IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received May 5, 2021, accepted May 14, 2021, date of publication May 19, 2021, date of current version May 26, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3081794

Autonomous Configuration of Communication Systems for IoT Smart Nodes Supported by Machine Learning

ANDRÉ F. X. GLÓRIA, (Member, IEEE), AND PEDRO J. A. SEBASTIÃO, (Member, IEEE)
Instituto Universitário de Lisboa (ISCTE–IUL), 1649-026 Lisbon, Portugal
Instituto de Telecomunições (IT), 1049-001 Lisbon, Portugal

Corresponding author: André F. X. Glória (afxga@iscte-iul.pt)

This work was supported in part by the ISCTE–Instituto Universitário de Lisboa from Portugal under Project ISCTE-IUL-ISTA-BM-2018.

ABSTRACT Machine Learning brings intelligence services to IoT systems, with Edge Computing contributing for edge nodes to be part of these services, allowing data to be processed directly in the nodes in real time. This paper introduces a new way of creating a self-configurable IoT node, in terms of communications, supported by machine learning and edge computing, in order to achieve a better efficiency in terms of power consumption, as well as a comparison between regression models and between deploying them in edge or cloud fashions, with a real case implementation. The correct choice of protocol and configuration parameters can make the difference between a device battery lasting 100 times more. The proposed method predicts the energy consumption and quality of signal using regressions based on node location, distance and obstacles and the transmission power used. With an accuracy of 99.88% and a margin of error of 1.504 mA for energy consumption and 98.68% and a margin of error of 1.9558 dBm for link quality, allowing the node to use the best transmission power values for reliability and energy efficiency. With this it is possible to achieve a network that can reduce up to 68% the energy consumption of nodes while only compromising in 7% the quality of the network. Besides that, edge computing proves to be a better solution when energy efficient nodes are needed, as less messages are exchanged, and the reduced latency allows nodes to be configured in less time.

INDEX TERMS Wireless communications, edge computing, Internet of Things, machine learning, random forest, sustainability.

I. INTRODUCTION

The Internet of Things (IoT) relies on the devices ability to share information among them and with the cloud, for storage and data processing. With 50 billion devices expected to be connected by 2030, for the deployment of IoT system to be easily scalable, nodes need to be more autonomous.

Communication is a major component in IoT but it is also a power-hungry operation, mainly over large distances [1]. With the proliferation of Smart Cities, these networks of nodes need to be powered in a more efficient and green way [2]. As X. Fafoutis said [3] ''any data that is wrongly gathered, transmitted, stored or processed is a potential waste of energy''. Consequently, for the devices in higher numbers and powered by batteries, the end devices, a change in transmission and processing data is needed to have more sustainable networks.

Edge nodes benefit from the use of wireless transmission as they can be far away from the gateway, so they might need to transmit data for long distances. As they are often powered by batteries, low power transmissions are also needed. As they merely need to gather sensor data and transmit it to the gateway or any other node, the nodes are usually composed only by a microcontroller and a communication module. After this they enter in a deep-sleep mode to save energy until the next iteration, which allows them to be powered by batteries for long periods of time.

For that, advances in technology introduced the Low-Power Wide-Area Networks (LPWAN), such as LoRa or SigFox, with the ability to maintain power efficiency while transmitting over large distances [4]. These protocols can exchange messages between nodes kilometers apart, by adjusting their

The associate editor coordinating the review of this manuscript and approving it for publication was Zhaoqing Pan.

configuration parameters, mainly the transmission power, being this also responsible for managing the energy used in the communication. This means that a bad configuration of a LoRa transmission can create a node with 100 times less battery life [5].

To cope with this need to configure the best parameters, artificial intelligence and machine learning can help to decide which are the best configurations depending on the specification and conditions of the node. This allows for the creation of self-configurable nodes, that adjust these parameters in real time, as they need, without human or external intervention. This type of intelligence can even be more in line with the edge nodes, as the computational power of these devices increases, allowing intelligent methods to be done directly on the edge devices instead of using cloud computation. In a low-power fashion, this edge computation analysis can reduce the amount of messages traded for configuration, reducing the amount of energy to send and receive them, and also reduce the latency between the need for a new setup and the decision, as it is done directly on the node, being awake for shorter periods of time and once again reducing the energy needed and extending the battery life.

The literature shows some work being done to create a better LoRa link using Machine Learning techniques, with [6], using Dynamic Selection, with 96% efficiency and 47% energy savings, and [7] using Neural Networks to improve the energy efficiency of LoRa connection, with a 99.92% accuracy, but only with 200 samples. Some works were also found with the use of machine learning to predict the link quality of BLE mesh networks [8].

Previous research by the authors [2], proved that when working with the correct transmission power values, the network can save up to 73% of power while reducing the quality of service by 25%. Although some good results can be retrieved with that methodology, it used a cloud computing approach and only for LoRa, with multiple messages being sent from the node, varying the transmission power value to check what was the best one by a remote server. Not only can this compromise the network, as multiple messages can congest the service, it also takes more time and requires more energy in the configuration process.

All these researches proved that improvements can be done, mainly in porting the decision models directly to the edge nodes.

Therefore, the motivation to create a new methodology that is capable of creating more sustainable communication systems and smart nodes, that depend on those systems, is to improve the overall sustainability of smart cities project deployments by reducing the need for replacing batteries and increasing the efficiency of those systems.

To achieve that goal, this paper presents a methodology for an implementation of an autonomous configuration system for peer-to-peer communication in smart nodes supported by machine learning, that uses regressions to predict the energy consumption and link quality of a connection and then chooses the best protocol and transmission power to use. Besides the methodology and implementation for the autonomous configuration system, contributions in the area of the use of machine learning for the prediction of link quality and energy consumption of wireless communication systems in IoT nodes and a comparison of the use of these techniques in an edge or cloud approach are presented. For that, multiple regression techniques are compared, to choose the ones that best fits the methodology and obtains the best accuracy. It also compares an edge and cloud computing approach to check if the decision done directly on the edge node and in real time, without the need of sending any message or flooding the network. This could save energy and time in the decision process.

The paper starts with this introduction, followed by a description of the most used wireless protocols in IoT and how transmission power configuration affects the energy consumption of the edge nodes. The methodology for the system is presented, including the data analysis that will be performed to choose the best protocol and transmission power. Then, the research and study done on regression models are presented, along with the used techniques and methodology, as well as the training and validation results for the best regression model. Finally, the implementation scenario where the system was tested is described and the obtained results and conclusions are presented.

II. IoT COMMUNICATION PROTOCOLS

As said, communication is a major part of IoT systems and new advances in technology introduced wireless networks developed solely for IoT projects or low-power devices, such as LoRa, BLE or ZigBee, with Wi-Fi still having a major contribution and being modified to fit these new specifications. Since each project might need a specific protocol, due to location, implementation conditions or energy supply, the developed system will be available to work with the following technologies.

A. ESP-NOW

ESP-Now is a peer-to-peer wireless protocol developed by Espressif which enables multiple devices to communicate with each another without using Wi-Fi. The pairing between devices is needed prior to their communication, so after pairing a device with each other, the connection is persistent [9]. This means that if suddenly one of the boards loses power or resets, when it restarts, it will automatically connect to its peer to continue the communication. This protocol enables a low power consumption between multiple devices, being more power-efficient and faster to deploy when compared to Wi-Fi [10], supporting up to 20 nodes and being limited to 250 bytes packets.

B. BLUETOOTH LOW ENERGY

Bluetooth Low Energy (BLE) is an upgrade on the Bluetooth technology, designed to consume the least amount of energy while using the same wireless standard [11]. Working on the same 2.4GHz as Bluetooth, BLE reduced the high speed and high rate transmission, allowing a decrease in power consumption by up to 80% and increasing the range by 10 times, with connections up to 100 meters [12]. BLE also introduced mesh and star topologies, in a one-to-many fashion, contrary to the simple one-to-one connection provided by classic Bluetooth. As BLE is designed to broadcast short messages in close spaces, it has become one of the most used technologies in IoT projects.

C. LoRa

LoRa is a long range low power wireless technology that uses unlicensed radio spectrum, usually on the 868 or 915 MHz range, based on Chirp Spread Spectrum (CSS) modulation to allow the communication reach [13]. As it aims to eliminate repeaters, reduce device cost, increase battery life and support a large number of devices, it is the perfect solution for most IoT projects that rely on gathering data on large areas with low-power devices [2]. These features are possible since LoRa works on a star topology, reducing complexity and congestion in the network, allowing a viable low power long communication, with a single gateway covering up to hundreds of square kilometers [14].

