Intelligent Configuration of PHY-Layer Parameters to Reduce Energy Consumption in LoRa

Mário Nascimento Carvalho Filho* and Miguel Elias M. Campista*
* GTA/PEE-COPPE/DEL-Poli, Universidade Federal do Rio de Janeiro – Rio de Janeiro, Brazil
E-mails: {carvalho,miguel}@gta.ufrj.br

***Abstract*—Communications over long distances and strong resilience to interference are vital aspects of LoRa. LoRa adjusts the modulation to allow higher data transmission rates, depending on the reception sensitivity threshold and the communication distance. The spreading factor and the transmission power, in turn, are directly related to energy consumption, influencing network performance. This paper proposes the use of supervised learning techniques to configure the spreading factor and the transmission power simultaneously. This approach differs from the literature as it configures two parameters instead of just one, the spreading factor. Different learning techniques are evaluated through simulations using a LoRa network. Our experiments compare the performance of our proposal with the traditional LoRaWAN and the state-of-the-art on intelligent configuration using only the spreading factor. The obtained results show that our proposal successfully reduces the energy consumption without affecting the packet delivery ratio.**

I. Introduction

Technologies behind the Internet of Things (IoT), from microelectronics to cloud computing, have become the basis for the development of numerous applications and services. As a result, the number of IoT devices has grown in recent years, requiring a strong research effort to meet society's expectations. In this context, there are applications that generate a relatively small amount of data despite the large number of connected devices. These scenarios favor network technologies that provide large-scale connectivity and low energy consumption, as is the case of LPWAN (Low Power Wide Area Networks) [1], [2].

Large amounts of energy-constrained devices can be connected over long distances using LPWANs, which is not the case for other networking technologies, such as WiFi or cellular networks. WiFi networks typically provide low-power consumption with limited transmission range, whereas cellular networks allow larger coverage areas at the cost of more energy consumption. This last characteristic is a particular issue for applications running atop battery-powered devices as recharging or replacement is not always viable [1]. The tradeoff between energy consumption and transmission range benefits LPWANs as they cover large areas and save energy at the cost of low transmission rates. Among LPWANs, the Long Range (LoRa) is a highlight.

LoRa specification includes physical (LoRa RF) and link (LoRaWAN) layers. The low energy consumption and the long-range communication are characteristics achieved by LoRa using the Chirp Spread Spectrum (CSS) modulation at the physical layer and the pure ALOHA medium access at LoRaWAN. These technologies, however, have to deal with an increasing number of packet collisions and transmissions consuming more Time-on-Air (ToA). In a nutshell, the increase on ToA is related to selecting unsuitable transmission power and spreading factors by wireless devices. While high transmission power decreases spatial reuse opportunities, high spreading factors extend the ToA of the transmitted signals. Therefore, to minimize the probability of collisions and improve medium access, one must seek an efficient configuration of both spreading factor and transmission power.

This work proposes the use of supervised learning techniques to select the appropriate values for transmission power and spreading factor of wireless devices in dense LoRa networks, i.e., containing up to 1,000 end devices[1]. The current literature either relies on the standard LoRa mechanism, called Adaptive Data Rate (ADR) [3] or on approaches that use machine learning to configure the spreading factor alone [4]. There are also works addressing the same problem differently, using reinforcement learning, which has shown to be more expensive in terms of computation [5], [6]. The idea of our paper is then to combine both parameters to successfully transmit packets without wasting energy. To accomplish that, we assume that the network is static enough to yield the reuse of the same machine learning model in short-medium term. We also implement machine learning models to configure the spreading factor alone for comparative analysis. We run simulations using 125 and 500 kHz bandwidth to evaluate the proposal under different physical resource conditions. Additional bandwidth leads to more gateway sensitivity and a higher data rate for all spreading factor values.

