits the classes of the training set with the largest margin. When the new unseen data is fed to the SVM, the prediction of the label is occurred based on which part of the hyperplane it falls. The SVMs can handle both binary and multi class problems and are supposed one of the best classification algorithms. The formulation of the problem, let take the binary case, is a linear classification task, and to be solved is handled as a constrained optimization problem.

Decision Trees

The Decision trees are different from the algorithms mentioned above. The Decision tree is a supervised learning method that can be used in both classification and regression tasks, and its goal is to create a model that uses simple if-else statements. The decision trees are very simple to understand and can be visualized easily. The problem with the Decision tree is the fact that it can be difficult to generalize a Decision tree, because as the problem becomes more complex the tree can be more complex and difficult to perform as well as other classification algorithms, such as the SVMs.

Feature Selection and Data Preprocessing

Problem Formulation

Before moving to the description of the steps that were followed in this paper, it is necessary to formulate our problem. We assume the problem of the SF assignment as a multiclass classification problem. The classes are the target SFs, so the classes are in the range of 7 to 12. When the ML algorithm is executed, according to the input data it gives as an output the class in which the node should be assigned, that in our case is the SF.

Dataset Creation and Preparation

Firstly, the simulation executed without the ADR mechanism enabled. The data created were used for the training phase. Before moving to the training phase, it is important to extract the necessary knowledge about the created dataset as part of our study. So, in Figure 2 the number instances of each class are presented on the left. The SF allocation through ADR creates an imbalanced dataset, where the SF with value 12 has most of the instances, while the instances of SF with value 7 is the minority in the dataset. For this reason, it was necessary to create synthetic data, according to the SMOTE-NC technique ("SMOTE-NC", 2021). This helps us to reduce the bias against some SFs values.

In Figure 3 the dataset's instances are plotted. As it is presented, the data are separable, as the classes (the cases with the same color is one class) are not mixed. There are some cases where a class is close to others, something that it can be a problem of misclassification, the classifiers preform very good and it is possible to create robust classifiers. Finally, the data was scaled, as the scale of the TP is different from the second feature that is the Energy consumed per packets sent, using the Max absolute value scaler.

Feature Selection

Also, in classification problems is necessary to define the features with which the classification task is done. In our case, using chi-squared analysis, the total energy divided by the total packets sent and the Transmit Power (TP) were selected as the features. Here, it is important to note that the

Figure 2. Bar chart of the number of the instances of each SF in the dataset

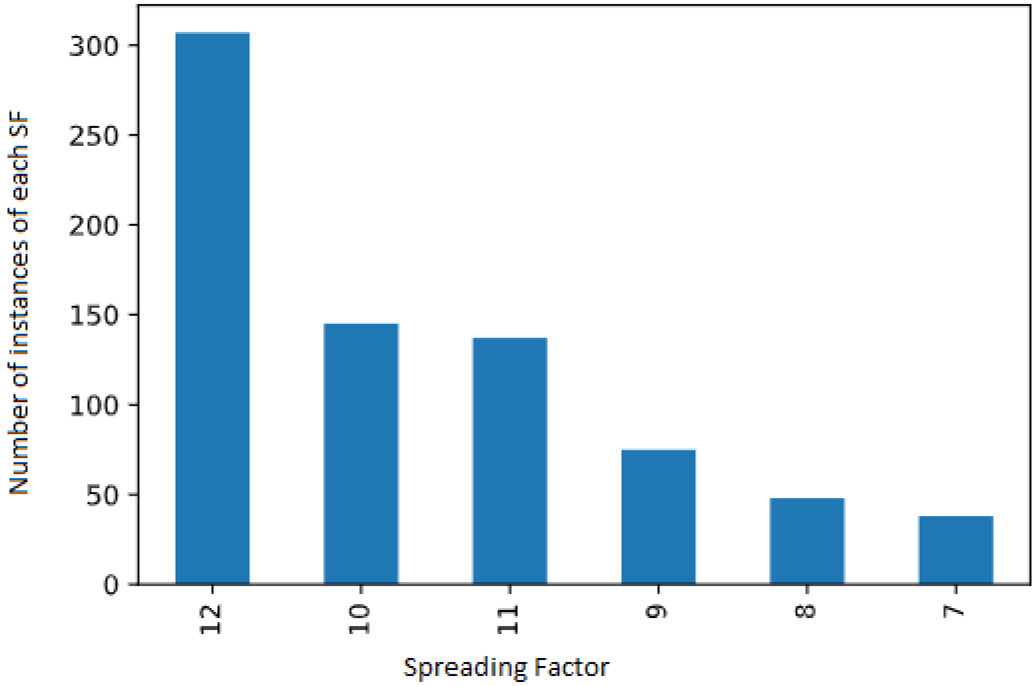
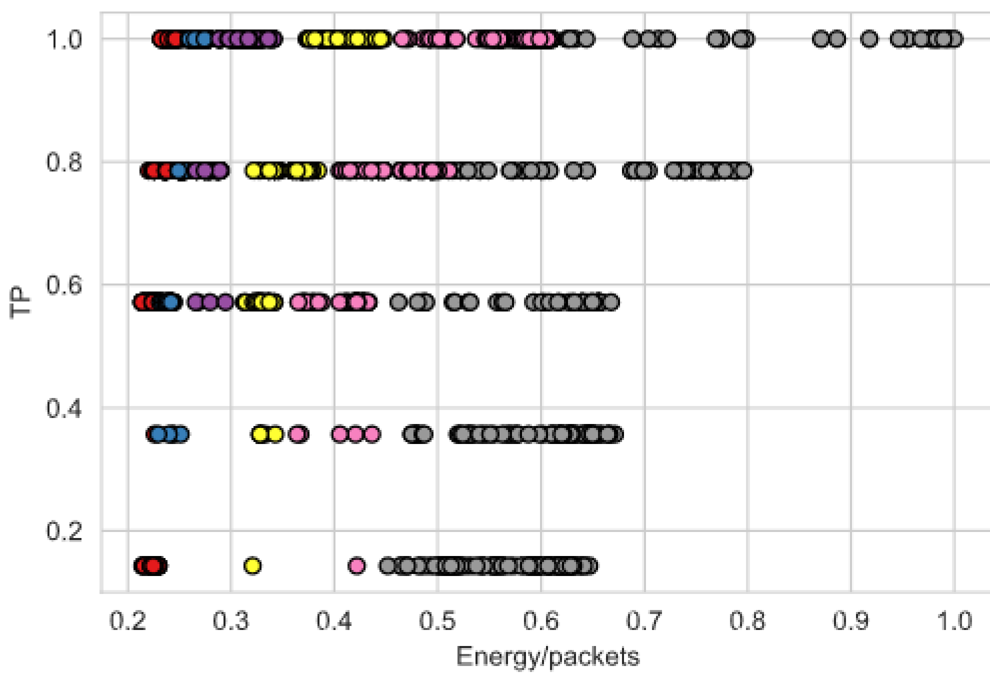