D. ZigBee

ZigBee enables low-cost, low-power and low-data rate Machine-to-Machine communications for IoT networks [15], in the 868 MHz, 915Mhz and 2.4GHz frequency bands. Based on the IEEE 802.15.4 physical layer and the Medium Access Control sublayer, it is complemented by an application framework layer defined by the ZigBee Alliance [16]. Capable of working in mesh, star or cluster topology, ZigBee can achieve distances up to 100 meters, when working on the 2.4GHz frequency, or up to 1 kilometer, when using the lower frequencies [15]. ZigBee is the preferred solution for smart homes solutions due to its low power capabilities.

E. TRANSMISSION POWER IMPACT

Previous studies [2] showed that Transmission Power (TP), the value that directly impacts the range of communication, as it is the one responsible for defining the dBm used for transmission, is the one that mostly impacts the energy consumption of the device, and as such, is the one that needs to be configured accordingly to the node specifications and location.

According to the datasheet of several radio modules capable of transmitting in the presented wireless protocols, the power consumptions required for transmitting on different TP values are shown in Table 1.

Using a higher TP value does not always result in a better communication link, since multiple nodes transmitting in full power can generate interference in the network [2]. This shows that not only is it possible to reduce the energy consumption of the device, but also improve the network reliability, by adjusting the TP of each end node.

TABLE 1. Power consumption based on transmission power.

Protocol	P_{tx} [dBm]	I_{tx} [mA]
ESP-Now [17]	-	88
BLE [17]	-14 -3 7	60 100 150
LoRa [18]	5 14 23	20 45 120
ZigBee [19]	-7 1 8	15 25 40

III. METHODOLOGY

The presented methodology aims to create an autonomous solution, capable of selecting the best communication protocol and its transmission power configurations for a smart node, based on its location, the nearby gateways and geography (urban or rural areas, obstacles and distance to the gateway), supported by Machine Learning algorithms that can run directly on the node or with cloud communication. Fig. 1 shows the system methodology.

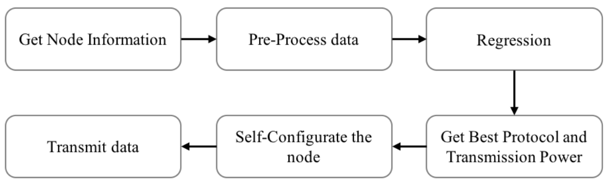

FIGURE 1. Point-to-point configuration methodology.

As it is possible to see, the methodology is divided into six steps. It starts with the node gathering its location based on GPS coordinates, and pre-process that data. After that, the learning algorithm makes a regression to predict the energy consumption and quality of the link for each of the available protocols and transmission power. Those predictions are then analyzed, and the best transmission power is selected, being the node configured, in an autonomous way, with that value. After that, the node is ready to send messages. From time to time, this process is repeated, to ensure the node is always working in the best conditions possible.

Regarding the data processing and regression algorithm, Fig. 2 shows the detailed process, from the data input to the output of the best protocol and transmission power value.

As said, it receives the node location and starts by comparing it to the list of available gateways, calculating for each one, a position (X,Y) facing the gateway in the center of a grid (0,0), the distance to the gateway and also any information about possible obstacles in the line of sight. With that information, it creates an array of data that will be used to predict the Received Signal Strength Indicator (RSSI) and energy consumption of the link while using those inputs for each of the communication protocols and transmission power values possible. This is done using a regression model.

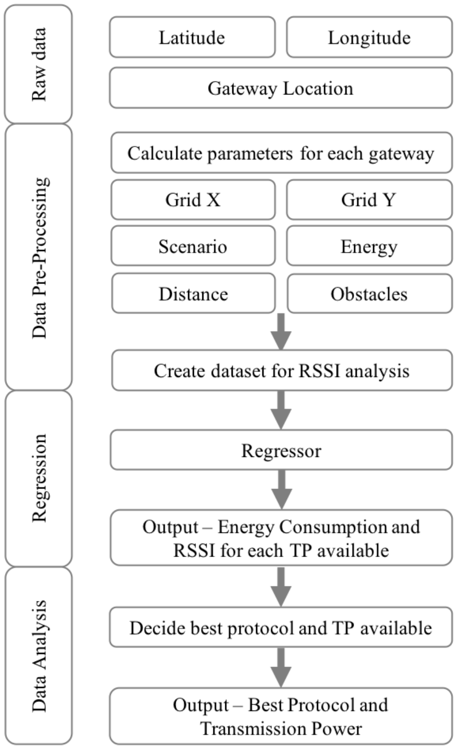

FIGURE 2. Point-to-point decision methodology.

After the regression, using the output values, three decision models are used to evaluate the best protocol and transmission power to use.

1) Best Link Model (BLM) - The TP is chosen solely based on the best link achieved, i.e., the one that gets the higher RSSI value. With this model, the link will always get the perfect conditions to ensure reliability, not considering the energy consumption. This mode can be used in nodes where information needs to be always delivered in real time, ensuring maximum reliability.
2) Energy Efficiency Model (EFM) - The TP is chosen based on the lowest energy used by a TP value capable of sustaining a communication link, even if the RSSI is higher or close to the threshold of the sensitivity on which each communication protocol can no longer transmit information, putting aside the reliability of the signal, to favor energy efficiency. This mode can be used in nodes where data is not sensitive and crucial, and if some packages are lost, it does not affect the system.
3) Reliable Link Model (RLM) - The TP is chosen based on the lowest energy used by a TP value capable of achieving a good communication link, i.e., with a RSSI close to -20dBm of the sensibility threshold, even if when using higher TP, a better RSSI values can be achieved. This mode is a middle solution between the previous two, it compromises some of the energy efficiency to guarantee a better connection.

IV. REGRESSION MODEL

As said before, our methodology uses a regression model for the computation analysis. For this model to work, it needs to be trained with previously known data and configured to achieve the best accuracy and lowest error possible. The next sections explain, in detail, how this process was done and the obtained model results.

The goal of this model is to receive as input the location of the node (X and Y) facing a gateway, the distance and obstacles towards the gateway and the protocols and TP to be tested, outputting the predicted RSSI and energy value for those conditions.

A. EDGE VS CLOUD COMPUTING

With the increase of IT solutions over the last decades, cloud computing (CC) services provide easy, high performance computation with a low investment on servers, since cloud computing works on remote central cloud base servers. As long as a device has an internet connection, it is able to send requests to the cloud and take advantage of all the available services, such as computing or storage, allowing also the connection between devices and services in different networks [20].

With the proliferation of IoT devices, the need to reduce latency in data analysis, in typical remote central cloud base computation services, allow the adaptation of end devices to perform some of this computation, thus creating the Edge Computation (EC) [21]. EC solves some challenges of managing systems on a central cloud, such as response time, security and quality of service, by executing tasks closer to the IoT devices, with management, storage, data analysis and decision making being done directly on multiple edge nodes inside the network [22]. These interconnected devices and EC techniques prevent overload of computer processes, as well as, obstructions in the flow of data and the services that are sent or requested to the cloud [23].

B. REGRESSION TECHNIQUES

Regression analysis is used as a technique for prediction, by searching the relationship between a dependent (*target*) and independent variable (*s*). As the training data is independently selected from the original dataset, the mean-squared error for a predictor variable X for the class y can be estimated by (1) [24].

$$E_{X,y}(y - X)^2 \tag{1}$$

There are multiple techniques for regressions in Machine Learning, being the most common Linear Regressions, Neural Networks, Support Vector Machine (SVM), Decision Trees or Random Forest.

1) LINEAR REGRESSION

Linear regression is the simplest method to predict data, being adopted by Machine Learning as they came from the statistical world. It performs the task to predict a dependable variable value (y) based on a given independent variable (x).

With this, it creates a simple linear relationship between the input and output, that can be presented in an equation as $y = A + B * x$, also called a plane or hyper-plane, being A and B the coefficients that the regression will define to characterize the output based on the input [25]. Besides being marked as inefficient and inaccurate by researchers [25] when compared to other ML techniques, is still commonly used for scoring modelling, since it gives a logistics distribution of the data.

2) DECISION TREE

Decision Trees (DT) are tree-based methods in which each path begins in a root node and multiple divisions are made, through a hierarchical partition of training data, taking into account the dataset, creating sub-trees based on a certain features used to split the data, representing a sequence of data divisions, with this split being done iteratively until it reach a leaf node with an outcome, containing the number of records that can be used to classify the data [25]. These methods can be applied for classification and regression. The final goal of this method is to reach a model that can predict the search value for that specific scenario by learning simple decision rules [26].