The simulation results show that by configuring only the spreading factor with machine learning techniques, there is an increase in the packet delivery ratio at the cost of higher energy consumption compared with ADR. In our proposal, which configures both the transmission power and the spreading factor, we generally achieve a lower energy consumption compared with the single-parameter approach. Our proposal increases the packet delivery ratio with significantly lower energy consumption than LoRa's ADR. This last result is particularly relevant since it differentiates our proposal and confirms that the use of supervised learning techniques can contribute to LoRa networks. Hence, we summarize our main

[1]This paper has a preliminary version in Portuguese published at a Brazilian national conference (http://www.gta.ufrj.br/ftp/gta/TechReports/CaCa22.pdf).

contributions as follows:

- We use supervised learning techniques to simultaneously configure the spreading factor and the transmission power of LoRa networks.
- We run experiments to show that the proposed approach does not equally suffer from scalability issues, even with different bandwidths, at the same time it reduces the energy consumption compared with LoRa's ADR.

This paper is organized as follows: Section II provides LoRa background to understand our proposal. Section III introduces the state of the art in LoRa's PHY-layer parameters configuration. Section IV presents the machine learning models evaluated in our proposal such as Section V presents the hiper parameters models evaluated. Section VI details the simulation setup. Section VII defines four different configurations for bandwidth and LoRa parameters and then presents the obtained results. Finally, Section VIII concludes this work and draws future directions.

II. LoRa Background

LoRa uses a proprietary modulation derived from the Chirp Spread Spectrum (CSS). In LoRa, the chirp represents a sinusoidal signal with frequency linearly increasing over time. The signal covers the entire bandwidth, allowing long-range wireless communication links and low-power consumption [7].

LoRa networks typically follow a star topology with the gateway acting as the intermediate node between wireless and wired devices. Wireless devices are energy-constrained and communicate only with the gateway. In the link layer, LoRaWAN defines three classes of devices, Class A, B, and C, which provide different services for different applications. Class A devices perform the transmission (uplink) to the gateway at any time and have only one reception window (downlink) after the end of the transmission. Class B devices have multiple reception windows, while Class C devices can perform reception at any time.

The communications between devices and the gateway require the definition of a spreading factor, a transmission power, and one of the available subchannels. The spreading factor defines the spectral spread as well as the ratio between the bit rate and the chirp rate [8]. Six different values exist for the spreading factor, ranging from 7 to 12 [7]. On the one hand, the greater the spreading factor, the more resilient to noise the signal is and the longer the transmission range becomes. On the other hand, the greater the spreading factor, the longer the transmission Time on Air (ToA), which increases the energy consumption. Thus, there is a tradeoff between transmission range and energy consumption.

In addition to the impact on the energy consumed, increasing the spreading factor also reduces the data transmission rate (R_b) of LoRa. The rate R_b (in bps) is calculated as $R_b = SF \times TC \times \frac{2^{SF}}{BW}$, in which SF, BW, and TC denote, respectively, the spreading factor, the bandwidth in Hz, and the number of redundant bits in LoRa messages. Note that the data rate decreases with an increasing SF, considering TC and BW fixed. The bandwidth (BW) can be equal to 125, 250, or 500 kHz. The signal reception sensitivity at the gateway and the transmission rate for each spreading factor depend on the value assigned for the bandwidth. The number of redundant bits (TC) in LoRa messages for error recovery is a function of the coding rate (CR), where $TC = \frac{4}{4+CR}$. CR takes an integer value between 1 and 4 [7].

Likewise other wireless technologies, packet collisions in LoRa networks can occur when two or more transmissions overlap on the receiving device. In LoRa networks, collisions, or even poor spatial reuse, possibly result from the simultaneous reuse of the same spreading factor by different devices [9]. Even configuring different spreading factors, a collision may occur in the same channel due to the spreading factor imperfect orthogonality [4]. If concurrent transmissions use the same network resources (same spreading factor and channel), the gateway can only receive one of the transmissions. This reception is conditioned to a Signal-to-Interference-plus-Noise Ratio (SINR) above a given reception threshold for any spreading factor.