Figure 3. Visualization of the dataset



localization in LoRa networks is not so accurate as of the Global Positioning System (GPS), with an error reaching about 400m (Daramouskas et al., 2019b) (Bouras, Gkamas, Kokkinos, & Papachristos, 2020). For this reason, the node's position was excluded and not considered as a potential feature.

Training

After the preprocessing and feature selection phase, we trained the algorithms. The dataset was divided into two parts, 75% being the training dataset, and the 25% being the testing dataset. 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 three classifiers. From the different variations of the Naïve Bayes algorithm, the Gaussian Naïve Bayes was chosen, as it gave better results by far, in terms of accuracy. The difference between the Gaussian to the other variants of Naïve Bayes was huge, e.g. the Multinomial variant achieved accuracy as low as 30%. As far as the SVM is concerned, again 10-fold cross-validation was used. Firstly, we limited the parameters by the Random Search method, and then using Grid Search, the final parameters were chosen. The parameters that achieve high scores are a) the linear function as the kernel function, and b) c = 10. The mean validation score in the 10-fold cross-validation is 0.946. Finally, as far as the decision tree classifier is concerned, again we conducted random and grid search in order to find the most suitable parameters. From the 10-fold validation that was conducted in each set of parameters, the model with the following parameters was selected: as the criterion of the quality of the split was selected to be the Gini function, the max depth of the tree is 23, the max features that are considered to be is 1 and the minimum number of the samples to split an internal node is 2. The mean validation score of this model is 0.99 with a standard deviation of 0.08.

Evaluation of Classification Algorithms

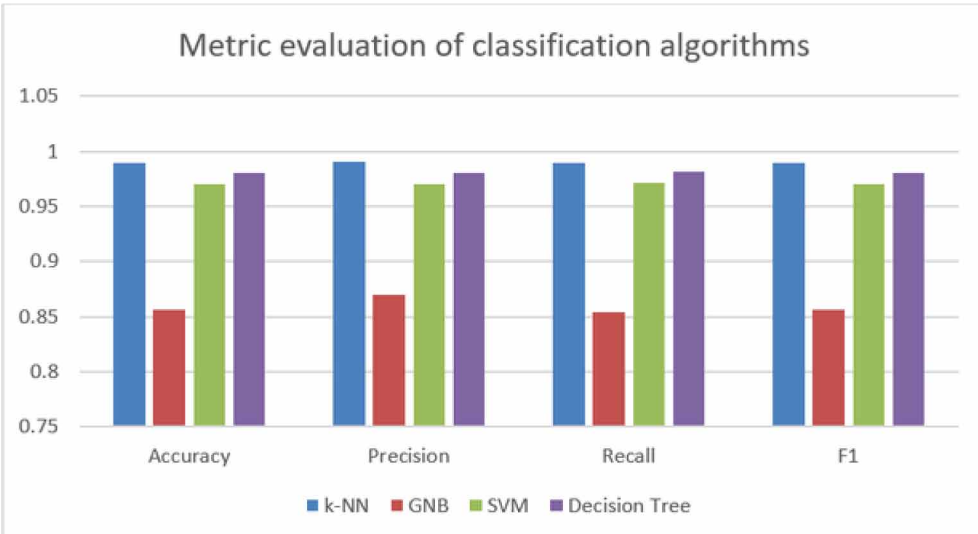
After the training phase, the evaluation of the models was conducted. We fitted the models with the training dataset and then, the accuracy, precision, recall, and the F1 Score were used as metrics for the ML algorithm evaluation, in the testing dataset. The accuracy refers to the ratio of correct predictions to the total predictions. The precision metric is 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. In Figure 4 a comparison figure shows the four classification algorithms in terms of accuracy, precision, recall, and F1. In all four metrics, the k-NN algorithm scored the highest, following by the Decision Tree, the SVM, and the Naïve Bayes. As all the algorithms are robust as in all metrics the scores ranged from 0.8 to 0.98. The reason that the k-NN algorithm scored the highest is that only two features were used for the classification problem and the phenomenon of the "curse" of the dimensionality, and the data were separable in most cases, thus the performance was great (Table 1).

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

Apart from the aforementioned classification metrics, in order to have a more thorough understanding of the classification algorithm's results, the confusion matrix of each classifier was plotted and presented in Figure 5.

Figure 4. Comparison of the Classification algorithms



ML Mechanism Integration in LoRa Network

In this subsection the overall ML based mechanism for the SF selection in LoRa is presented. In the system model, the decisions of the mechanism about the SF are happening in the Network Server who is responsible for the whole transmission. As far as the simulation process is concerned the communication between the FLoRa simulator and the scikit learn library is presented in Figure 6. The mechanism consists of three steps: a) Export NS values to the ML server, b) Selection of SF based on ML algorithms and finally c) Setup of the SF configuration to the LoRa network.