3) RANDOM FOREST

Random Forest (RF) is a decision tree method, developed for classification and regression [24] and is composed by a large number of trees, each voting for the final outcome, being the final result determined by a majority vote from all decision trees [27]. With a great performance in predictive tasks, it is ideal for analyzing large numbers of parameters [28], even with small datasets, being highly applicable for classification problems. RF incorporates the process of aggregation, bagging and Decision Trees, with a selection of a subset of features from each node of the tree, avoiding the correlation in the bootstrapped set [25]. The generated forest can also get great performance when new data is added [29].

4) NEURAL NETWORKS

Neural Networks (NN) algorithms are defined as computational models of neural systems composed of several neurons connected one to the other by synapses, in the same way as the human nervous system. Each of the neurons analyzes parts of the input and sends the information to the next layer and neurons continuously, until it is able to reach a valid output [25]. This process continues until a final output is found. It is ideally used in nonlinear and complex problems which requires large computational power and has some disadvantages when working with IoT systems due to low complex and low power devices [26]. In this case, Multilayer Perceptron (MLP), a variation of the NN algorithm which consists of multiple neurons organized into layers [30], was used. These MLP networks are characterized by being general-purpose, flexible and non-linear. Their complexity can be changed according to their application by varying the number of layers and units of each layer.

5) SUPPORT VECTOR MACHINES

Support Vector Machines (SVM) use a hyperplane to create a decision boundary in order to separate different classes of objects. Used for classification and regression, it uses complex mathematical functions to create this hyperplane and be able to assign the members of each class [31]. To discover the position of the hyperplane it uses a small subset of vectors from the training data, the support vectors, that define the edge of the class [32]. Most used for classification, since it can divide a dataset into classes, it lacks some probability estimate, mainly when large or more complex datasets are used [31].

C. DATASET

For a machine learning model to work, it needs to be trained with a set of supervised data, that include the output desired for a set of parameters. It is with these data that the model will learn and predict upon future data.

The used dataset is composed of data collected by the authors in an unpublished Point-to-Point protocol comparison study done in various environments, to check the performance of several wireless protocols in indoor and outdoor environments. The data contains the RSSI and power consumption values of the various protocol's transmissions performed by an IoT smart node in different scenarios around a single gateway, while varying the TP value, as well as the distance to the gateway and the number of obstacles in the line of sight. The dataset is publicly available at [33].

The dataset is composed of 18448 entries, with the following parameters:

- X – X grid position of the node facing the gateway at position (0,0);
- Y – Y grid position of the node facing the gateway at position (0,0);
- scenario – Characteristics of the transmission scenario (indoor, outdoor, . . .);
- distance – Distance, in meters and in line of sight, from the node to the gateway;
- obstacles – Number of obstacles, in line of sight, between the node and the gateway;
- protocol – Communication protocol used for the transmission;
- power – Transmission Power value used for the transmission;
- energy – Energy used to perform the transmission;
- rssi – RSSI value registered from the transmission;

D. TRAINING METHODOLOGY

In order to train the regression model the following steps were done, using Python, the scikit-learn libraries [34] and the Anaconda environment.

1) Each regression model was trained using the default configuration presented in their documentation and described in Appendix and Table 8. For that, the presented dataset was used for the training process.

With this, it is possible to compare the performance, in terms of margin of error, of each model, allowing us to assess which are more likely to guarantee best results and which need to be improved to achieve them.

2) The default model for each algorithm is then submitted into a hyper parametrization tuning, that trains the model with different configuration of the default parameters, as presented in Appendix and Table 9, in order to compare the model performance and assess which is the configuration that obtains the best performance, facing the dataset and the goal. For this, a method provided by scikit-learn called RandomizedSearchCV was used, which performs the fit and training of the algorithm under study, calculating which parameters are best suited to it [35];
3) In the final step, and to guarantee the validity of the model, the best configuration, obtained in the previous step, is submitted into a Stratified K-Fold cross validation. This allows us to understand if the model is under or over fitted. Using a set of five folds, each using a different part of the training dataset, allowing the model to check on every single datapoint, it is possible to validate the real model performance, as each of the folds will create a different outcome, that are averaged, allowing for a better perception of the error margin and variation, as more data is used to fit the model;
4) Finally, the best model is exported and ported to a C file using the micromlgen library [36], in order to be used on Edge Computing, or ported to a new Python file to be used on Cloud Computing.

E. RESULTS

The presented training methodology was followed in order to obtain the best model possible to predict the energy consumption and RSSI value of data transmissions, based on node position and the transmission power used.

To train, validate and test the model, the presented dataset will be used, being divided into three groups: 70% for training, 20% for validation and 10% for testing. To evaluate the model performance, and since a regression is used, the Mean Absolute Error (MAE) metric will be used, as it is the most common metric for regression. It measures the average absolute error between the real data and the estimated value, using (2) [37], where P_{rx} is the real value, $\hat{P}_{rx}$ is the estimated value, and N the number of samples.

$$MAE[dBm] = \frac{1}{N} \cdot \sum_{i=1}^{N} |P_{rx_i} - \hat{P}_{rx_i}| \quad (2)$$

The estimated data nearly matched the real data when MAE is near 0.

1) ENERGY CONSUMPTION

The results of the regression models to predict the energy consumption, for each step of the presented methodology, are displayed in Table 2 and allows us to conclude which is the best model to use in this scenario.

TABLE 2. Energy regression models MAE results.

Model	Default	Hyper parametrization	Cross-Validation
Linear Regression	27.595	27.217	27.358
Decision Tree	1.504	1.504	1.519
Random Forest	1.504	1.503	1.519
Neural Network	6.339	5.078	5.336
SVM	5.943	5.284	5.529

The first thing to notice, in all models, is that the hyper parametrization values always have the lower MAE, followed by the cross-validation values and finally the default values. This is justified by the methodology followed, as the default model, in the first step, is tuned to improve the default result, obtaining always a better result. Then, in the cross-validation step, as multiple combinations of the dataset are tested, and the MAE for each fold is averaged, it is expected to increase. Although this happens, the cross-validation results allow a better knowledge of the model accuracy, as it was exposed to a higher variety of unknown data.

Considering only the cross-validation results, as they are the ones best fitted to evaluate the model accuracy, the best regression model for predicting the energy consumption value is Random Forest. It achieves the lowest MAE among the tested models, with the same results as Decision Tree, being only 0.001 mA better on the hyper parametrization results. When compared to Neural Networks, SVM or Linear Regression, it achieved much better results, by over 3.5 mA, 3.8 mA and 26 mA, respectively.

Since Random Forest presents the best solution for this scenario and dataset, it will be the one used in the methodology. For that, the hyper parametrization tuning showed that the best configuration for the Random Forest regression model was with the following parameters:

- n_estimators – 127
- max_features – 'sqrt'
- max_depth – 70
- min_samples_split – 5
- min_samples_leaf – 1
- bootstrap – False

This model achieved a MAE of 1.503 mA and an accuracy of 99.88%. Fig. 3 shows the projected values facing the real values, obtained by the model.

This shows that the model can predict the energy consumption of a transmission with a 1.503 mA margin of error, which is an acceptable value. Also, as Fig. 3 shows, the projected values follow a proportional line, meaning that the model is well fitted for the dataset.

To further validate the model accuracy, and following the methodology presented, the model was validated with a

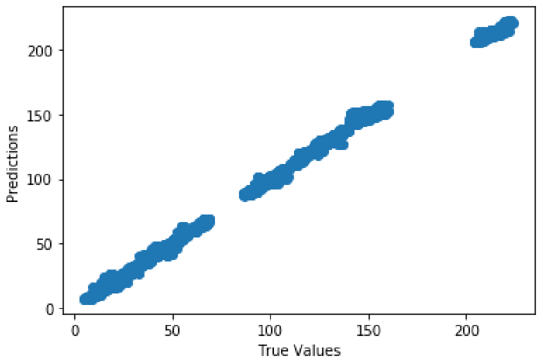

FIGURE 3. Energy Predicted vs Real values.

Stratified K-Fold Cross Validation, using 5 folds and 20% of the data as validation points. Fig. 4 shows the learning curve for the validation test.

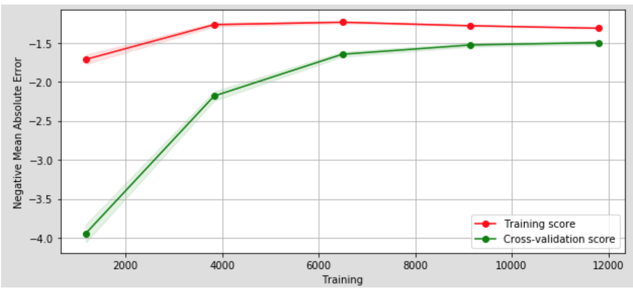

FIGURE 4. Energy regression learning curve.