In LoRaWAN, devices do not initially know their distance to the closest gateway. They, however, can calculate this distance through the signal strength received from the downlink transmission. If the received signal strength is too high, the sensor may decrease the spreading factor to save energy. If the spreading factor is at the lowest level (level 7) and the received signal power is still high, the device can reduce the transmission power. This mechanism is called Adaptive Data Rate (ADR) in LoRaWAN. In this context, devices close to the gateway tend to configure their spreading factor to the minimum value, following the ADR mechanism, which leads to collisions between devices using the same spreading factor. If the number of devices increases, the number of collisions also increases. Note that, to provide spatial reuse, decreasing the spreading factor is not enough to reduce the transmission range and ToA. As a consequence, the transmission power must also be adjusted.

III. Related Work

Works tackling improvements over LoRa configuration approaches follow three main directions. The first one uses different techniques to compute optimal values, for example, optimization algorithms based on the number of connected devices. The second approach considers the use of supervised learning to configure a single LoRa parameter. Finally, the third approach addresses configuration improvements using reinforcement learning.

Cuomo et al. [3] proposed two new algorithms to select the spreading factor of LoRaWAN networks. One of the algorithms selects spreading factors based on the total number of connected devices. At the same time, the second algorithm performs the selection based on the packet ToA for each spreading factor. This proposal is restricted to the spreading factor. Yatagan and Orkut [4], on the other hand, explore supervised learning techniques to select the spreading factor that would provide the best packet delivery ratio in LoRa networks. The models used are Decision Tree and Support

Vector Machine. The authors only evaluate the effect of imperfect orthogonality because of the spreading factor, since this is implemented in the simulator developed and used by Yatagan and Orkut. All other LoRa's physical layer parameters are maintained.

Ta et al. [5] worked with reinforcement learning when proposing the approach LoRa-MAB. LoRa-MAB adjusts the spreading factor, the power transmission, and the transmission channel of LoRaWAN network in a distributed fashion (technique implemented at the devices) using the Exponential Algorithm Weights for Exploration and Exploitation (EXP3). Despite the increase in packet delivery ratio and the reduction in energy consumption, the computational complexity of the proposal is very high. For instance, simulations consider only 100 devices, not allowing evaluations in denser networks. Park et al. [6] proposed the use of deep reinforcement learning to adjust the spreading factor and the transmission power. The simulation area considered has only 1.5 km^2, used to position only 30 devices. Similarly, these simulation parameters do not allow performance evaluation in denser networks, as a consequence of high computational complexity.

IV. Intelligent Configuration of LoRa Parameters

The use of supervised learning techniques to configure the spreading factor and the transmission power is our main proposal to improve the medium access in LoRa networks. The idea behind is illustrated in Figure 1. In this figure, colored circles represent the transmission ranges of the wireless devices, while the triangles represent the gateways in a LoRa network. The different colors for the circles denote different transmission ranges. Note in Figure 1(a) that part of the devices has a transmission range too short or too long to communicate with the gateway. This range is a function of the spreading factor and the transmission power. However, even using LoRa's ADR mechanism, such configuration may not be the most efficient, given the lack of coordination. This paper proposes to adjust the LoRa parameters so that the range becomes sufficient to successfully communicate with the gateway, as depicted in Figure 1(b). The goal is to save energy and, at the same time, reduce the number of collisions in the network by promoting spatial reuse. Although the literature brings approaches involving machine learning, none of them simultaneously configures the spreading factor and the transmission power in dense networks.