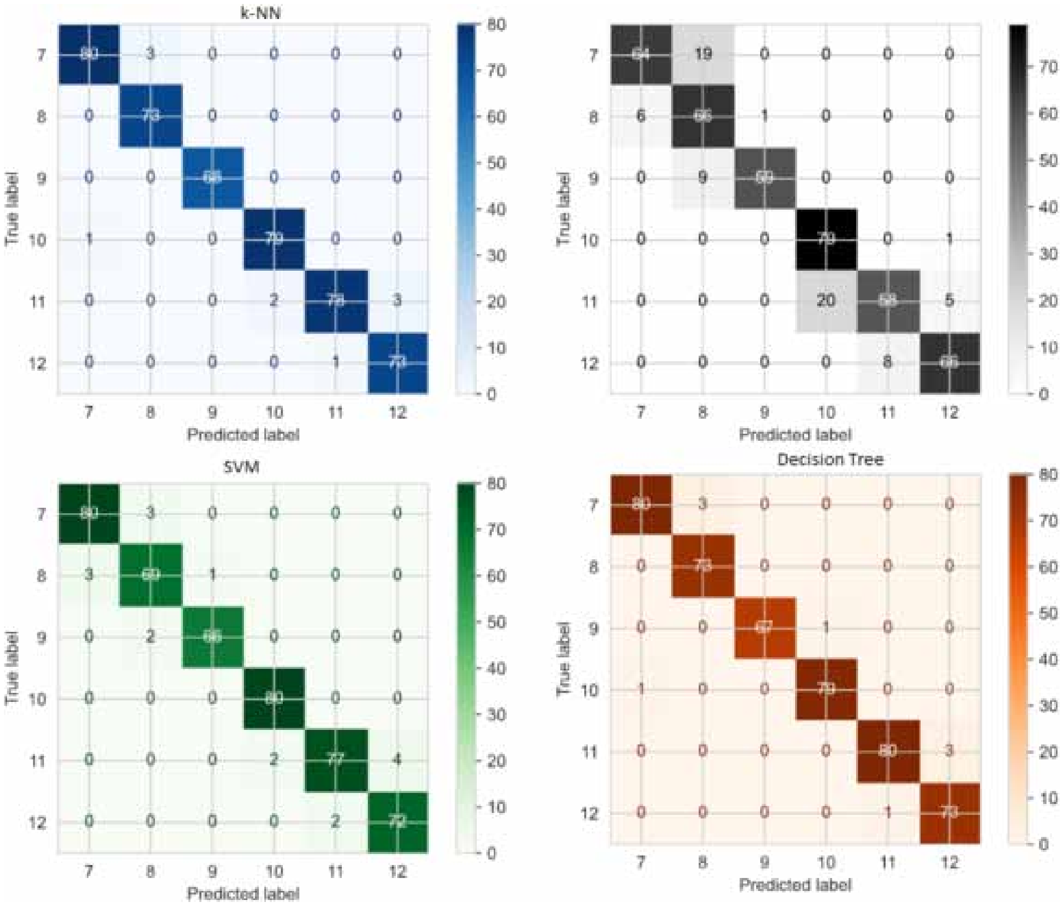
Step 1: Export NS Values to ML Server

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 a file with.csv extension. This file contains a test data set for our experiments, and it will be needed as input to the ML server. This information is the TP, the packets sent as well as the energy consumed. The above information is used as input for the ML server during our experiments.

Step 2: SF Selection

In order to select 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 ideal SF that meets the conditions based on the input data of TP, packets, and energy, is exported and written to a file ending in.csv. Thereafter, the csv file is used as an input in the NS for the continuation of the transmission.

Figure 5. The classifier's confusion matrices



Step 3: SF Integration and Transmission

In this step, the NS is being updated through the.csv file about the newly ideal SF and sets the parameters to the files and classes involved. The NS then sends a downlink message to the right node. After that the whole transmission process is being continued.

SF Selection Mechanisms Examined

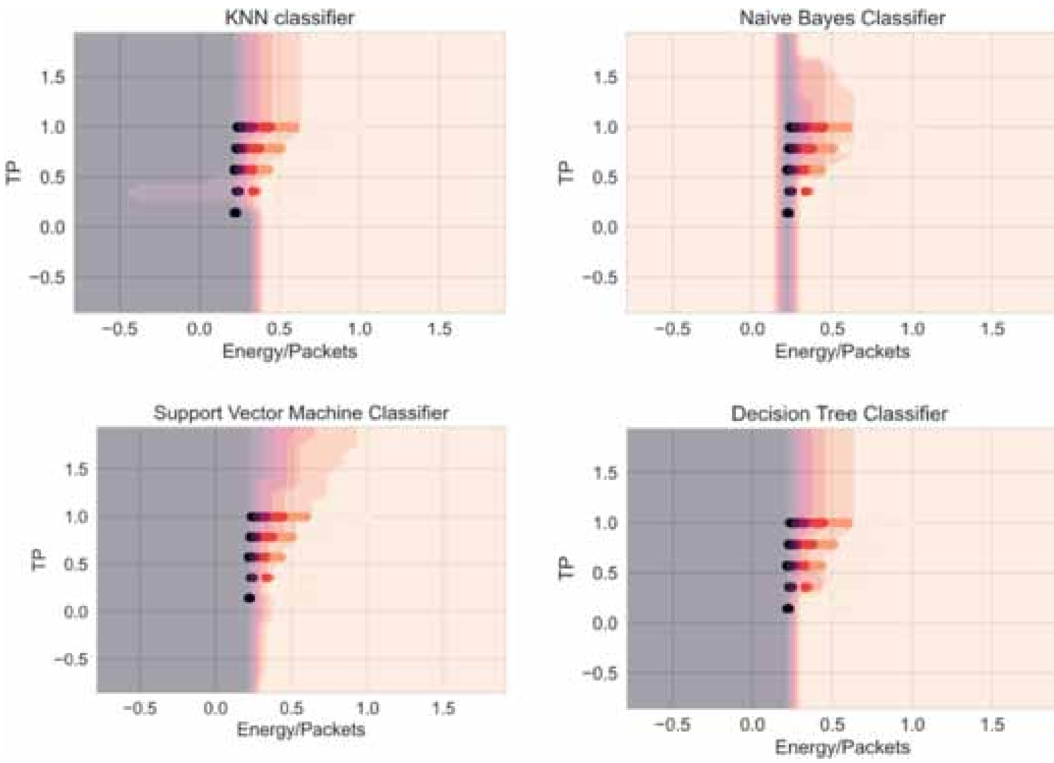
This section examines three different approaches regarding the selection of the SF during transmissions in a LoRaWan environment.