It is possible to see, at the end of the learning curve, that both training and validation converge to a similar and lower MAE value, showing that the model is well fitted. The validation MAE was 1.519 mA, 0.015 mA higher than the training MAE, with an accuracy of 99.60%, 0.28% lower, being these values too small to be considered.

As such, is it possible to conclude that the trained model is well fitted and capable of predicting the energy consumption values of communication transmissions, based on location, distance and obstacles to the gateway, and the transmission power value, and can be ported to the node implementation.

2) RSSI

The results of the regression models to predict RSSI values, for each step of the presented methodology, are displayed in Table 3 and allows us to conclude which is the best model to use in this scenario.

As in the energy consumption scenario, only the cross-validation results will be considered. For predicting the RSSI value the best regression model is also Random Forest. It achieves the lowest MAE among the tested models, with results close to Decision Tree, being only 0.05 dBm better, but achieving results much better than Neural Networks, SVM or Linear Regression, by over 3 dBm, for both NN and SVM, and 11 dBm, for LR.

TABLE 3. Regression models MAE results.

Model	Default	Hyper parametrization	Cross-Validation
Linear Regression	13.223	13.223	13.161
Decision Tree	2.149	2.068	2.081
Random Forest	2.019	1.955	2.031
Neural Network	5.301	4.961	6.606
SVM	5.717	4.942	5.389

Since Random Forest presents the best solution for this scenario and dataset, it will be the one used in the methodology. For that, the hyper parametrization tuning, showed that the best configuration for the Random Forest regression model was with the following parameters:

- n_estimators – 157
- max_features – 'auto'
- max_depth – 90
- min_samples_split – 5
- min_samples_leaf – 1
- bootstrap – True

This model achieved a MAE of 1.9558 dBm and an accuracy of 98.68%. Fig. 5 shows the projected values facing the real values, obtained by the model.

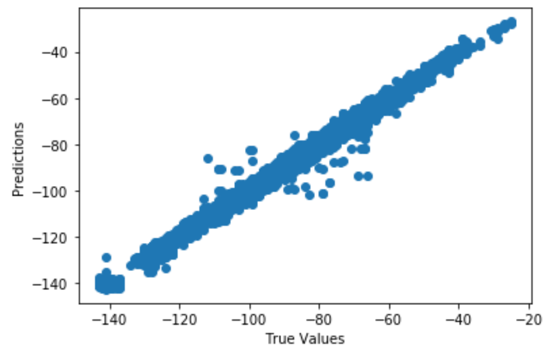

FIGURE 5. RSSI Predicted vs Real values.

This shows that the model can predict the RSSI of a transmission with a 1.9558 dBm margin of error, which is an acceptable value. Also, as Fig. 5 shows, the projected values follow a proportional line, meaning that the model is well fitted for the dataset.

Following the methodology model accuracy was validated with a Stratified K-Fold Cross Validation, using 5 folds and 20% of the data as validation points. Fig. 6 shows the learning curve for the validation test.

It is possible to see, at the end of the learning curve, that both training and validation converge to a similar and lower MAE value, showing that the model is well fitted. The validation MAE was 2.031 dBm, 0.07 dBm higher than the training MAE, with an accuracy of 98.42%, 0.25% lower, being these values too small to be considered.

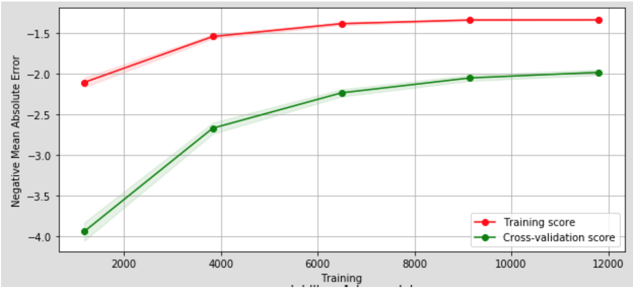

FIGURE 6. RSSI regression learning curve.

As such, it is possible to conclude that the trained model is well fitted and capable of predicting the RSSI values of communication transmissions, based on location, distance and obstacles to the gateway, as well as the transmission power value, being then capable of being ported to the node implementation.

V. IMPLEMENTATION

To compare if the autonomous configuration system works better using an edge or cloud computing approach, the presented methodology was implemented in a smart node, Fig. 7, capable of transmitting with all the communication protocols studied.

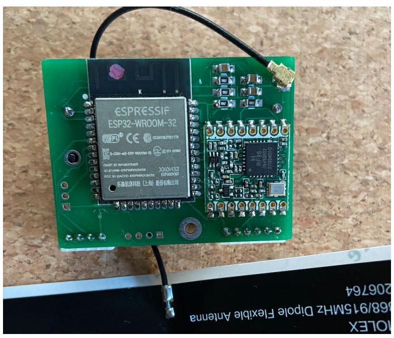

FIGURE 7. Smart node.

Several nodes were deployed around one gateway, in an urban environment, covering an implementation area of 36ha, with a radius of 600 meters from the gateway position, as shown in Fig. 8, where the red dot represents the gateway and the green dots represent the smart node locations.

Fig. 9 represents the specifications of each node location, with grid position (X,Y), distance and number of obstacles.

The nodes were deployed for a period of one and a half months, fifteen days using the BLM scenario, fifteen days using the EML scenario and the other fifteen using the RLM scenario, being the ones working on the edge computing model self-configured every 48 hours, or if any message was not able to be delivered, since these do not need external messages and require less energy, and the ones working with the cloud computing model self-configured once a week, since they need to transmit a message via the gateway to the cloud asking for the configuration parameters, therefore needing more power to perform this task.

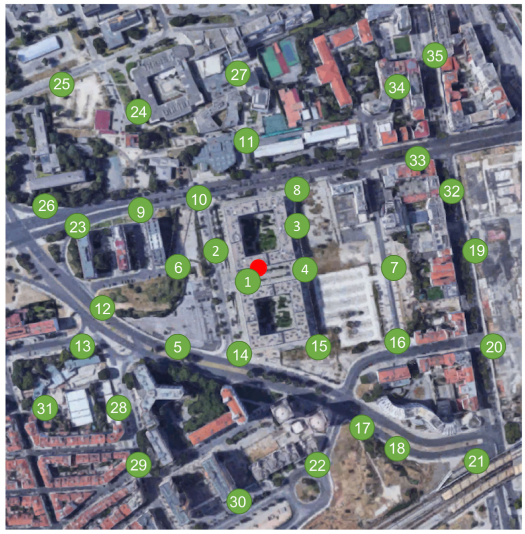

FIGURE 8. Smart node locations.

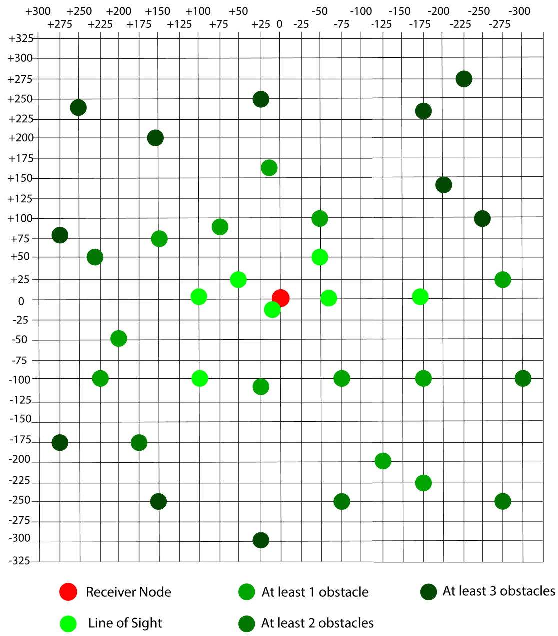

FIGURE 9. Locations specifications.

VI. RESULTS

After one and half months of deployment, running all the scenarios described, the results were analyzed and are outlined using a scatter plot with a linear distribution, showing the average Transmission Power and RSSI used by each node during the implementation period, as well as the chosen protocol, for both edge and cloud models. Table 4 summarize the obtained results.

A. BEST LINK MODEL SCENARIO

As described in the methodology, the Best Link Model (BLM) chooses the protocol and transmission power based solely on the best link achieved, i.e., the one that gets the highest RSSI value.

TABLE 4. Implementation results.

