2022 IEEE 42nd International Conference on Distributed Computing Systems Workshops (ICDCSW) | 978-1-6654-8879-2/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICDCSW56584.2022.00043

Deep Learning Based Joint Collision Detection and Spreading Factor Allocation in LoRaWAN

Seham Ibrahem Abd Elkarim*, M.M. Elsherbini*†, Ola Mohammed†, Wali Ullah Khan‡,
Omer Waqar§, Basem M. ElHalawany*¶

*Department of Electrical Engineering, Faculty of Engineering at Shoubra, Benha University, Egypt
†Department of Electrical Engineering, Egyptian Academy of Engineering and Advanced Technology (EAEAT), Cairo, Egypt
‡SnT, University of Luxembourg, Luxembourg
§Department of Engineering, Thompson Rivers University (TRU), BC, Canada.
¶Corresponding Author
Emails: sehamibrahem0@gmail.com, motaz.ali@feng.bu.edu.eg, ola.m@eaeat.edu.eg,
waliullah.khan@uni.lu, owaqar@tru.ca, basem.mamdoh@feng.bu.edu.eg

***Abstract*—Long-range wide area network (LoRaWAN) is a promising low-power network standard that allows for long-distance wireless communication with great power saving. LoRa is based on pure ALOHA protocol for channel access, which causes collisions for the transmitted packets. The collisions may occur in two scenarios, namely the intra-spreading factor (intra-SF) and the inter-spreading factor (inter-SF) interference. Consequently, the SFs assignment is a very critical task for the network performance. This paper investigates a smart SFs assignment technique to reduce collisions probability and improve the network performance. In this work, we exploit different architectures of artificial neural networks for detecting collisions and selecting the optimal SF. The results show that the investigated technique achieves a higher prediction accuracy than traditional machine learning algorithms and enhances the energy consumption of the network.**

***Index Terms*—LPWAN, LoRa, LoRaWAN, Spreading Factor, Collision Detection, Deep Learning, Machine Learning**

I. Introduction

Due to the rapid growth in Internet of things (IoT) market and applications, several new technologies have been explored to enhance the performance of wireless communication between devices specially in terms of energy efficiency due to the nature of IoT node [1]. One of the important objectives is to design a low power wide area network (LPWAN) to support limited energy IoT nodes. LPWANs have to provide several features including low-power, long-range, low bit error rate, and low-cost wireless networking. Many network standards have been proposed for LPWAN such as Narrowband IoT (NB-IoT), Sigfox, and LoRa. However, LoRa is the most popular technology in LPWAN, which operates in the free-license ISM frequency band. LoRa supports long range communication, where a single LoRaWAN gateway may cover an entire city [2]. Moreover, LoRa transmission provides security mechanisms from end device to application server. The success of the LoRa transmissions depends on a group of configurable parameters, namely the spreading factor (SF), bandwidth (BW), transmission power (TP), carrier frequency (CF) and coding rate (CR). The SF is considered the most important parameter with great influence on the collision probability. Consequently, this work concentrates on using deep-learning (DL) based approach to detect collisions and provides an efficient SF assignment. Two different DL approaches are introduced. Fully connected neural network (FCNN) and convolutional neural network (CNN) are employed and compared with other traditional machine learning classifiers (support vector machine (SVM), decision tree (DT), and random forest (RF)). It is shown that the proposed DL-based SF assignment techniques give promising results.

Technical Overview of LoRa Network: LoRa technology is divided into two main parts: LoRa and LoRaWAN. The former, LoRa, is the physical layer presented by Semtech, which uses M-ary chirp spread spectrum (CSS) modulation [3]. Chirps are sinusoidal signals whose frequencies sweep linearly with time. LoRa technology may use different values of the bandwidth, however, the most widely used values are 125, 250, and 500 kHz. The higher bandwidth provides higher data rate and increases sensitivity against noise. On the other hand, the SF is considered the most important parameter in LoRa, which is defined as the ratio between symbol rate and the chirp rate. The SF's value can be any integer number from 7 to 12. The transmission success is affected by the allocated SF to the end node. Higher SF increases the signal-to-noise ratio (SNR), transmission range and time on air (ToA) while decreasing the data rate [4]. According to specification, the transmit power in LoRa network can take value between 2 and 20 dBm, where the default value is 14 dBm. The Carrier frequency depends on region restrictions and can take value between 137 MHz and 1020 MHz. The achievable data rate in bits per second in LoRa depends on SF, BW, and CR and can be mathematically expressed as $R_b = SF.\frac{\frac{4}{4+CR}}{\frac{2^{SF}}{BW}}$ [3]. On the other hand, LoRaWAN added the upper layer protocol developed by LoRa Alliance, which is built on top of the LoRa physical layer. The network components of LoRaWAN are the end node, the gateway, the network server, and the application server, which are shown in Fig. 1. The end node transmits on one of the available frequency channels at random. The number of packets per day is not limited