In practice, we rely on a central server to configure the PHY-layer parameters of all participating devices. In the beginning, the server sets random values for the parameters (obviously using valid random values) and initializes a routine to receive data from devices during a time frame defined by the network administrator. This initial configuration with random values is interesting as it permits a prior understanding of the communication performance between devices and the gateway under different circumstances. Data regarding the transmission success of various devices configured with all possible PHY-layer parameter combinations is collected during this initial

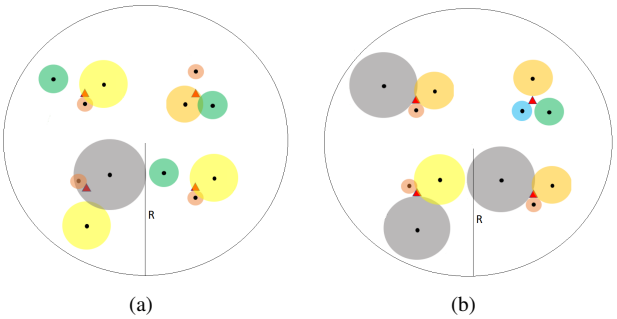

Fig. 1. Effect of PHY-layer parameter adjustments on the communication range of LoRa devices: (a) without parameter adjustment, and (b) with parameter adjustment. In both figures, R is the maximum network range.

step. Other approaches could also be used, such as LoRa's ADR mechanism. Using the collected dataset, the proposal performs offline training and testing of classifiers. After finishing the training and testing step, the central server configures all devices with the lowest values for each parameter that result in successful transmissions.

The dataset is composed of up to four attributes. If the transmission power is not considered, the server requires samples from wireless devices containing their coordinates (x, y) in the network topology and the spreading factor used.

This paper adds the transmission power to the collected samples since we also configure this parameter. During the data acquisition step for training and testing, the implementation of a client-server application in which the server can send control messages to the wireless devices should be foreseen. These messages could be used to adjust LoRa parameters and monitor the success of message transmissions.

V. Selected hyperparameters

The Decision Tree, the Support Vector Machine, and the Random Forest classifiers were used and shown to be adequate for obtaining a packet delivery ratio similar to that obtained by the ADR mechanism with reduced energy consumption for transmissions.

We used the grid search mechanism (GridSearch) of scikit-learn library [10] to carry out an exhaustive search for the best hyperparameters for the Decision Tree, the Random Forest, and the Support Vector Machine models during training. The hyperparameters evaluated for each classifier are described below:

- **Decision Tree**: the best impurity criterion between "entropy" and "gini";
- **Random Forest**: the best number of estimators between 3, 10, 60, 100, 200, and 400, and the best criterion of impurity between "entropy" and "gini";
- **Support Vector Machine**: the best regularization parameter between 1, 1.25, and 1.5 and the best kernel between "linear", "poly", and "rbf".

VI. Simulation Setup

We use the simulator developed in Python by Yatagan and Oktug [4], considering a circular topology with radius R and

composed of four gateways. These gateways are located in positions $(\frac{R}{1+\sqrt{2}}, \frac{R}{1+\sqrt{2}})$, $(\frac{R}{1+\sqrt{2}}, -\frac{R}{1+\sqrt{2}})$, $(-\frac{R}{1+\sqrt{2}}, \frac{R}{1+\sqrt{2}})$, and $(-\frac{R}{1+\sqrt{2}}, -\frac{R}{1+\sqrt{2}})$. The simulator implements spreading factor imperfect orthogonality and collisions due to concurrent transmissions. This makes the analysis as close as possible to reality. We adapted the simulator to additionally allow the configuration of the transmission power and bandwidth, originally set to 14 dBm and 125 kHz, respectively [4]. The propagation model used is the free space, and the channel used is at 868 MHz. The sum of the signal gains and losses considered in the transmitting and receiving devices is 7 dB. The CR correction rate code used has value 1.

The number of wireless devices varies from 100 to 1,000, which allowed the performance evaluation of supervised machine learning techniques in denser networks. The packet size sent by devices is 100 Bytes. The devices are Class A, as they have low energy consumption. The interval between consecutive transmissions of LoRa packets follows a Poisson distribution with average $0,01 \cdot t$ [11], [12]. Data reception by wireless devices (downlink) is not considered.