	Model	Protocol	P_{tx} [dBm]	RSSI [dBm]	I_{tx} [mA]
BLM	EC	LoRa (92%)	20 (-)	-115.36 (-)	90 (-)
	CC	LoRa (93%)	20 (-[2])	-117.48 (+2%[2])	112 (+24%[2])
EFM	EC	LoRa (87%)	10 (-50%[1])	-130.25 (+13%[1])	29 (-68%[1])
	CC	LoRa (84%)	11 (-45%[1] ‖ +10%[2])	-123.88 (+5%[1] ‖ -5%[2])	43 (-62%[1] ‖ +48%[2])
RLM	EC	LoRa (98%)	14 (-30%[1])	-124.15 (+8%[1])	50 (-45%[1])
	CC	LoRa (96%)	16 (-20%[1] ‖ +14%[2])	-120.24 (+2%[1] ‖ -3%[2])	68 (-39%[1] ‖ +36%[2])

1 - Facing the BLM scenario under the same model
2 - Facing the EC model under the same scenario

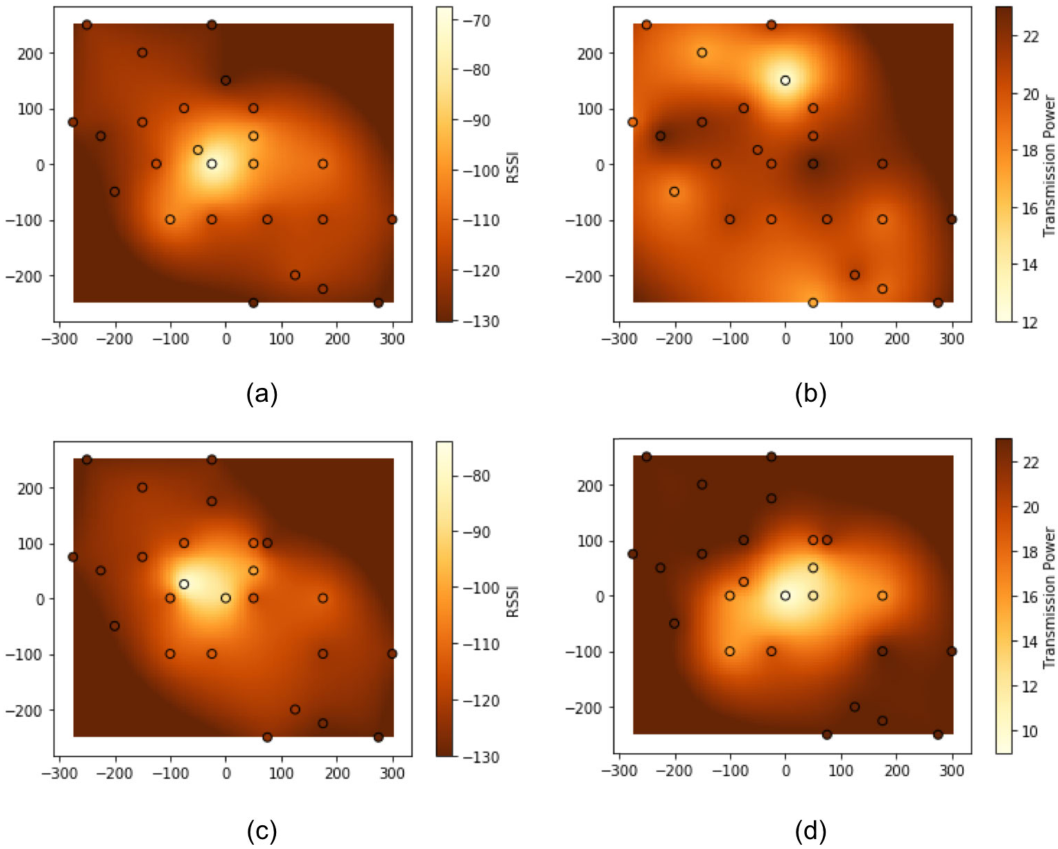

FIGURE 10. BLM scenario results. (a) Edge computing RSSI; (b) Edge computing transmission power; (c) Cloud computing RSSI; (d) Cloud computing transmission power.

In this scenario, both edge and cloud computing models choose LoRa as the main protocol to use, in 92% and 93% of the cases, respectively. For edge and cloud computing, the remaining cases used Zigbee. These variations in the selection of the communication protocols come in line with LoRa being the best overall protocol, as studied by the authors in [38], for urban environment and long distances, with the other protocols being selected only for close range nodes without obstacles.

The LoRa scenarios, being the vast majority of cases, will be the only ones analyzed in this scenario. The results obtained for RSSI and transmission power, for both edge and cloud computing, can be found in Fig. 10, that shows each of the location tested and the obtained RSSI and transmission power used and a linear interpolation and a colorbar of the values to assess the distribution. Table 5 shows the average values for this scenario.

TABLE 5. BLM scenario results.

	P_{tx} [dBm]	RSSI [dBm]	I_{tx} [mA]
Edge Computing	20	-115.36	90
Cloud Computing	20	-117.48	112

It is possible to check, Fig. 10 (a), that when using Edge Computing alongside the BLM mode, the closest nodes transmit with an average value of −85 dBm, while mid-range and further nodes transmit with an average −110 dBm and −125 dBm, respectfully. For the same distance, the average TP value, Fig. 10 (b), is 18, 20 and 22, respectfully.

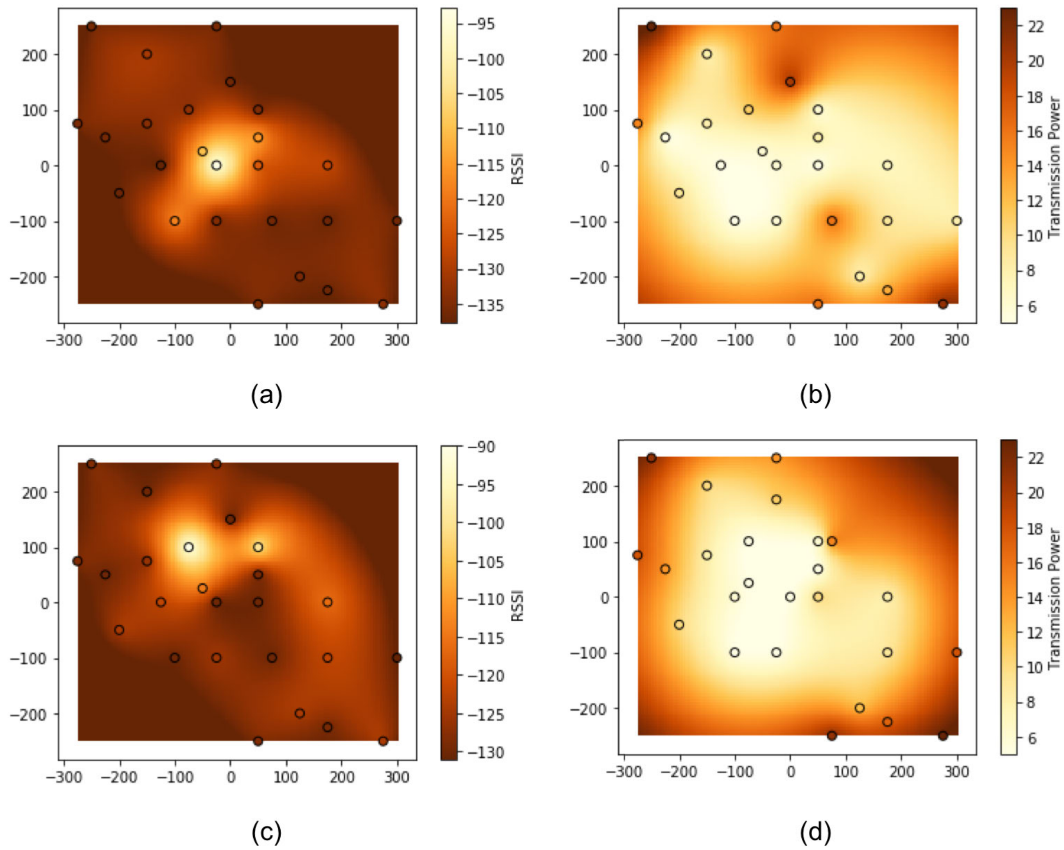

FIGURE 11. EFM scenario results. (a) Edge computing RSSI; (b) Edge computing transmission power; (c) Cloud computing RSSI; (d) Cloud computing transmission power.

One thing that is interesting to see, is that some mid-range nodes can transmit with an average of 14 dBm of transmission power. Also, the further nodes with more obstacles were not able to create a connection.