2332-5666/22/$31.00 ©2022 IEEE
DOI 10.1109/ICDCSW56584.2022.00043

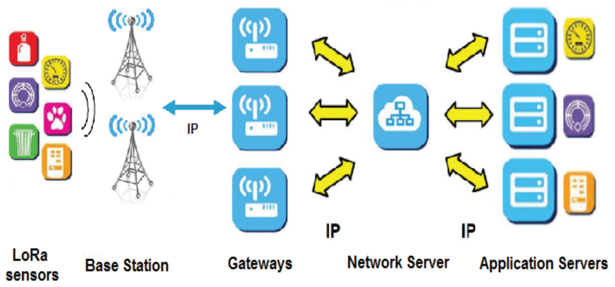


Fig. 1: LoRaWAN architecture [5]

but end node is restricted by duty cycle limitation. The end node and the gateway are connected through LoRa signals, while the gateway is connected to network server via WiFi or Ethernet. The gateway can receive packets from all end nodes and forward these packets toward the network server. The network server performs many tasks as filtering duplicated packets, handling the acknowledgments, and forwarding the application payloads. The application server performs many jobs like the decryption and managing users authentication [4].

Literature Survey on SFs Allocation: The process of finding the proper SF for a LoRaWAN device is critical for long-term performance. There are many approaches to allocate SF to end nodes including equal distance-based, gateway sensitivity-based, and SNR-based [6]. All these approaches depend on distance from gateway in unconfirmed mode. In [6], the authors proposed two SF allocation schemes. The first is the channel-adaptive SF recovery algorithm, which depends on channel status in the network to increase or decrease SF. The second technique assigns SFs to end nodes based on their sensitivities. The results show that the proposed techniques achieve higher packet delivery ratio when compared with traditional SF assignment techniques. In [7] the authors utilized the lowest possible SF assignment scheme to enhance the performance of the network. In this technique, the end node can determine the distance from gateway by observing the received signal power. If the received power is strong, this means end node is near to the gateway and can reduce SF to reduce power consumption.

Due to the recent wide-spread of machine-learning (ML) applications and algorithms in communication networks [8]–[17], several authors have tried to explore ML techniques to enhance the SF assignment process in LoRaWAN. The authors in [7] have proposed a SF assignment technique using traditional ML-based support vector machine (SVM) and decision tree (DT) techniques. A comparison of classification methods for allocating SFs in LoRa networks is proposed in [18]. The three used techniques are k-nearest neighbor (KNN), Naive Bayes, and SVM. This work is based on the fact that SF assignment's problem can be considered as a classification problem since the SF can have certain values (from 7 to 12). The experiment is conducted in FLoRa simulator environment. The result showed that SVM and KNN are better in predicting the optimal SF. In [19] the authors used K-means clustering to allocate SF in a large LoRa network. In the proposed technique, the range of each SF annulus is defined. They first computed the limits of the outermost SF ring, SF12, before moving on to the inner bounds of lower SFs. They looked at how the distance between end-devices and the gateway, as well as the number of nodes in each SF, affected network performance. The proposed technique improves coverage probability of the network by 5%.

The motivation behind this work is to exploit several ML and deep-learning (DL) architectures to enhance the performance of LoRa resource allocation by detecting collisions and selecting the optimal SF for the end nodes. In this work a smart DL-based SF assignment technique is proposed using fully connected neural network (FCNN) and convolutional neural network (CNN). A comparison between classifiers used in [7],i.e., SVM, DT, and other classifiers including FCNN, CNN, and RF is performed in terms of accuracy, packet delivery ratio (PDR) , and transmit energy consumption. The evaluation Metrics are calculated for different topologies and end nodes.

This paper is organized as follows: Section II presents the proposed collision detection and SF Assignment technique, while the performance evaluation is discussed in Section III. Finally Section IV concludes the paper.