1st Approach – RSF (Random SF) Selection

The 1st approach relies on random SF selection for the data transmission. The algorithm chooses a random value between 7 and 12 in order to be used during the transmission. The procedure for obtaining the SF is presented below using pseudo code. The SF values do not change during the 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.

Pseudo Code of the SF Selection in RSF Case

Figure 6. Classification Comparison in terms of Energy per Packet




```
int SF=0; % initialize the SF
Set SF by random (7,12)
transmit(); % NS transmits the data
```

2nd Approach – ADRSF (ADR SF) Selection

The 2nd approach that is examined in this paper is the ADR mechanism ("The Things Network", 2021). 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 ADR changes the data rate based on simple rules: If the link budget is high, the data rate can be decreased (i.e. the SF is increased) If the link budget is low, the data rate can be lowered (i.e. the SF is reduced) ("Understanding the LoRa: Adaptive Data Rate, 2021). The pseudo code of the 2nd approach is presented below.

The ADR mechanism is very easy to comprehend and to implement and is heuristic. The dynamic change of the parameter selection in general is beneficial to both the node's and the network, as it can lead to reduced energy consumption and increase to delivery ratio. The main problem of the ADR is the fact that is heuristic and not always leads to optimal resource allocation in contrast to other proposed mechanism in the literature. Furthermore, the ADR converges very slowly, and in many cases a lot of unsuccessful uplink transmissions need to be occurred before the SF or the TP change. These main drawbacks have increased the interest of researcher around the world for better alternatives.

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 (Slabicki, Premsankar, & Di Francesco, 2018) uses the average of the latest received frames. The pseudo code that follows refer to both variants with the difference that instead of the max operator the average operator is used. The ADR variant using max operator we define it as MaxADR, while the variant using the average operator as AvgADR.

Pseudo Code of the SF Selection in ADR Case in Node

```
int initSF=0; % initialize the SF
int ADR_uplink = 0
int threshold = 96
while (uplink transmissions):
        ADR_uplink ++;
        if ADR_uplink > threshold;
           initSF = increaseSF(initSF);
        else
                Request Downlink frame
                transmit(); % NS transmits the data
function increaseSF(SF):
        if (SF > 7 and SF <12):
                SF = SF + 1;
                return SF;
```

Pseudo Code of the SF Selection in ADR Case in NS

```
SNRm = max of the last 20 frames
SNRmargin;
steps = floor(SNRmargin/3)
int threshold = 96
while (steps>0 & SF>7):
SF--; steps--
while steps >0 and TP>2
TP =TP-3; steps--;
while steps<0 and TP<2
TP =TP+3;
steps++;
```

3rd Approach – MLSF (Machine Learning SF) Selection

The goal of the 3rd approach is to find the suitable SF based on ML techniques. Initially the algorithm has already a trained data set with data (in normalized form) and gets input from the NS through a direct communication using a.csv file with information about the transmission. The ML model extracts the information (TP, packets sent as well as energy consumed), and feeds the k-NN algorithm or the Naïve Bayes for the SF selection. Next step is to write the selected SF value to a.csv file which will be used as input to the NS for the transmission process. NS receives the converted IP packets, and then with a downlink packet, updates the node SF. The procedure is presented below using pseudo code.

Furthermore, a simple application layer mechanism has been created in order 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 received successfully by the GW. This is feasible in real life scenarios, as the method to achieve this kind of tracking needs basic operations that the microcontroller's Arithmetic and Logic Unit (ALU) can handle. Also, the ADR part that runs in the nodes is used too, to deal with the cases where the initial SF assigned (before reaching to the NS) are lower than the minimum SF.

Pseudo Code of the SF Selection in MLSF Case

```
int SF=0; % initialize the SF
data = retrieveInput(); Retrieve input written from NS in
exported.csv
newlySF = analyzeAndRunML(data); % extract data and run k-NN/Naïve
Bayes using python module and return the ideal SF.
storeSFVariable(); % Store the selected value at config.csv
retrieve_configuration(); % NS reads configuration from the
config.csv and sets the SF to the involved classes and functions
transmit(); % NS transmits the data
```

SIMULATIONS

Description of Testbed

Regarding the needs of the results' presentation, we conducted the following experiment in the FLoRa simulator environment. The necessary simulation parameters for the conduction of experiments are presented in Table 2. The LoRa topology consists of multiple end-nodes varying from 20-250 with a 50-node step, for two different cases.

Table 2. Simulation Configuration

Parameter	Values for urban deployment
Network Size	480m*480m
Number of Nodes	50-250
σ	0
Spreading Factors	7-12
Code Rate	4
Number of GWs	4
Bandwidth	125KHz

In our simulations, we considered a network of urban and suburban setup. For this reason, we used two different models derived from paper (Slabicki et al., 2018) for both cases. Two different areas examined 480m*480m and a topology based on Oulu town with coverage area of 9800m*9800m. The deployment of the end-nodes was determined randomly in the topology. In this simulation, the stationary mobility model was used. Moreover, the energy consumption metric is considered as the ratio of the energy consumption of all LoRa nodes and the cardinality of the messages received by the NS.

