LoRaWAN optimization using optimized Auto-Regressive algorithm, Support Vector Machine and Temporal Fusion Transformer for QoS ensuring

HousemEddine Elbsir *†, Mohamed Kassab ‡, Sami Bhiri ‡§, Mohamed Hedi Bedoui †,
David Castells-Rufas ¶, Jordi Carrabina ¶
* Higher Institute of Computer Science and Communication Techniques, University of Sousse, 4011, H. Sousse, Tunisia
† Laboratory of Technologies and Medical Imaging, Faculty of Medicine of Monastir, University of Monastir, Tunisia
‡ Higher Institute of Informatics and Mathematics of Monastir, University of Monastir, Tunisia
§ OASIS, ENIT University, Tunis El Manar
¶ Department of Microelectronics and Electronic Systems, Universitat Autònoma de Barcelona, Bellaterra, Spain
{houssem.bsir.gt, mohamed.kassab, sami.bhiri, hedi.bedoui2015}@gmail.com, {david.castells, jordi.carrabina}@uab.cat

***Abstract*—The number of LoRaWAN networks have grown worldwide last years, offering a solution for the integration of the Internet of Things in rural and urban areas. After years of development, several performance issues and scalability limitations require to be enhanced for LoRa such as high collision rates and duty cycle limitations. Machine learning offers a chance for LoRaWAN to rise as the reference communication technology that offers the adequate communication performances for IoT. In this paper, our goal is to optimize the LoRaWAN network performances using detection mechanism and artificial intelligence to predict its behavior. first, we evaluate the full potential of the LoRaWAN factory setting, and we introduced a Quality of Service demanding application. Second, we constructed our proper database using available application criteria, we included a quality of service mechanism to simulate the effect of a new application connecting to a stable network and the perturbing causes.**

Then, we used two different methods one for classification, the second for prediction, and then the optimization. For classification using Auto-Regressive and optimization it using burg algorithm and firefly algorithm, then we used the support vector machine for traffic classification, results are very promising, we were able to detect normal traffic, a normal surge, and an abnormal surge of network traffic with up to 99% accuracy. For prediction, we used a new algorithm developed by google Temporal Fusion Transformer. We were able to predict the network behaviour ahead with 14 days with 95% accuracy and up to 30 days with 80% accuracy. We were able to optimize the network to absorb the abnormal surge and return to normal in less than 60% of the normal time, uplifting the packet delivery ratio for uplink traffic by 20% and downlink traffic by 50%.

***Index Terms*—LoRaWAN, channel occupancy, Machine Learning, prediction analysis, network optimization, smart city, cluster analysis**

I. Introduction

LoRaWAN technology [1], which relies on the LoRa physical layer, has become a strong contender to dominate the low power wide area network (LPWAN) arena required for the future Internet of Things (IoT). It has introduced a successful network model using a star topology based on wireless access gateways and a centralized network server. LoRaWAN offers three classes of MAC layer to respond to the majority of IoT applications: class A for uplink-focused applications, class B for uplink-focused applications with occasional downlink traffic, and class C for applications with continuous downlink traffic. LoRaWAN protocol, based on ALOHA communication MAC layer, suffer from a high packets loss related to the number of end devices and their generated traffic. Also LoRaWAN does not provide smart control over end devices in the network. In the past years, new technology has emerged to propose a long time waited for solutions for many problems in different life areas, from the stock market, and cancer detection to farming. Artificial intelligent (AI) have to potential to detect some behavior in the data-set and can predict the future results. AI algorithms have the potential to fix the LoRaWAN default settings problem.

In this paper, we propose a new approach to optimize the LoRaWAN network using artificial intelligence algorithms. We address two main issues:

the detection and the identification of the different patterns in the network (normal vs abnormal traffic), and the prediction of the network behaviour. Then, we propose an optimization for the network based on the obtained results.

This paper is organized as follows. In Section II, we present the related work. In Section III describes the artificial intelligence (AI) algorithms that will be used. In Section IV, we present our experiments, scenario, and data-set. Section 5 presents the performance evaluation results and section 6 details the results of the algorithms. In Section 7, we present the proposed optimization of LoRaWan, and in Section 8 we conclude the paper.