As for the Cloud Computing model, when associated with BLM mode, in terms of RSSI, it has similar results as the Edge Computing model, as can be seen when comparing Fig. 10 (c) and Fig. 10 (a), with the connection link being only 2% worst, about -2 dBm. In terms of transmission power, although both models achieve an average of 20 dBm, when looking to Fig. 10 (d) and Fig. 10 (b), it is possible to check that, in the Cloud Computing model, the further nodes have higher transmission power, while the Edge Computing presents a more even distribution between all nodes.

In terms of energy consumption, the nodes using the Edge Computing model used less 24% power facing the Cloud Computing nodes. As such, comparing the two models it is possible to check that the edge computing achieves better results on all fields, using 20 mA less, while increasing the quality of the link connection by 2 dBm. So, a better connection can be achieved with a lower power consumption.

B. ENERGY EFFICIENCY MODEL SCENARIO

The Energy Efficiency Model (EFM) chooses the protocol and transmission power based on the lowest energy value of a transmission power that can achieve a connection.

As in the BLM mode, in this scenario, both edge and cloud computing models choose LoRa as the main protocol to use, although in a smaller value, with 87% and 84% of the cases, respectively. For edge computing, the remaining cases used Zigbee, 6%, BLE, 5%, and ESP-Now, 2%, as for cloud computing, Zigbee, 9%, BLE, 5%, and ESP-Now, 2%, were the selected ones. Once again, the variations in the protocol chosen are accounted for only in the nodes in close range, without obstacles.

The results obtained for RSSI and transmission power, for both edge and cloud computing, can be found in Fig. 11, being once again presented only the LoRa results for each of the tested locations including the RSSI and transmission power used and a linear interpolation and a colorbar of the values to assess the distribution. Table 6 shows the average values for this scenario.

TABLE 6. EFM scenario results.

	P_{tx} [dBm]	RSSI [dBm]	I_{tx} [mA]
Edge Computing	10	-130.25	29
Cloud Computing	11	-123.88	43

The results show that when using Edge Computing alongside EFM mode, the quality of the signal decreases while lower transmission power values are used, facing the BLM mode. Fig. 11 (a) shows that the closest nodes transmit with

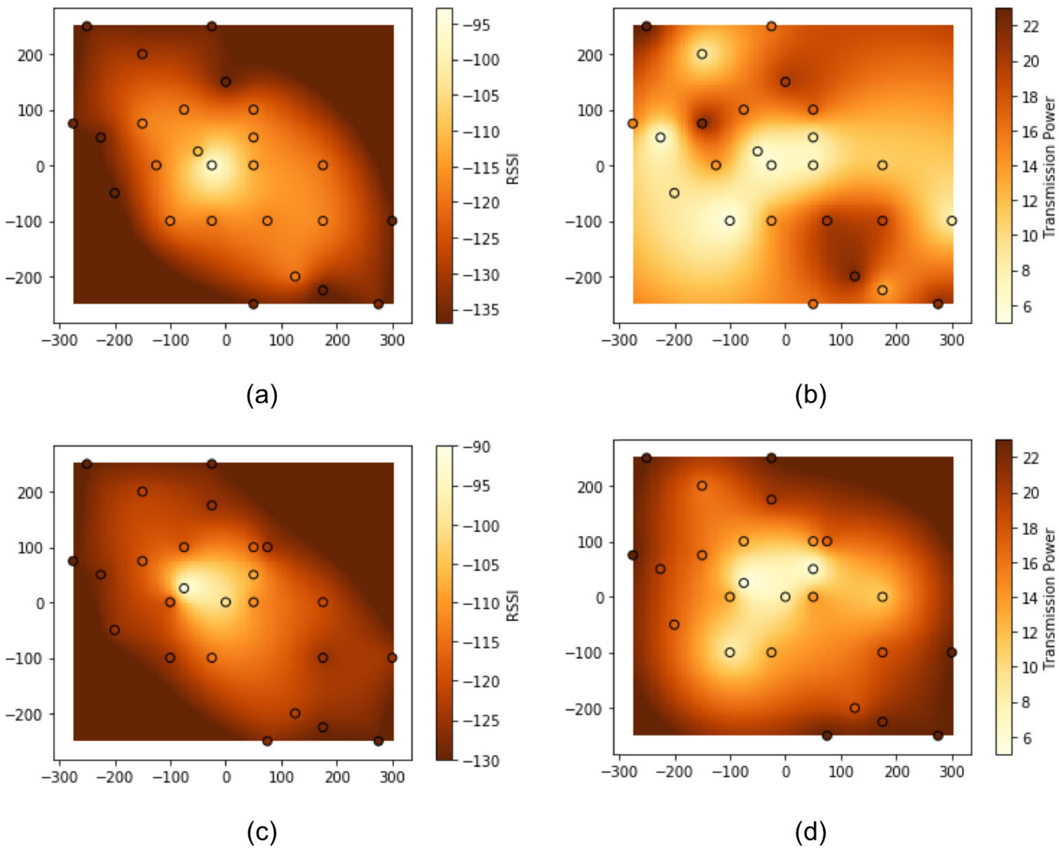

FIGURE 12. RLM scenario results. (a) Edge computing RSSI; (b) Edge computing transmission power; (c) Cloud computing RSSI; (d) Cloud computing transmission power.

an average value of −115 dBm, while mid-range and further nodes transmit with an average −123 dBm and −131 dBm, respectfully. For the same distance, Fig. 11 (b), the average transmission power value is 6, 10 and 17, respectfully. As in the previous scenario, the further nodes with more obstacles were not able to create a connection.

In the Cloud Computing model, as in the BLM scenario, a similar behavior can be found when comparing with the Edge Computing, Fig. 11 (c). Contrary to the BLM scenario, in the EFM mode, this model achieved a better link connection, being 7 dBm higher, or −5%. This is partially caused by the higher transmission power being used by the Cloud Computing model, Fig. 11 (d), that is, on average, 1 dBm higher than the Edge Computing, allowing a better signal quality.

As for energy consumption, the Cloud Computing, with a higher transmission power, consumed on average 43mA, which is 48% more than the Edge Computing model. With that in mind, the Edge Computing model, besides the 5% lower link quality, presents once again the best alternative, mainly when considering this is an energy efficient mode focused only on decreasing the power consumption. So, a worst connection can be achieved but it is done with a lower power consumption.

C. RELIABLE LINK MODEL SCENARIO

Finally, the Reliable Link Model (RLM) compromised some of the energy efficiency of the ELM model and the link quality of the BLM model, choosing the protocol and transmission power based on the lowest energy value of a transmission power that can achieve a good connection, i.e., close to −20 dBm of the sensibility threshold for that network.

Following the path of the previous modes, in this scenario, both edge and cloud computing models choose LoRa as the main protocol to use, with a higher value, 98% and 96% of the cases, respectively. For edge and cloud computing, the remaining cases used Zigbee.

As for the previous scenarios, only the LoRa results will be analyzed, with the obtained for RSSI and transmission power results, for both edge and cloud computing, presented in Fig. 12, that shows each of the location tested and the obtained results with a linear interpolation and a colorbar of the values to assess the distribution. Table 7 shows the average values for this scenario.

TABLE 7. RLM scenario results.

	P_{tx} [dBm]	RSSI [dBm]	I_{tx} [mA]
Edge Computing	14	-124.15	50
Cloud Computing	16	-120.24	68

TABLE 8. Default configuration parameters.

Algorithm	Parameter Configuration
Linear Regression	$fit\_intercept = True, normalize = False, copy\_X = True, positive = False$
Random Forest	$n_{e}stimators = 100, criterion =' mse', max_{d}epth = None,$ $min_{s}amples_{s}plit = 2, min_{s}amples_{l}eaf = 1, min_{w}eight_{f}raction_{l}eaf = 0.0,$ $max_{f}eatures =' auto', max_{l}eaf_{n}odes = None, min_{i}mpurity_{d}ecrease = 0.0,$ $min_{i}mpurity_{s}plit = None, bootstrap = True, oob_{s}core = False, random_{s}tate = None,$ $verbose = 0, warm_{s}tart = False, ccp_{a}lpha = 0.0, max_{s}amples = None$
Neural Network	$hidden\_layer\_sizes = 100, activation =' relu', solver =' adam',$ $alpha = 0.0001, batch\_size =' auto', learning\_rate =' constant',$ $learning\_rate\_init = 0.001, power\_t = 0.5, max\_iter = 200, shuffle = True,$ $random\_state = None, tol = 0.0001, verbose = False, warm\_start = False,$ $momentum = 0.9, nesterovs\_momentum = True, early\_stopping = False,$ $validation\_fraction = 0.1, beta\_1 = 0.9, beta\_2 = 0.999, epsilon = 1e-08,$ $n\_iter\_no\_change = 10, max\_fun = 15000$
SVM	$penalty =' l2', loss =' squared\_hinge', dual = True, tol = 0.0001,$ $C = 1.0, multi\_class =' ovr', fit\_intercept = True, intercept\_scaling = 1,$ $class\_weight = None, verbose = 0, random\_state = None, max\_iter = 1000$
Decision Tree	$criterion =' mse', splitter =' best', max\_depth = None,$ $min\_samples\_split = 2, min\_samples\_leaf = 1, min\_weight\_fraction\_leaf = 0.0,$ $max\_features = None, random\_state = None, max\_leaf\_nodes = None,$ $min\_impurity\_decrease = 0.0, min\_impurity\_split = None, ccp\_alpha = 0.0$

When using the Edge Computing model with the RLM mode, as in the EFM mode, the quality of the signal decreases while lower transmission power values are used, facing the BLM mode, but achieves a better quality of signal with slightly higher transmission power values, facing the EFM mode. The closest nodes, Fig. 12 (a), transmit with an average value of −105 dBm, while mid-range and further nodes transmit with an average −115 dBm and −128 dBm, respectfully. For the same distance, Fig. 12 (b), the average transmission power value is 7, 12 and 18, respectfully.