Simulation Results

In this paragraph, the experimental results are presented. More specifically, 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 LoRaWAN. As far as the ML algorithms used are concerned, we used two very good algorithms, and as Table 1 presents, they achieve high scores in 4 metrics.

The goal on LoRa and devices is to maintain a balance between user comfort and energy requirements, such that the user can achieve the desired comfort level with the minimum amount of energy consumption. For this reason, delivery ration 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.

This paragraph presents the impact of the average energy consumption compared to the end nodes in our simulation. The figure depicts the energy consumption in NoADR case, with ADR and with ML mechanisms. Specifically, in Figure 7, the energy consumption of all examined mechanisms is presented. According to Figure 7, the random assignment of the SF is the worse method. Comparing the 4 remaining methods, the ADR has the least energy consumption while the k-NN based ML mechanism follows closely the AvgADR method. The Naïve Bayes based ML mechanism consumes less than the k-NN in the cases with fewer nodes, but the energy consumption is increasing faster as the number of nodes increases, in contrast to the k-NN. In the experiment with 50 nodes the AvgADR, the k-NN and Naïve Bayes based ML mechanisms are almost identical. The reason that 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. The ML based algorithms can get close to the ADR algorithm, despite the fact that there is no policy to change the TP.

As far as the delivery ratio is concerned, the delivery ratio of the NoADR case, the case with ADR enabled and the with ML mechanisms enabled is presented in Figure 8. There is no significant

Figure 7. Energy consumption of the examined mechanisms, as the number of the nodes is increasing

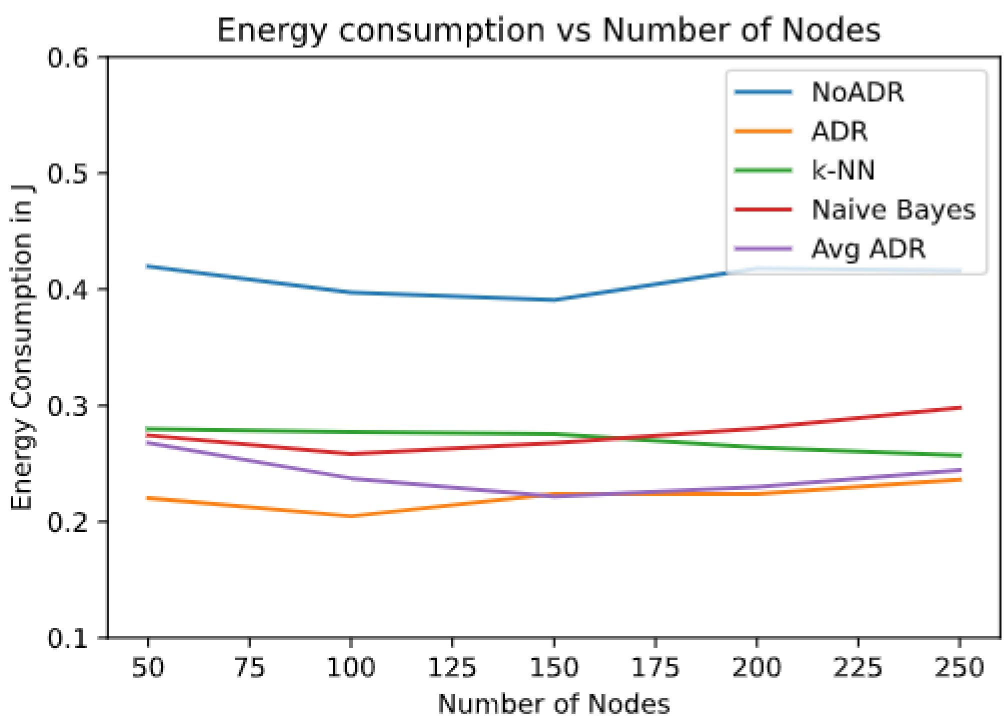
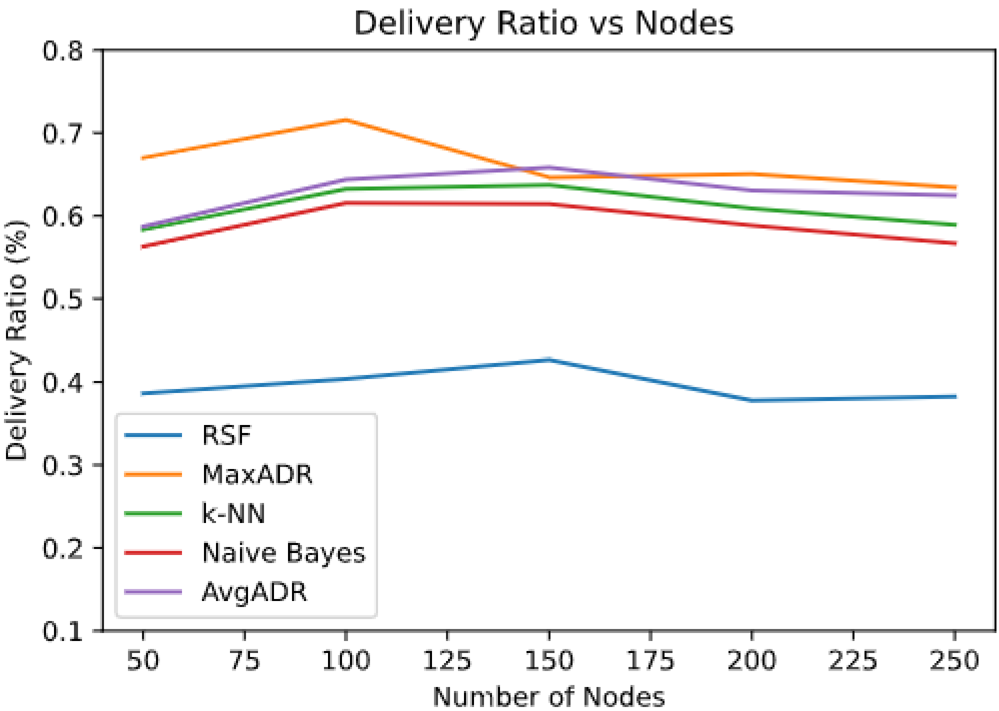


Figure 8. Delivery ratio of the examined mechanisms, as the number of the nodes is increasing



difference between the optimized cases, especially as the number of nodes increases, while the randomly assigned SF method yields the worst results.