II. Proposed DL-Based SF Assignment Technique

Different from traditional ML techniques that utilize numerous feature extraction algorithms to handcrafted engineering features before applying learning algorithms, DL techniques automatically extract features using several layers of artificial neural networks (ANNs) [20], which try to emulate the human brain by combining data inputs, weights, and bias. These components work together to effectively recognize, classify, and characterize data objects. There are different types of neural networks such as multi-layer perceptrons (MLPs), recurrent neural network (RNN), FCNN, and CNN. In this work, we focus on using FCNN and CNN for collision detection in LoRaWAN.

Data-Driven Multi-Gateway LoRaWAN System: In this work, we assume a multi-gateway LoRaWAN system, where several end-nodes are randomly distributed and communicating with the network server (NS) through three gateways. Since it is difficult to force one SF assignment algorithm for every network distribution, it is imperative to use smart algorithms. Similar to the work on [7], we assume that the NS is monitoring the network performance and use ML to learn the transmission success and failure of nodes at different coordinates with their SFs. The failure is usually due to the collided transmissions. Those information can be used by the NS to train different types of classifiers to predict collisions of future transmission using a specific location-SF combination. Consequently, exploiting those predictions to assign SFs to end nodes taking into consideration their coordinates and the collision probability.

Dataset Generation: The simulation tool in [7] generates a simulated dataset with three features, namely the x and y

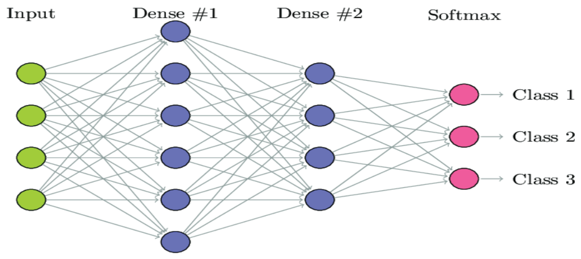


Fig. 2: Four layers fully connected neural network [21]

coordinates of the node in addition to the SF of the transmission. Both x and y assume continuous values, while SF assume integer number from 7 to 12. The output or the label of the dataset is defined as the result of the transmission, which takes one of three possible values (successful, interfered, and under sensitivity). We investigate both FCNN and CNN, which deals differently with the data samples. For the FCNN model, we use the tabluar data (x, y, SF) directly as an input to the model, while for the CNN model we have converted these data into images since CNN only takes images as input. In the following, we briefly describe both FCNN and CNN.

Fully Connected Neural Networks (FCNNs): are composed of a series of fully connected dense layers, such that every neuron is linked with every neuron in the adjacent layer as shown in Fig. 2. The outputs of the n^{th} layer is given as non-linear combination of the output of the preceding layer as follows [21]:

$$A^n = g^n(W^n \ A^{n-1} + b^n) \tag{1}$$

where W and b are the weights and biases of the n^{th} layer, while $g(.)$ is a non-linear activation function.

Convolutional Neural Networks (CNNs): are widely used in image classification, facial recognition, and semantic segmentation. It's built to learn spatial hierarchies of features, from low-level to high-level patterns, automatically and adaptively [22]. In comparison to a FCNN of the same size, a CNN contains less parameters because it is based on weight sharing, where kernels are shared across all picture places as shown in Fig. 3. CNNs are mainly composed of three types of layers, namely the convolution layer, pooling layers, and fully connected layers [20].

- Convolution Layer: several operations are performed in this layer as linear convolution and non-linear activation operations. Learnable kernels are convolved using feature maps from preceding layers. The output feature maps are formed by passing the kernel output through activation function [20] as shown:

$$x_j^L = F(\sum x_i^{L-1} k_{ij}^L + b_j^L) \tag{2}$$

  where: k_{ij}^L is the kernel for the present layer. The basic parameters in convolution are kernel size, number of kernels, stride, padding, and activation function.
- Pooling Layer: The pooling process is a down-sampling procedure that decreases the feature maps size in order to reduce variance and distortion. In pooling, the average or the maximum over a window of size k is taken, with a stride equal k. The most famous types of pooling are max pooling and global average pooling. The hyper-parameters in pooling are pooling method, filter size, stride, and padding.
- Fully Connected Layer: finally, the features which extracted from previous processes are mapped to output of the network by using a group of fully connected layers. The output presents the probabilities of each class in classification problem.