As in BLM mode, one thing that is interesting to see is that some mid-range nodes can transmit with a higher transmission power value, with an average of 22 dBm of transmission power. As in the previous scenarios, the further nodes with more obstacles were not able to create a connection.

Regarding the Cloud Computing model, as in the previous modes, it follows the Edge Computing in terms of behavior, Fig. 12 (c), having a better communication link, 3% higher, than the Edge Computing model. This, as in the EFM mode, can be justified by the use of a higher transmission power, Fig. 12 (d), in this case by 2 dBm. Also, when comparing the transmission power results, it is possible to check that, as in the BLM mode, the further nodes have a higher transmission power than the Edge Computing nodes, that have a more even distribution.

This higher transmission power, as in the EFM mode, draws more power from the nodes, consuming 36% more than the Edge Computing nodes. Considering this and the slight difference between link quality, of only 3%, the Edge Computing gets, once again, the advantage facing the Cloud Computing. So, a slightly worse connection can be achieved, but is it done with a lower power consumption.

VII. CONCLUSION

This paper presents a methodology for an autonomous implementation of a self-configuring smart node supported by machine learning, that uses regressions to predict the link quality of a connection and then chooses the best Transmission Power and communication protocol to use. The methodology for the system was presented, as well as the training and validation of the computing model, including a comparison between various regression techniques. A practical implementation of the methodology is also presented, not only to validate the methodology, but also to compare how an edge or cloud model affects the system accuracy and the energy consumption of the edge nodes. For all these scenarios the results were presented.

The first thing to conclude is that with proper configuration, the smart node can act in a lower power fashion, creating more sustainable networks.

Regarding the computation model, several regression techniques, including Linear Regressions, Decision Trees, Random Forest, SVM and Neural Networks were tested, to assess which performs better in terms of predicting the energy consumption and RSSI of a wireless communication, based on the location of the node, distance and obstacles to the gateway and the transmission power value. It was possible to conclude that Random Forest was the best solution for both scenarios, achieving an accuracy of 99.88% and 98.68%, with a margin of error of 1.504 mA and 1.9558 dBm, respectively for energy and RSSI prediction. Random Forest results surpass Linear Regression, SVM and Neural Network by almost 26mA, 3.8mA and 3.2mA, for energy prediction, and 11 dBm and 3 dBm, for RSSI prediction, respectively, being Decision Trees the only technique that provides results more similar

TABLE 9. RandomizedCV configuration parameters.

Algorithm	Parameter Configuration
Linear Regression	$fit\_intercept = [True, False], normalize = [True, False], copy\_X = [True, False]$
Random Forest	$n\_estimators = [10, 20, .., 190, 200], max\_features = ['auto',' sqrt'], max\_depth = [10, 20, .., 90, 100],$ $min\_samples_split = [2, 5, 10], min\_samples_leaf = [1, 2, 4], bootstrap = [True, False], criterion = ['mse',' mae']$
Neural Networks	$activation = ['identity',' logistic',' tanh',' relu'], solver = ['lbfgs',' sgd',' adam'],$ $alpha = [0.0001, 0.0005, .., 0.005, 0.01]$
SVM	$penalty = ['l1',' l2'], loss = ['hinge',' squared\_hinge'], fit\_intercept = [True, False],$ $dual = [True, False], multi\_class = ['ovr',' crammer_singer']$
Decision Tree	$splitter = ['best',' random'], max\_features = ['auto',' sqrt'], max\_depth = [10, 20, .., 90, 100],$ $min\_samples\_split = [2, 5, 10], min_samples_leaf = [1, 2, 4], criterion = ['mse',' friedman_mse']$

to Random Forest, being only 0.05 dBm worst on RSSI and having the exact same results in energy prediction.

One of the other goals of this research was to understand how the edge computing methodology faces a cloud computing methodology for deciding which is the best protocol and transmission power value for a smart node to transmit messages. The edge computing methodology can achieve better results, while sometimes having a lower quality of service, although only by a slight margin, proving to be a better solution, since it takes less time to decide and configure the node, being done locally and without external inputs.

Finally, three modes of selecting the best transmission power value were presented, each with a specific role depending on the needs of the network. The BLM mode proves to be reliable, with the best quality link being achieved, with higher RSSI values, but with more energy being used. The EFL mode proves to be a solution for low power nodes, using 68% less energy but compromising the reliability of the network by 13%. The RLM mode is a balanced solution, it can save 45% more energy than the BLM mode with a 7% better quality than the EFL mode.

By applying this methodology to a network, not only can it extend the life cycle of the nodes but also reduce the need for maintenance and interference between nodes, creating a more sustainable network.

APPENDIX

REGRESSION MODEL CONFIGURATIONS

To train each regression model using the presented methodology, the default configuration of each model was used, according to their documentation. Table 8 contains the parameters of each model.

For the second stage of the training methodology, the RandomizedCV method was used to test multiple combinations of the model parameters configuration, in order to find the best one possible. For that, the combinations presented in Table 9 were used.

REFERENCES

[1] Z. Dawy, W. Saad, A. Ghosh, J. G. Andrews, and E. Yaacoub, ''Toward massive machine type cellular communications,'' *IEEE Wireless Commun.*, vol. 24, no. 1, pp. 120–128, Feb. 2017.

[2] A. Gloria, C. Dionisio, G. Simoes, and P. Sebastiao, ''LoRa transmission power self con-Guration for low power end devices,'' in *Proc. 22nd Int. Symp. Wireless Pers. Multimedia Commun. (WPMC)*, Nov. 2019, pp. 1–6.

[3] X. Fafoutis, L. Marchegiani, A. Elsts, J. Pope, R. Piechocki, and I. Craddock, ''Extending the battery lifetime of wearable sensors with embedded machine learning,'' in *Proc. IEEE World Forum Internet Things*, May 2018, pp. 269–274.

[4] V. M. Suresh, R. Sidhu, P. Karkare, A. Patil, Z. Lei, and A. Basu, ''Powering the IoT through embedded machine learning and LoRa,'' in *Proc. IEEE World Forum Internet Things*, May 2018, pp. 349–354.

[5] M. Bor and U. Roedig, ''LoRa transmission parameter selection,'' in *Proc. 13th Int. Conf. Distrib. Comput. Sensor Syst. (DCOSS)*, Jun. 2017, pp. 27–34.

[6] A. Gupta and M. Fujinami, ''Battery optimal configuration of transmission settings in LoRa moving nodes,'' in *Proc. 16th IEEE Annu. Consum. Commun. Netw. Conf. (CCNC)*, Jan. 2019, pp. 1–6.

[7] H. Yan and H. Hu, ''Study on energy saving algorithm of LoRa terminal based on neural network,'' in *Proc. 3rd Int. Conf. Smart City Syst. Eng. (ICSCSE)*, Dec. 2018, pp. 908–911.

[8] R. Li, X. Li, and Y. Ding, ''Link prediction algorithm for BLE mesh network in health monitoring system,'' in *Proc. Chin. Control Decis. Conf. (CCDC)*, Aug. 2020, pp. 1997–2001.

[9] (2021). *ESP-Now Overview*. Accessed: Oct. 3, 2021. [Online]. Available: https://www.espressif.com/en/products/software/esp-now/overview

[10] T. N. Hoang, S.-T. Van, and B. D. Nguyen, ''ESP-NOW based decentralized low cost voice communication systems for buildings,'' in *Proc. Int. Symp. Electr. Electron. Eng. (ISEE)*, Oct. 2019, pp. 108–112.