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 of the slight worse performance compared to the ADR algorithm can be understood from the Figure 9. As Figure 9 shows, the Random SF selection case, yields the worst results, because has the largest number of packets that could not be received by the GW and therefore drops the performance of our system. 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.

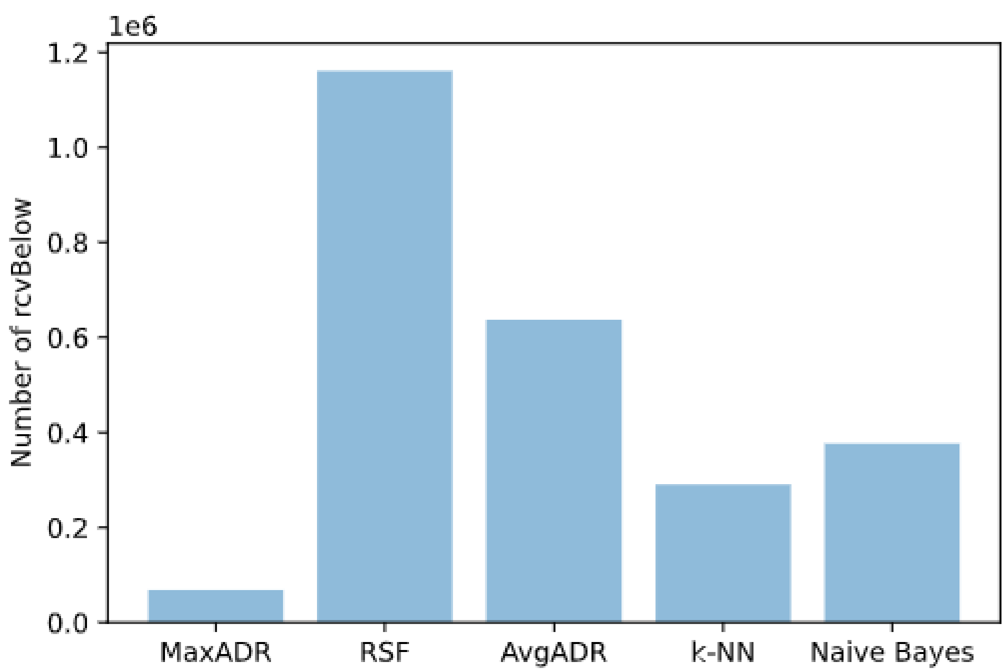
To explain this behavior, it is important to understand the part of the ADR that runs in the nodes. After 64 uplink transmissions the node requests from the NS to send a downlink packet, within the next 32 uplink packets. In the scenario where the node's SF is below the lowest necessary value to be received 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. Despite the robust results of the classifiers as presented in Figure 3 and 6 the classes are in small proximity, thus in some cases some classification errors can be occurred. Furthermore, it is worth mentioning that in contrast to work (Yatagan, & Oktug, 2019) no prior knowledge of lowest SF was assumed, because we wanted the simulations to be more realistic.

In order to find the lowest SF, we made a simple application layer mechanism that keeps track the lowest SF. In our scenario no prior knowledge of the lowest SF was assumed, thus in order to find the lowest SF in which the node should transmit so as the GW to receive the packet, some unsuccessful uplink transmissions were occurred. Finally, in contrast to (Yatagan, & Oktug, 2019), in this paper the nodes had the ability to transmit in the range of the accepted TP values. In paper (Yatagan, & Oktug, 2019) 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, in this work the nodes could transmit with different TP values, but the ML mechanisms did not change dynamically the TP values. 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.

CONCLUSION

In this research work, LoRaWAN and ML has been studied in terms of classification for SF assignment in order to save module's energy requirements. LoRa is one promising wireless technology that deals with applications that need low latency, long range and low energy communication. We created a library that enables the communication between the OMNeT++ based LoRa simulator called FLoRa and the python based scikit 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. The simulation results revealed that classic classification algorithms can be used in the context of LoRa such as the k-NN, Gaussian Naïve Bayes classifier, the SVMs and the Decision trees, because achieve high scores in terms of accuracy, precision, recall, and F1. Then we tested the k-NN and Gaussian Naïve Bayes compared to two variants of the ADR in a simulated

Figure 9. Number of packets that GW did not receive, due to signal was weaker than the GW's sensitivity



LoRa deployment. The results showed that the ML based mechanisms performed in general better than the AvgADR and slightly worse than the MaxADR in terms of energy consumption and delivery ratio. The reason for first is that the classification error of the ML based mechanisms 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. Finally, the effectiveness of the classifiers is also presented in the confusion matrices, where in both urban and suburban cases.