II. Related Work

The characteristics of network traffic regarding packet delivery ratio, latency, and power consumption for the three different classes of devices (A,B,C) has been studied with some extend. [2], [3] results have identified some major problems with the network default setting such as huge packet

978-1-6654-6975-3/22/$31.00 ©2022 IEEE

loss when there is a surge in traffic and there is no mechanism to fix that [4][5][6][7]. There is rapidly increasing interest in applying machine learning approaches to IoT systems because of its ability to model an underlying, usually unknown, closed-form process with a finite set of data. [8][9]. This novel technology has provided a useful toolkit for modeling systems of IoT with a given data set that mirrors the real scenario as closely as possible. [10] provides a description of machine learning techniques for IoT, including a discussion of how ML algorithms may be used for IoT smart content and what the real-world properties of IoT data are. Other publications suggest using machine learning for abnormal recognition or security concerns[11].

Some recent work [12] has used Deep Learning (DL) to detect Joint Collision and Spreading Factor Allocation but their work suffers from major flaws such as the use of small number of devices (between 100 and 1000) that is not sufficient to saturate the network and then to show the real effect of collisions on the performance.

III. ARTIFICIAL INTELLIGENCE ALGORITHMS

Many AI classification algorithms are available. After carefully examining the state of the art and relevant applications we choose to use the Auto-regressive (AR) for the classification and the Temporal Fusion Transformer (TFT) developed by google [13] for the prediction.

A. Autoregressive algorithm

A pattern is defined in mathematics as a repeated arrangement of numbers, shapes, colours and so on. The Pattern can be related to any type of event or object. If the set of numbers are related to each other in a specific rule, then the rule or manner is called a pattern. It is used for predictions when a certain correlation exists between time series values and the values that come before and after them[14]. AR model permits describing the network traffic signal as a linear representation. In this part we will implement three strategy stages: the autoregressive methodology, extraction of features employing Burg's method (AR-B), a mixed approach employing the firefly (FA) technique to acquire the right model order P, which is used to select the best AR model coefficients, and finally an SVM classification strategy.

Burg's method, firefly algorithm (AR-FA) refinement of the AR model order P, and support vector machines are all discussed in the following sections.

1) Burg algorithm: Since the Burg algorithm differs from other approaches that ensure the production of a stable model, it is used to determine the parameters of an AR model. Network traffic input X(n); n = 1,2,3,. . . ., N, in which N is the signal's time in milliseconds and X(n) is the magnitude in terms of network traffic such as packet sent/receive/lost. N can be the number of packet sent/received/lost. In order to reduce both forward and backward forecast error, the approach is a recursive method that relies on the lattice filter structure.

2) Algorithm AR-FA: Finding a suitable model order P, which has been one of the typical issues in AR approaches, is among the key goals of this work. To obtain the ideal model order P, the firefly Optimization technique has been used [14] . The AR-FA selects an optimal model order that chooses the best parameters.

3) Support vector machines (SVM): Support vector machine is a sophisticated classifier technique that has been utilized in a variety of applications, including the classification of face recognition [15]. The SVM algorithm is based on kernel functions, which are used to separate data into groups by solving a quadratic optimization model. Radial basis functions (RBF) were utilized as kernel functions of SVM classifiers in this study, and the results were satisfactory. The RBF is expressed by:

$$K(x, y) = \exp^{\left(-\frac{|x-1|^2}{2\sigma^2}\right)} \tag{1}$$

4) Features extraction of network traffic using AR model: The feature extraction was determined using the FA method's optimum model order P, which allows for the identification of the best AR coefficients with the least amount of residual variance. The SVM classifier relied on these coefficients. For each set, the AR-FA was applied independently.