We conduct an initial simulation during 3,600 seconds. In this simulation, wireless devices are individually configured with random parameters, as proposed in Section IV. A dataset is obtained to log the transmission performance of all considered wireless devices. The dataset generated, although unbalanced, is enough to provide satisfactory results during machine learning model training. The number of samples from the "unsuccessful transmission" class is minimal compared with the number from the "transmission successful" class.

After the above simulation, the generated dataset is used in the offline training of the classifiers using the scikit-learn library [10]. After training and testing each classifier, we picked the parameters with the lowest values providing successful transmission for each device. Then, we started a new simulation to obtain the packet delivery ratio and energy consumption. We run a parallel simulation with LoRa's ADR for performance comparison regarding packet delivery ratio and energy consumption with the proposed models.

VII. Results

The results compare the performance of four different configurations, in addition to the ADR mechanism defined in LoRaWAN. Among the four configurations, Configurations 3 and 4 refer to our proposal, as seen in Table I. Note that the spreading factor and the transmission power can assume different values, where the spreading factor can vary from 7 to 12, and the transmission power can vary between values 8, 11, and 14 dbm. We use these configurations to evaluate the performance of ADR and the supervised learning models under different available network resources.

A. Models precision

Figure 2 plots the precision obtained by each one of the four configurations (see Table I) for the different machine learning models. The precision is computed as the ratio between the number of true positives (successful transmissions) and the number of positive predictions (sum of true positives and false positives). Precision is an essential metric as false positives may have a considerable negative impact in our case. In LoRa networks, a false positive can lead to an incorrect configuration of PHY-layer parameters. All results plot vertical lines with 95% of confidence interval.

TABLE I
Configurations used in our simulations.

Configuration	Spreading Factor (SF)	Transmission Power (PW)	Bandwidth (BW)
1	Variable	14 dBm	125 kHz
2	Variable	14 dBm	500 kHz
3 (Proposed)	Variable	Variable	125 kHz
4 (Proposed)	Variable	Variable	500 kHz

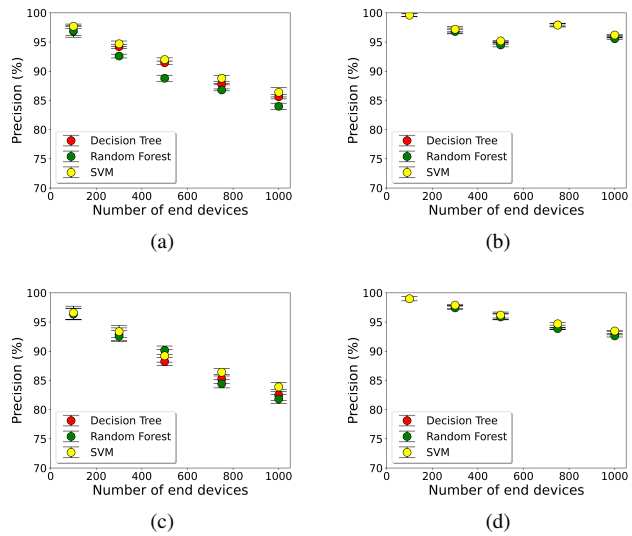

Fig. 2. Precision of the evaluated models: (a) Configuration 1, (b) Configuration 2, (c) Configuration 3, and (d) Configuration 4.

In Configuration 1, according to Figure 2(a), the Decision Tree and the Support Vector Machine provided the same precision for all evaluated number of devices, considering the confidence interval of 95%. However, the Random Forest achieved a lower precision than Decision Tree and Support Vector Machine from 300 to 1,000 devices. Similarly, for Configuration 2, according to Figure 2(b), all models obtained the same precision for the increasing number of wireless devices.

Figure 2(c) shows that, for 100 and 300 devices, all models have the same precision. On the one hand, for 500 devices, the Decision Tree has slightly lower precision than Random Forest. On the other hand, for 750 and 1,000 devices, the Random Forest has slightly lower precision than Support Vector Machine, even though this is very close to the results obtained with the Decision Tree. According to Figure 2(d), all models have approximately the same precision for the increasing number of wireless devices. Considering Configurations 1 and 3, for 125 kHz bandwidth, the results for precision are

very similar. The same is true if we compare Configuration 2 and 4 for 500 kHz bandwidth.