FUTURE WORK

As far as the future work is concerned, we intent to integrate these ML mechanism to real nodes and conduct a small-scale experiment and a comparison with the simulation-based experiments, in order to validate our results. Moreover, it is intended to evaluate the above accuracy improvements in real life scenarios such as for Search and Rescue (SAR) operations, in the framework of WeSAR project. The evaluation will be conducted using hardware such as the Pycom modules (e.g. LoPy, FiPy) and the Dialog DA14861 wearable module. Furthermore, an investigation of the ML for LoRa network optimization will be conducted, based on the results of this paper. Most importantly, the study of the dynamic change of the TP values in par with the SF will be made. Last but not least, a method that could be used in LoRa, but in our best of knowledge has not been used in LoRa is Adaptive Distributed Artificial Intelligence (ADAI). Mukherjee, Goswami, Yan, Yang, & Rodrigues, (2019) have used ADAI technique with a hierarchical resource allocation strategy to address the issue of resource allocation in wireless sensor networks, showing better resource allocation in terms of energy consumption.

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

Adelantado, F., Vilajosana, X., Tuset-Peiro, P., Martinez, B., Melia-Segui, J., & Watteyne, T. (2017). Understanding the Limits of LoRaWAN. *IEEE Communications Magazine*, *55*(9), 34–40. doi:10.1109/MCOM.2017.1600613

Bor, M. C., Roedig, U., Voigt, T., & Alonso, J. M. (2016). Do LoRa Low-Power Wide-Area Networks Scale? *Proceedings of the 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, 59–67. doi:10.1145/2988287.2989163

Bouras, C., Gkamas, A., & Katsampiris Salgado, S. A. (2021). Energy efficient mechanism for LoRa networks. *Internet of Things*, *13*, 100360. doi:10.1016/j.iot.2021.100360

Bouras, C., Gkamas, A., Katsampiris Salgado, S. A., & Kokkinos, V. (2020). Comparison of LoRa Simulation Environments. In L. Barolli, P. Hellinckx, & T. Enokido (Eds.), *Advances on Broad-Band Wireless Computing, Communication and Applications* (Vol. 97, pp. 374–385). Springer International Publishing. doi:10.1007/978-3-030-33506-9_33

Bouras, C., Gkamas, A., Katsampiris Salgado, S. A., & Papachristos, N. (2021). Spreading Factor Analysis for LoRa networks: A supervised learning approach. WorldCist 2021, Azores, Portugal.

Bouras, C., Gkamas, A., Kokkinos, V., & Papachristos, N. (2020). Geolocation analysis for Search And Rescue systems using LoRaWAN. *International Journal of Communication Systems*, *4593*, e4593. Advance online publication. doi:10.1002/dac.4593

Buurman, B., Kamruzzaman, J., Karmakar, G., & Islam, S. (2020). Low-Power Wide-Area Networks: Design Goals, Architecture, Suitability to Use Cases and Research Challenges. *IEEE Access: Practical Innovations, Open Solutions*, *8*, 17179–17220. doi:10.1109/ACCESS.2020.2968057

Cui, S., & Joe, I. (2020). Collision prediction for a low power wide area network using deep learning methods. *Journal of Communications and Networks (Seoul)*, *22*(3), 205–214. doi:10.1109/JCN.2020.000017

Cuomo, F., Garlisi, D., Martino, A., & Martino, A. (2020). Predicting LoRaWAN Behavior: How Machine Learning Can Help. *Computers*, *9*(3), 60. doi:10.3390/computers9030060

Daramouskas, I., Kapoulas, V., & Paraskevas, M. (2019b). Using Neural Networks for RSSI Location Estimation in LoRa Networks. *2019 10th International Conference on Information, Intelligence, Systems and Applications (IISA)*, 1–7. doi:10.1109/IISA.2019.8900742

Daramouskas, I., Kapoulas, V., & Pegiazis, T. (2019a). A survey of methods for location estimation on Low Power Wide Area Networks. *2019 10th International Conference on Information, Intelligence, Systems and Applications (IISA)*, 1–4. doi:10.1109/IISA.2019.8900701

FLoRa simulator. (n.d.). https://flora.aalto.fi/

Li, Y., Yang, J., & Wang, J. (2020). DyLoRa: Towards Energy Efficient Dynamic LoRa Transmission Control. *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications*, 2312–2320. doi:10.1109/INFOCOM41043.2020.9155407

Liao, C.-H., Zhu, G., Kuwabara, D., Suzuki, M., & Morikawa, H. (2017). Multi-Hop LoRa Networks Enabled by *Concurrent Transmission. IEEE Access: Practical Innovations, Open Solutions*, *5*, 21430–21446. doi:10.1109/ACCESS.2017.2755858

Mukherjee, A., Goswami, P., Yan, Z., Yang, L., & Rodrigues, J. J. P. C. (2019). ADAI and Adaptive PSO-Based Resource Allocation for Wireless Sensor Networks. *IEEE Access: Practical Innovations, Open Solutions*, *7*, 131163–131171. doi:10.1109/ACCESS.2019.2940821

Mukherjee, A., Jain, D. K., & Yang, L. (2020). On-demand efficient clustering for next generation IoT applications: A Hybrid NN approach. *IEEE Sensors Journal*, 1–1. doi:10.1109/JSEN.2020.3026647

Park, G., Lee, W., & Joe, I. (2020). Network resource optimization with reinforcement learning for low power wide area networks. *J Wireless Com Network*, *2020*(1), 176. doi:10.1186/s13638-020-01783-5

Paul, B. (2020). A Novel Energy-Efficient Routing Scheme for LoRa Networks. *IEEE Sensors Journal*, *20*(15), 8858–8866. doi:10.1109/JSEN.2020.2983765

Routray, S. K., & Mohanty, S. (Eds.). (2021). *Principles and Applications of Narrowband Internet of Things (NBIoT)*. IGI Global., doi:10.4018/978-1-7998-4775-5

Sagir, S., Kaya, I., Sisman, C., Baltaci, Y., & Unal, S. (2019). 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)*, 68–71. doi:10.1109/ICDIPC.2019.8723666