B. Temporal Fusion Transformer

A revolutionary attention-based framework that offers superior multi-horizon predicting with temporal dynamics observations that can be understood. TFT is meant to use canonical elements to quickly construct feature representations for every input type, resulting in strong prediction performance in a range of situations. TFT is composed of the following fundamental elements:

- Adaptive depth and network complexity to suit a broad range of datasets and situations, as well as gating methods to pass through any unneeded components of the design.
- To pick relevant input variables at each time, variable selection networks are used.
- Stationary covariate encoders use context vectors to condition temporal dynamics and integrate static information into the network.
- Temporal computing can be used to learn long- and short-term temporal correlations from visible and known time-varying data. A sequence-to-sequence layer handles local processing, while an interpretable multi-head attention component handles long-term dependencies.

IV. EXPERIMENTS

For all simulation we will be using the NS3 simulator with the LoRaWAN module. First, we fixed the scenarios, then the quality of service algorithm and we created the data set for the algorithms.

A. scenario

To obtain a realistic setup, we consider the traffic models described in the related work[2], [3] [4][5][6] and other real applications. The first application (App 1) mimics the functioning of a public lighting application in a smart city with 100000 sensors sending data when turning on (sunset) and off (sunrise) with 25% sending confirmed traffic each time to confirm the lamp conditions. The second application (App 2) mimics the functioning of a sensor such as a smart switch, turning on/off every 2 hours and reporting its status each time. The third application (App 3) mimics the functioning of more active sensors like the one in a fridge or air conditioner. The fourth application (App 4) mimics the functioning of an actuator application with a lot of traffic but a limited number. The total number of devices is also fixed to 100000, which can be supported by a GW in a realistic scenario. Table I summarizes the traffic setting.

application	% of end devices	uplink traffic	Unconfirmed /confirmed
01	40%	12h	75 % unconf 25% conf
02	40%	2 hours	75% unconf 25% conf
03	15%	1 hour	75% unconf 25% conf
04	5%	30 min	75% unconf 25% conf

TABLE I: Traffic setting

B. Quality of Serves (QoS)

In this paper, we implement a QoS mechanism allowing specific devices to request a preferred quality of service from the network server. We consider this possibility for devices running App 4 (5000 devices as shown in Table I). The QoS Parameter are the throughput, availability and the time response. We assign a weight for each parameter and we give it a preferred value, depending on the network status we allowed, the network server to negotiate it. We fixed the tolerance at 10% and with random steps for acceptance between 1% and 3%.

To implement the QoS mechanism, we integrate a manager on the devices and on the network server. A strategy manager will calculate the permitted QoS depending on the network status and monitor the QoS to maintain the fixed QoS after achieving it.

C. Data-Set

Creating the data set is challenging. There is no public data set available for testing. As result, We built our own data set. For that, we used NS3 simulator with LoRaWAN module. We run the simulation 10 times changing the random seed of the simulator with results each time with different placement of the end devices in the grid. End devices are placed randomly in 8 km circle with a centred gateway. We considered all possible criteria as features for end-devices, the inter packets time, the SNR, the duty cycle, number of transitions, SF, frequency user, ADR, successful transmission and failed transmission. We run each simulation for 365 days. Thus, we end up with a 100GB database. This database is available upon request.

V. Performance Evaluation

In this part, we are going to measure the performance of the real scenario with three and eight channels enabled and disabled and determine the packet loss causes.

A. 3 channel vs 8 channels

First, we start with the stock setting, we run the simulations for 48h to determine the uplift of the 8 vs 3 channels. Results are presented in Table II.

Traffic	3 channels	8 channels	
Pdr Uplink	55%	86%	
Pdr downlink	38%	92%	

TABLE II: Performance evaluation 3 vs 8 channels

Then, we check the causes of the packet loss in the 8 channel configurations, duty cycle is the major cause for packet loss with 89.5% as shown in TableIII.

Packet loss causes	
Duty cycle gateway	89.5% with 85% on RX2 channel
Internal collision	10.5%

TABLE III: Packet loss causes in 8 channels

B. 24h and 1 week

In this section, we show the evolution of sent, received and lost packets in 24 hours and one week in Figures 1 and 2. As we can see, there is an apparent pattern that we want to detect and predict.