B. Packet delivery ratio

The packet delivery ratio computes the fraction of packets received over all transmitted by the gateways. Hence, Figure 3(a) shows that all supervised learning models have the same performance as ADR for 100 and 300 devices. For 500, 750, and 1,000 devices, all machine-learning-based models outperform ADR. The Support Machine Vector obtained the best performance, which is coherent with the precision result, as shown in Figure 2(a). The difference in performance between ADR and all machine-learning-based approaches increases with the number of wireless devices. This is a consequence of the spreading factor selected by each model, as observed in Figure 4(a) for 1,000 devices. Using the Support Vector Machine, wireless devices tend to more often select a higher spreading factor than the other devices running different models and ADR. This increases the Time on Air (ToA) of the packets, consequently improving the delivery ratio at the cost of higher energy consumption.

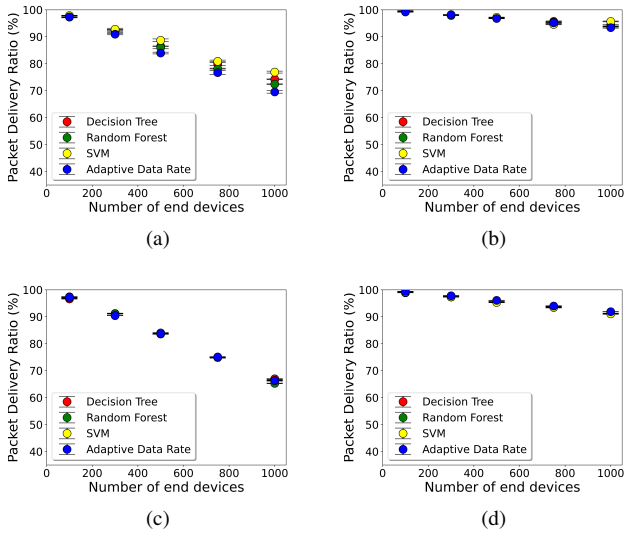

Fig. 3. Packet delivery ratio: (a) Configuration 1, (b) Configuration 2, (c) Configuration 3, and (d) Configuration 4.

Considering the bandwidth of 500 kHz (Configuration 2), shown in Figure 3(b), the performance of the models is similar to that of ADR. This means that the increase in network resources can reduce the difference between the performance of ADR and the machine-learning-based proposals. For 1,000 devices, devices using the Support Vector Machine select higher values of spreading factor than those selected by ADR and the other approaches, as shown in Figure 4(b). This also increases the Time on Air, consequently improving the packet delivery ratio.

In Configuration 3, which assigns both transmission power and spreading factor, Figure 3(c) shows that the performance for packet delivery ratio of all the proposed models is similar to ADR. The difference is a lower energy consumption than ADR, as shown in the next section. The same behavior of Configuration 3 can be observed in Configuration 4, as seen in Figure 3(d), where the machine-learning-based models achieve a performance similar to ADR. Compared with Configuration 3, the packet delivery ratio increases since the network bandwidth is 500 kHz and no longer 125 kHz.

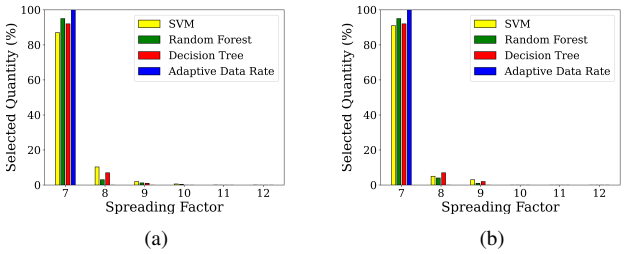

Fig. 4. Spreading factor selection for 1,000 wireless devices: (a) Configuration 1 and (b) Configuration 2.