Sandoval, R. M., Garcia-Sanchez, A.-J., & Garcia-Haro, J. (2019). Optimizing and Updating LoRa Communication Parameters: A Machine Learning Approach. *IEEE eTransactions on Network and Service Management*, *16*(3), 884–895. doi:10.1109/TNSM.2019.2927759

Slabicki, M., Premsankar, G., & Di Francesco, M. (2018). Adaptive configuration of lora networks for dense IoT deployments. *NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium*, 1–9. doi:10.1109/NOMS.2018.8406255

SMOTE-NC. (n.d.). https://imbalanced-learn.org/stable/generated/imblearn.over_sampling.SMOTENC.html

Tehrani, Y. H., Amini, A., & Atarodi, S. M. (2020). A Tree-structured LoRa Network for Energy Efficiency. *IEEE Internet of Things Journal*, 1–1. doi:10.1109/JIOT.2020.3034142

The Things Network. (n.d.). https://www.thethingsnetwork.org/docs/lorawan/adaptive-data-rate.html

Turmudzi, M., Rakhmatsyah, A., & Wardana, A. A. (2019). Analysis of Spreading Factor Variations on LoRa in Rural Areas. *2019 International Conference on ICT for Smart Society (ICISS)*, 1–4. doi:10.1109/ICISS48059.2019.8969846

Understanding the LoRA: Adaptive Data Rate. (n.d.). https://lora-developers.semtech.com/uploads/documents/files/Understanding_LoRa_Adaptive_Data_Rate_Downloadable.pdf

Vedaei, S. S., Fotovvat, A., Mohebbian, M. R., Rahman, G. M. E., Wahid, K. A., Babyn, P., Marateb, H. R., Mansourian, M., & Sami, R. (2020). COVID-SAFE: An IoT-Based System for Automated Health Monitoring and Surveillance in Post-Pandemic Life. *IEEE Access: Practical Innovations, Open Solutions*, *8*, 188538–188551. doi:10.1109/ACCESS.2020.3030194

Yatagan, T., & Oktug, S. (2019). Smart Spreading Factor Assignment for LoRaWANs. *2019 IEEE Symposium on Computers and Communications (ISCC)*, 1–7. doi:10.1109/ISCC47284.2019.8969608

Yu, Y., Mroueh, L., Li, S., & Terre, M. (2020). Multi-Agent Q-Learning Algorithm for Dynamic Power and Rate Allocation in LoRa Networks. *2020 IEEE 31st Annual International Symposium on Personal, Indoor and Mobile Radio Communications*, 1–5. doi:10.1109/PIMRC48278.2020.9217291

Zhu, G., Liao, C.-H., Sakdejayont, T., Lai, I.-W., Narusue, Y., & Morikawa, H. (2019). Improving the Capacity of a Mesh LoRa Network by Spreading-Factor-Based Network Clustering. *IEEE Access: Practical Innovations, Open Solutions*, *7*, 21584–21596. doi:10.1109/ACCESS.2019.2898239

Zourmand, A., Kun Hing, A. L., & Wai Hung, C. (2019). Internet of Things (IoT) using LoRa technology. *2019 IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS)*, 324–330. doi:10.1109/I2CACIS.2019.8825008

Christos Bouras is Professor in the University of Patras, Department of Computer Engineering and Informatics. Also, he is a scientific advisor of Research Unit 6 in Computer Technology Institute and Press - Diophantus, Patras, Greece. His research interests include Analysis of Performance of Networking and Computer Systems, Computer Networks and Protocols, Mobile and Wireless Communications, Telematics and New Services, QoS and Pricing for Networks and Services, e-learning, Networked Virtual Environments and WWW Issues. He has extended professional experience in Design and Analysis of Networks, Protocols, Telematics and New Services. He has published more than 400 papers in various well-known refereed books, conferences and journals. He is a co-author of 9 books in Greek and editor of 1 in English. He has been member of editorial board for international journals and PC member and referee in various international journals and conferences. He has participated in R&D projects.

Apostolos Gkamas obtained his Diploma, Master Degree and Ph.Dfrom the Computer Engineering and Informatics DepartmentOther site of Patras University (Greece)Other site. He is currently Associate Professor in University Ecclesiastical Academy of Vella, IoanninaOther site. His research interests include Computer Networks, Telematics, Multimedia transmission and Cross Layer Design. More particular he is engaged in transmission of multimedia data over networks and multicast congestion control. He has published more than 70 papers in international Journals and well-knownrefereed conferences. He is also co-author of three books (one with subject Multimedia and Computer Networks one with subject Special Network Issues and one with subject IPv6). He has participated in various R&D project (in both EU and national) such as IST, FP6, FP7, Intereg eLearning, PENED, EPEAEK, Information Society.

Spyridon Aniceto Katsampiris Salgado was born in 1996. He entered the Department of Computer Engineering and Informatics at University of Patras in 2014 and he received his diploma in 2019. Currently, he is a postgraduate student on Human-Computer Interaction. He joined the Laboratory of Distributed Systems and Telematics in 2018 and he is interested in machine learning, Internet of the Things and LPWA networks. Also, he has published research work on international conferences, and has participated in R&D projects.

Nikolaos Papachristos was born in 1993 in Patras, Greece. He entered the Department of Information and Communication System Engineering, University of the Aegean in 2011 and obtained his diploma in 2016. After that he obtained his master's degree in "Computer Science and Technology" in Computer Engineering and Informatics Department (University of Patras) in 2018. Currently he is a PhD student in the same department in the area of IoT using LoraWan & NB-IoT. Nikolas joined RU6 in 2015 and he is also interested in mobile telecommunications networks, mobile software development and D2D communications. He has obtained the Michigan Proficiency in English and the Zertifikat diplom in German. He has participated in many activities at the department of his studies as an IEEE Member, has taken part in IEEE's online contests and has participated in many online courses related to mobile software development as well as telecommunications networks for further education.