C. QoS effect

After introducing the QoS mechanism for devices running App4, we see from Figure 3 with the factory setting, the abnormal surge and disruption it causes to the network. It took the network up to seven hours to regain its stable position. Data losses are between 35% and 99% in that period. With is more than ten times the normal amount. We tried to add two channels that can be used for QoS negotiation, in figure4 , we can see that the network took 6 hours to regain its stable position and data losses were between 30% and 80%. For this, detection of this surge and optimization of the network is needed. In 24h, when introducing a QoS requesting application we can see a significant performance loss in table IV. From 92 to 44 performance loss in downlink and 86 to 62 in the uplink.

Traffic	Without QoS	With QoS
Pdr Uplink	86%	62%
Pdr downlink	92%	44%

TABLE IV: QoS effect on network

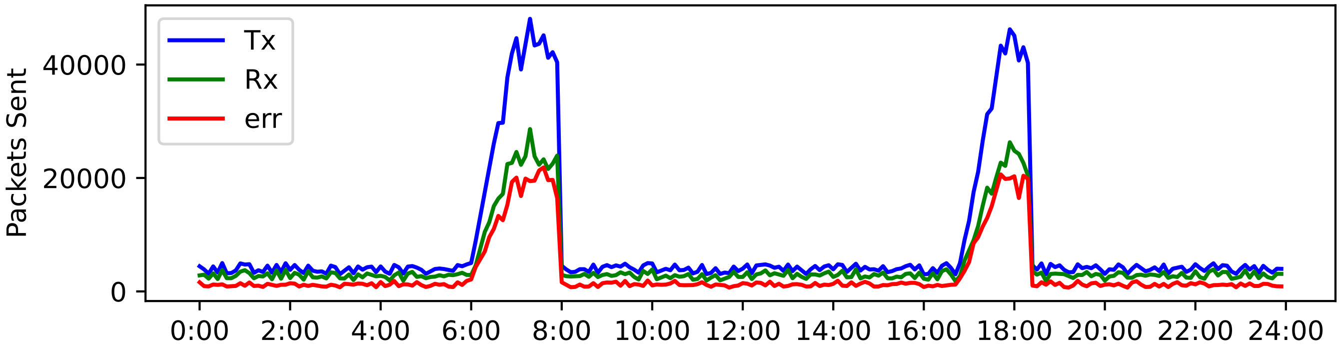

Fig. 1: Network traffic for 24h

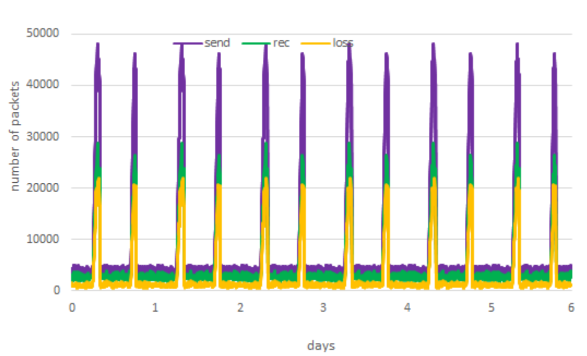

Fig. 2: Network traffic for one week

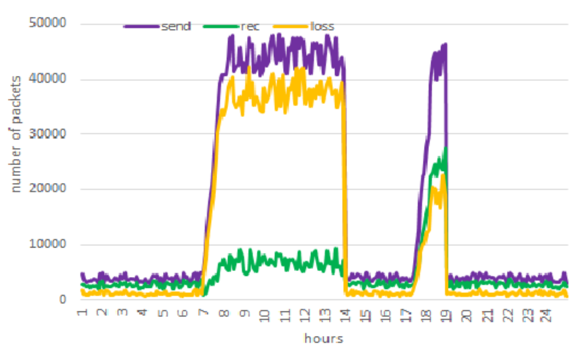

Fig. 3: Qos response on RX1 and RX2 channel

VI. ML MODELS PERFORMANCE

A. autoregressive

The quality was assessed by Accuracy (ACC), sensitivity (SEN) and specificity (SPE) (SPE). TP is a genuine positive. TN is genuine negative. FP is a false positive. FN is false-negative findings. The ACC, SEN, and SPE were calculated as follows:

$$ACC = (TP + TN)/N \tag{2}$$

$$SEN = TP/(TP + FN) \tag{3}$$

$$SPE = TN/(TN + FP) \tag{4}$$

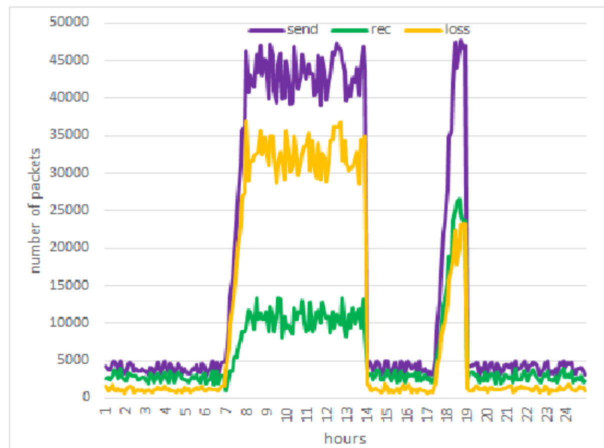

Fig. 4: QoS response on RX1 and 2 other channels,s not RX2

traffic	Normal	with surge	with abnormal surge
ACC	99.99	98.95	96.4
SEN	99.99	98.99	98
SPE	99.99	98.99	98

TABLE V: Auto-Regressive algorithm detection results

Results are shown in table V, we can conclude that the optimised AR algorithm has successfully detected the different network behaviour, normal, surge and abnormal surge due to QoS requesting.

B. Temporal Fusion Transformer

For the TFT algorithm, we used different algorithm settings. State size from 10 to 320 with a step of 10 , dropout rate from 0.1 to 0.9 with a step of 0.1 , minibatch size 64, 128 and 256 , learning rate from 0.0001 to 0.01 with a coefficient of 10 and the heads number equals to 1 or 4.

The best results are obtained with the following parameters and are used in the rest of the paper. For the data-set specification the output is R (time en s) and number of packets, number of end-devices equals 100000, and time period equals 365 days. For the algorithm parameters: k equals 168, tau max equals 24, dropout rate 0.3, state size 320 and the number of heads equal 4. For the training parameters: minibatch size equal 128, learning rate equal 0.001 and max gradient norm equal 100.

We partition the dataset it into 3 parts : a training set =60%, a validation set for hyperparameter tuning=20%, and a test set for performance evaluation=20%. Hyperparameter tuning is accomplished by random search, employing 240 cycles for Volatility, and 60 cycles for others.

Results from figure 5show that the TFT algorithm follows the path of the real traffic with different levels of accuracy over time. Normal traffic accuracy is up to 96% and it is down to 90% when a surge in traffic happens. This is due to the difference in traffic surge each time, as we are using a random inter-packet time.

The same is for the received packets and lost packets.