C. Energy consumption

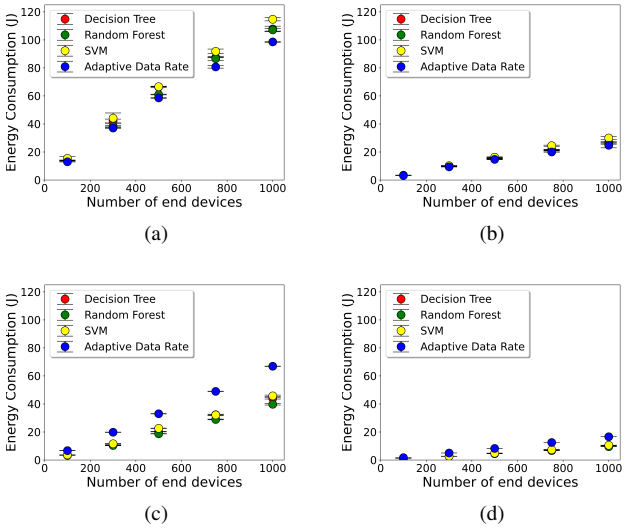

Fig. 5. Energy Consumption: (a) Configuration 1, (b) Configuration 2, (c) Configuration 3, and (d) Configuration 4.

As discussed in the last section, the Support Vector Machine obtained the best performance in terms of packet delivery ratio in Configuration 1, but with the highest energy consumption, as shown in Figure 5(a). The results of Figure 3(a) and 5(a) show that the energy consumption increases with the packet delivery ratio for Configuration 1. Considering the 500 kHz bandwidth (Configuration 2), as shown in Figure 5(b), the performance of the machine-learning-based models is similar to ADR in terms of energy consumption. Hence, increasing the network resources reduce the difference in energy consumption between ADR and the other supervised learning approaches.

Figure 5(c) shows that all supervised learning models obtain a better performance than ADR in terms of energy consumption in our proposed Configuration 3. Therefore, the results in

Figure 3(c) and 5(c) show that the proposal to use supervised learning techniques to assign both transmission power and spreading factor provides a reduction in the energy consumed compared with ADR. Moreover, the decrease in energy consumption occurs while keeping the same performance for the packet delivery ratio. Figure 6(a) and 6(b) show that, even though ADR selects smaller values for the spreading factor, it selects higher values for transmission power. This explains the higher energy consumption obtained compared with the supervised learning models in Configuration 3.

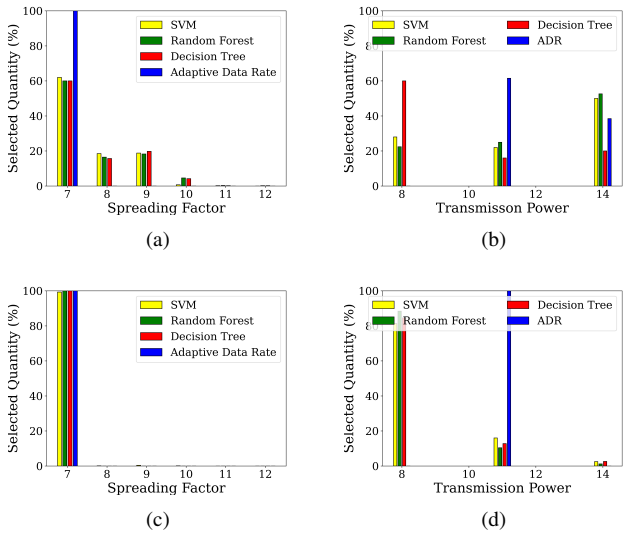

Fig. 6. Spreading factor and transmission power selection for 1,000 devices: (a) spreading factor selection for Configuration 3, (b) transmission power selection for Configuration 3, (c) spreading factor selection for Configuration 4, and (d) transmission power selection for Configuration 4.