[11] F. J. Dian and R. Vahidnia, ''Formulation of BLE throughput based on node and link parameters,'' *Can. J. Electr. Comput. Eng.*, vol. 43, no. 4, pp. 261–272, 2020.

[12] K. E. Jeon, J. She, P. Soonsawad, and P. C. Ng, ''BLE beacons for Internet of Things applications: Survey, challenges, and opportunities,'' *IEEE Internet Things J.*, vol. 5, no. 2, pp. 811–828, Apr. 2018.

[13] A. J. Wixted, P. Kinnaird, H. Larijani, A. Tait, A. Ahmadinia, and N. Strachan, ''Evaluation of LoRa and LoRaWAN for wireless sensor networks,'' in *Proc. IEEE Sensors*, May 2017, pp. 5–7.

[14] LoRa Alliance. (2015). *LoRaWANT Specification*. Accessed: Oct. 3, 2021. [Online]. Available: https://www.lora-alliance.org/portals/0/specs/LoRaWANSpecification1R0.pdf

[15] A. I. Ali, S. Z. Partal, S. Kepke, and H. P. Partal, ''ZigBee and LoRa based wireless sensors for smart environment and IoT applications,'' in *Proc. 1st Global Power, Energy Commun. Conf. (GPECOM)*, Jun. 2019, pp. 19–23.

[16] ZigBee Alliance. (2015). *ZigBee Specifications*. Accessed: Nov. 3, 2021.[Online]. Available: https://zigbeealliance.org/wp-content/uploads/2019/11/docs-05-3474-21-0csg-zigbee-specification.pdf

[17] Espressif Systems. (2018). *ESP32 Series Datasheet*. Accessed: Oct. 3, 2021. [Online]. Available: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf

[18] (2019). *HopeRF.RFM95W Datasheet*. Accessed: Jun. 3, 2021. [Online]. Available: https://cdn.sparkfun.com/assets/learn_tutorials/8/0/4/RFM95_96_97_98W.pdf

[19] Digi. (2021). *XBee 3—RF Module Datasheet*. Accessed: Nov. 3, 2021. [Online]. Available: https://www.digi.com/resources/documentation/digidocs/pdfs/90001543.pdf

[20] P. R. Kumar, P. H. Raj, and P. Jelciana, "Exploring data security issues and solutions in cloud computing," *Procedia Comput. Sci.*, vol. 125, pp. 691–697, May 2018.
[21] W. Rafique, L. Qi, I. Yaqoob, M. Imran, R. U. Rasool, and W. Dou, "Complementing IoT services through software defined networking and edge computing: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 3, pp. 1761–1804, 3rd Quart., 2020.
[22] I. Sittón-Candanedo, R. S. Alonso, Ó. García, A. B. Gil, and S. Rodríguez-González, "A review on edge computing in smart energy by means of a systematic mapping study," *Electronics*, vol. 9, no. 1, p. 48, Dec. 2019.
[23] I. Sittón-Candanedo, R. S. Alonso, J. M. Corchado, S. Rodríguez-González, and R. Casado-Vara, "A review of edge computing reference architectures and a new global edge proposal," *Future Gener. Comput. Syst.*, vol. 99, pp. 278–294, Oct. 2019.
[24] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.
[25] R. Saravanan and P. Sujatha, "A state of art techniques on machine learning algorithms: A perspective of supervised learning approaches in data classification," in *Proc. 2nd Int. Conf. Intell. Comput. Control Syst. (ICICCS)*, Jun. 2018, pp. 945–949.
[26] M. Mamdouh, M. A. I. Elrukhsi, and A. Khattab, "Securing the Internet of Things and wireless sensor networks via machine learning: A survey," in *Proc. Int. Conf. Comput. Appl. (ICCA)*, Aug. 2018, pp. 215–218.
[27] H. Luo, X. Pan, Q. Wang, S. Ye, and Y. Qian, "Logistic regression and random forest for effective imbalanced classification," in *Proc. IEEE 43rd Annu. Comput. Softw. Appl. Conf. (COMPSAC)*, Jul. 2019, pp. 916–917.
[28] M. Kayri, I. Kayri, and M. T. Gencoglu, "The performance comparison of multiple linear regression, random forest and artificial neural network by using photovoltaic and atmospheric data," in *Proc. 14th Int. Conf. Eng. Modern Electric Syst. (EMES)*, Jun. 2017, pp. 1–5.
[29] J. K. Jaiswal and R. Samikannu, "Application of random forest algorithm on feature subset selection and classification and regression," in *Proc. 2nd World Congr. Comput. Commun. Technol.*, Oct. 2017, pp. 65–68.
[30] T. Hastie, R. Tibshirani, and J. Friedman, *Springer Series in Statistics The Elements of Statistical Learning Data Mining, Inference, and Prediction*, 2nd ed. New York, NY, USA: Springer-Verlag, 2008.
[31] S. Ray, "A quick review of machine learning algorithms," in *Proc. Int. Conf. Mach. Learn., Big Data, Cloud Parallel Comput. (COMITCon)*, Feb. 2019, pp. 35–39.
[32] J. Nalepa and M. Kawulok, "Selecting training sets for support vector machines: A review," *Artif. Intell. Rev.*, vol. 52, no. 2, pp. 857–900, Aug. 2019.
[33] Glória, André and Sebastião, Pedro. (2021). *P2P IoT Communications Energy and Link QUality Dataset*. Accessed: Dec. 3, 2021. [Online]. Available: https://www.kaggle.com/andregloria/p2p-iot-communication-energy-and-link-quality
[34] (2021). *Scikit-Learn*. Accessed: Feb. 16, 2021. [Online]. Available: https://scikit-learn.org/stable/
[35] (2021). *RandomizedSearchCV*. Accessed: Feb. 16, 2021. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.RandomizedSearchCV.html
[36] S. Salerno. (2021). *RandomizedSearchCV*. Accessed: Mar. 1, 2021. [Online]. Available: https://github.com/agrimagsrl/micromlgen
[37] D. F. S. Fernandes, A. Raimundo, F. Cercas, P. J. A. Sebastiao, R. Dinis, and L. S. Ferreira, "Comparison of artificial intelligence and semi-empirical methodologies for estimation of coverage in mobile networks," *IEEE Access*, vol. 8, pp. 139803–139812, 2020.
[38] A. Gloria, F. Cercas, and N. Souto, "Comparison of communication protocols for low cost Internet of Things devices," in *Proc. South Eastern Eur. Design Autom., Comput. Eng., Comput. Netw. Social Media Conf. (SEEDA-CECNSM)*, Sep. 2017, pp. 1–6.

ANDRÉ F. X. GLÓRIA (Member, IEEE) received the B.S. and M.S. degrees in telecommunication and computer engineering from the ISCTE–Instituto Universitário de Lisboa, in 2015 and 2017, respectively, where he is currently pursuing the Ph.D. degree in information science and technology, with a specialty in the Internet of Things and machine learning. From 2016 to 2017, he was a Teaching Assistant of the Artificial Intelligence class with the Information, Science and Technology Department, ISCTE–IUL. Since 2018, he has been a Researcher with ISCTE–IUL and the Instituto de Telecomunicações–IUL, with research on the Internet of Things, machine learning, wireless sensor networks, wireless communications, sustainability, and green tech.

PEDRO J. A. SEBASTIÃO (Member, IEEE) received the Ph.D. degree in electrical and computer engineering from IST. He is currently a Professor with the ISCTE–IUL's Information Science and Technology Department. He is also the Board Director of the AUDAX-ISCTE-Entrepreneurship and Innovation Center, ISCTE, responsible for the LABS LISBOA Incubator and a Researcher with the Institute of Telecommunications. He has oriented several master's dissertations and Ph.D. theses. He is the author or coauthor of more than 200 scientific articles and he has been responsible for several national and international research and development projects. He has been an expert and an evaluator of more than one hundred national and international civil and defense research and development projects. It has several scientific, engineering, and pedagogical awards. Also, he has organized or co-organized more than 55 national and international scientific conferences. He planned and developed several postgraduate courses in technologies and management, entrepreneurship and innovation, and transfer of technology and innovation. He has supported several projects involving technology transfer and creation of start-ups and spin offs of value to society and market. He has developed his professional activity in the national defense industries, initially in the Office of Studies and later as the Board Director of the Quality Department of the production of new products and technologies. He was also responsible for systems of communications technology in the Nokia-Siemens business area. His research interests include monitoring, control and communications of drones, unnamed vehicles, planning tools, stochastic process (modeling and efficient simulations), the Internet of Things, and efficient communication systems.