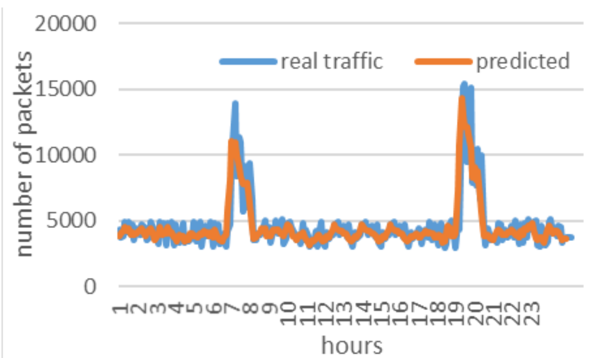

Fig. 5: traffic Send prediction in 24h

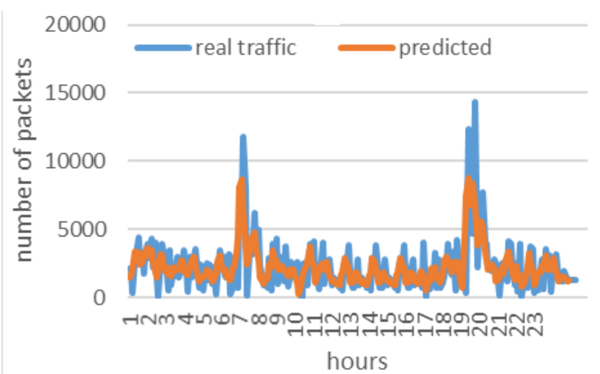

Fig. 6: Traffic received prediction in 24h

Results from figure 5,6,7 , we can conclude that the TFT algorithm and using the AR-FA and the SVM as inputs has successfully predicted the network behaviour with accuracy ranging from 85% to 96% in sent traffic , received traffic and lost traffic. The main thing to mention is that the TFT algorithm has successfully and with up to 99% got the pattern correct, although it did not get the value of the surge correct with an accuracy of 60% to 85%.

We were able to predict the network behaviour ahead with 14 days with 95% accuracy and up to 30 days with 80% accuracy from figure 8.

We can observe that the TFT appeared to modify its behaviour between regimes by devoting equal attention to historical inputs while there was low volatility, but it grew more interested in fast trend shifts during high volatility periods, reflecting temporal differences between the two regimes. This shows changes in the temporal dynamics learnt in each of these scenarios.

Fig. 7: Traffic loss prediction in 24h

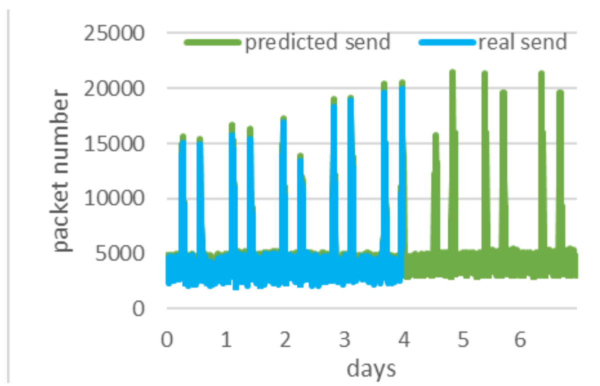

Fig. 8: Traffic prediction for 1 week

VII. OPTIMIZATION

Using the AR and TFT algorithms, we constructed an optimization mechanism, the TFT serves to predict the behaviour of the network for days to come, and the AR serves to detect the surge in traffic.

Using the predicted model, the network server conserves a necessary percentage of the duty cycle for upcoming traffic. For example, if the network predicts that it will have to send 2000 packets in the next hour, it will take into consideration limiting ACK for low urgent applications.

When the network server detects a surge in traffic due to QoS request, it stops sending an ACK to other less important applications. Figure 9 shows that the network is able to absorb the abnormal surge and maintain normal functioning. Besides, when the network detects saturation of the duty cycle, it limits the ACK and downlink during the surge period to maintain a stable status. Table VI shows the impact of the normal surge and abnormal surge prediction and detection and changing the network server configuration to maintain a stable network.

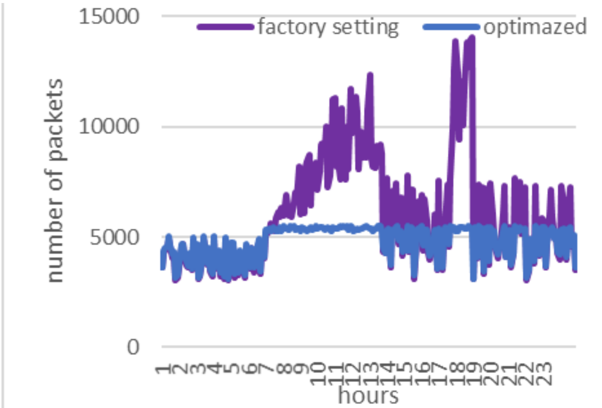

Fig. 9: Factory setting vs optimized

	factory setting	optimized
PDR up	62	89
PDR down	44	96
packet loss duty cycle	94	92
average re-transmission	4.8	2.9