The same behavior of Configuration 3 can be seen in Configuration 4, as shown in Figure 5(d). The proposed models obtained better performance than ADR in terms of energy consumption. However, the gap between the performance of the proposed models and ADR reduces compared with Configuration 3. This occurs because the amount of network resources, i.e., the bandwidth, increases from 125 kHz to 500 kHz. Figure 6(c) and 6(d) show that, even though ADR and the supervised learning techniques select spreading factors of value 7, ADR selects even higher values for transmission power. This is the reason behind the higher energy consumption of ADR compared with the supervised learning techniques.

VIII. Conclusion

This work evaluated the use of supervised learning techniques to configure two PHY-layer parameters of LoRa, the spreading factor and the transmission power. The performance of our proposal was evaluated in dense LoRa settings using four alternative approaches. We compare the performance of configuring both parameters using machine-learning-based models (Support Vector Machine, Random Forest, and Decision Tree) with the standard approach from the literature, which changes only the spreading factor. We also evaluate the impact of using different bandwidth values, i.e., 125 and 500 kHz.

Simulation results with both bandwidth values show that the proposed machine learning models obtained the same performance of LoRa's Adaptive Data Rate (ADR) in terms of packet delivery ratio. However, our proposal reduces the energy consumed for packet transmissions, mainly in scenarios with limited bandwidth resources.

As future work, we plan to analyze different learning techniques, such as neural networks and logistic regression. Yet, in addition to the spreading factor and the transmission power, we would like to evaluate the impact of an intelligent configuration of frequency as an additional parameter of the physical layer of LoRa.

Acknowledgement

This work is partially supported by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001. It was also supported by CNPq, FAPERJ Grants E-26/211.144/2019 and E-26/202.689/2018, and FAPESP Grant 15/24494-8.

References

[1] U. Raza, P. Kulkarni, and M. Sooriyabandara, "Low power wide area networks: an overview," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 2, p. 855–873, 2017.

[2] O. Georgiou and U. Raza, "Low power wide area network analysis: Can LoRa scale?" *IEEE Wireless Communications Letters*, vol. 6, no. 2, p. 162–165, 2017.

[3] F. Cuomo, M. Campo, A. Caponi, G. Bianchi, G. Rossini, and P. Pisani, "EXPLoRa: Extending the performance of lora by suitable spreading factor allocations," in *IEEE 13th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*. Rome, Italy: IEEE, 2017.

[4] T. Yatagan and S. Oktug, "Smart spreading factor assignment for LoRaWANs," in *IEEE Symposium on Computers and Communications (ISCC)*. Barcelona, Spain: IEEE, 2019.

[5] D.-T. Ta, K. Khawam, S. Lahoud, C. Adjih, and S. Martin, "LoRa-MAB: A flexible simulator for decentralized learning resource allocation in iot networks," in *12th IFIP Wireless and Mobile Networking Conference (WMNC)*. Paris, France: IEEE, 2019.

[6] G. Park, W. Lee, and I. Joe, "Network resource optimization with reinforcement learning for low power wide area networks," *EURASIP Journal onWireless Communications and Networking*, vol. 2020, no. 176, 2020.

[7] Semtech Corp., "LoRa modulation basics an1200.22," Available at: https://www.frugalprototype.com/wp-content/uploads/2016/08/an1200.22.pdf, 2015, accessed in January 2022.

[8] M. Bor, J. Vidler, and U. Roedig, "Lora for the internet of things," in *EWSN '16: Proceedings of the 2016 International Conference on Embedded Wireless Systems and Networks*. United States: Junction Publishing, 2016.

[9] M. C. Bor, U. Roedig, T. Voigt, and J. M. Alonso, "Do lora low-power wide-area networks scale?" in *MSWiM '16: Proceedings of the 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*. New York, NY, United States: Association for Computing Machinery, 2016.

[10] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[11] H. C. Tijms, *A first course in stochastic models*. New York, NY: Wiley, 2003.

[12] S. Ross, *Stochastic processes*. New York, NY: Wiley, 1996.