TABLE VI: network performance before and after optimization

VIII. Conclusion

In this paper, we proposed two artificial intelligent methods to optimize the LoRaWAN network performance. First, we studied the full potential of the network and introduce an application with QoS requirements. Second, we used a combining autoregressive model optimized by Firefly algorithm and SVM to classify and detect normal traffic, a surge in traffic and an abnormal surge due to QoS request. The suggested technique was able to get a fair overall of classifications precision between 94.50 and 99%. Third, we present an attention-based model for high-performance multi-horizon forecasting. We used the TFT to predict the network's behavior successfully for 14 days with 95% accuracy and up to 30 days with 80% accuracy. Finally, we optimized the LoRaWAN network to absorb the network surge in traffic after detecting it and to use the prediction model to actively monitor the duty cycle to maintain a stable network.

In future work, we will implement the new LoRaWAN 2.4Ghz on NS3, evaluate it, figure out the problems and implement artificial intelligence to optimize it.

References

[1] *What is lorawan® specification*. Oct. 2021. URL: https://lora-alliance.org/about-lorawan/.

[2] Houssem Eddin Elbsir et al. "Evaluation of LoRaWAN Class B efficiency for downlink traffic". In: *2020 16th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*. 2020, pp. 105–110.

[3] Dara Ron et al. "Performance analysis and optimization of downlink transmission in LoRaWAN class B mode". In: *IEEE Internet of Things Journal* 7.8 (2020), pp. 7836–7847.

[4] Helin Yang et al. "Deep Reinforcement Learning Based Resource Allocation for Heterogeneous Networks". In: *2021 17th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*. 2021, pp. 253–258.

[5] Allan Espindola et al. "A Deep Autoencoder and RNN Model for Indoor Localization with Variable Propagation Loss". In: *2021 17th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*. 2021, pp. 391–396.

[6] Blaz Bertalanic et al. "A Deep Learning Model for Anomalous Wireless Link Detection". In: *2021 17th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*. 2021, pp. 265–270.

[7] David Castells-Rufas, Adrià Galin-Pons, and Jordi Carrabina. "The Regulation of Unlicensed Sub-GHz bands: Are Stronger Restrictions Required for LPWAN-based IoT Success?" In: ().

[8] Junfei Qiu et al. "A survey of machine learning for big data processing". In: *EURASIP Journal on Advances in Signal Processing* (May 2016).

[9] Alessio Martino, Alessandro Giuliani, and Antonello Rizzi. "Granular computing techniques for bioinformatics pattern recognition problems in non-metric spaces". In: Jan. 2018.

[10] Mohammad Saeid Mahdavinejad et al. "Machine learning for internet of things data analysis: a survey". In: *Digital Communications and Networks* (2018), pp. 161–175. ISSN: 2352-8648.

[11] Giuseppe Aceto et al. "Mobile Encrypted Traffic Classification Using Deep Learning: Experimental Evaluation, Lessons Learned, and Challenges". In: *IEEE Transactions on Network and Service Management* 16 (2019), pp. 445–458.

[12] Wali Ullah Khan et al. "Deep Learning Based Joint Collision Detection and Spreading Factor Allocation in LoRaWAN". In: (2022).

[13] Bryan Lim et al. "Temporal fusion transformers for interpretable multi-horizon time series forecasting". In: *International Journal of Forecasting* 37.4 (2021), pp. 1748–1764.

[14] Abdelouahab Attia, Abdelouahab Moussaoui, and Youssef Chahir. "Epileptic seizures identification with autoregressive model and firefly optimization based classification". In: *Evolving Systems* (Sept. 2021).

[15] Slim Ben Chaabane et al. "Face recognition based on statistical features and SVM classifier". In: *Multimedia Tools and Applications* 81.6 (2022), pp. 8767–8784.