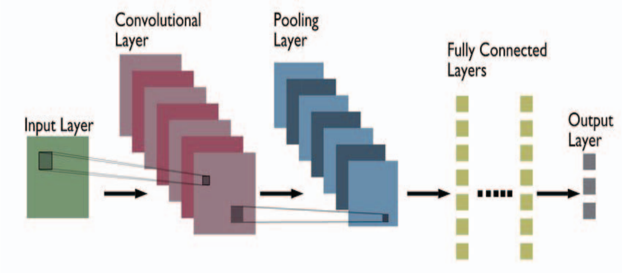


Fig. 3: Architecture of CNN [23]

The Structures of the Proposed Models: In this work, we have adopted two DL models using FCNN and CNN architectures. The FCNN model is composed of five layers with sigmoid activation function. On the other hand, the CNN model consists of two 2d convolution layers with kernel size = 3, stride = 1, padding = 1 and one fully connected layer. The max pooling uses a kernel size = 2 and stride = 2. The hyper-parameters for both FCNN and CNN are in Table I.

The input for FCNN is the data sample (3X1) vector, while the same data samples are fed to the CNN model after converting into (2X2) array using the library tab2img (https://github.com/nicomignoni/tab2img). The (2X2) images are then up-scaled to three channels of (16x16) array each using nearest neighbor interpolation. An example of forming an image that has been used for training our CNN with their corresponding (x, y, SF) is shown in Fig. 4.

III. Performance Evaluation

A. Simulation Environment

The LoRa simulator in [7] is used to generate dataset. At the first run, the simulator assumes a random SF scheme. When the random SF simulation phase is completed, the transmission logs are concatenated into three feature columns

TABLE I
Parameters of Proposed FCNN and CNN

Parameter	FCNN	CNN
loss function	Cross Entropy Loss	Cross Entropy Loss
learning rate	0.001	0.001
batch size	32	32
optimizer	Adam	Adam
epochs	15	15
activation function	sigmoid	ReLU
class weight	no	yes
input type	vector of lengh 3	images (16 *16 *3)

(x, y, SF) in addition to one label column that represents the collision status or label. Once this dataset is generated, it is used to train a set of classifiers. Then, the trained classifiers are used in the second simulation phase for collision prediction.

The output of the trained classifiers can be used to propose an SF assignment algorithm to avoid collision events. The simulator can adjust the SF according to the output class label from the classifiers. For every transmission, the simulator check the transmission result label for the lowest possible SF, if the predicted transmission label is interfered then simulator increase SF and check again for transmission result label. If the predicted transmission result is successful, then simulator will continue work. If all SF values are checked and no successful transmission label is predicted then the simulator continue work with lowest SF.

The simulation parameters are: packet size = 60 byte, packet rate = 0.01 pps, and simulation duration = 3600 seconds. We simulate the system on different network size, where the size of the dataset is scaled with the number of deployed end nodes. As an example, for a radius of 7000 m and the 100 nodes, the dataset consists of 3653 samples and split into two groups: 80% of samples are used for training phase and 20% for testing phase. While at the same radius with 500 and 1000 nodes, dataset consists of 18211 and 36086 samples, respectively.

Communication Link Budget: The link budget can be used to measure the quality of a wireless links between the end nodes and the gateways. A wireless link's budget can be expressed as follows [7] :

$$P^{dBm}_{RX} = P^{dBm}_{TX} + G^{dB}_{SYS} - L^{dB}_{SYS} - L^{dB}_{PATH} \tag{3}$$

where P^{dBm}_{RX} is received power at receiver, P^{dBm}_{TX} is transmit power of the transmitter. G^{dB}_{SYS} is transmitter and receiver antenna gains. L^{dB}_{SYS} is the transmitter and receiver line, circuit, and antenna losses. L^{dB}_{PATH} is the propagation path loss between transmitter and receiver antennas. In the used simulator, the sum of system gains and system losses is assumed to be +7 dB and 14 dBm maximum transmit power for European ISM band. Free space propagation loss can be mathematically expressed as [7]:

$$\begin{aligned} P^{dB}_{PATH} = 40\ (1 - 4 \text{ x } 10^{-3} \text{ x } h) \log_{10}(R) \\ -18\ \log_{10}(h) + 21\ l \log_{10}(f) + 80 \end{aligned} \tag{4}$$

where h is the gateway altitude, f is the frequency of the signal and R is the radius of the network. Since this work assumes SFs orthogonality, only single channel transmissions are used. So, it is assumed that h = 15 m and f = 868 MHz and 125 kHz bandwidth receive sensitivities are used. When the received signal power is greater than the gateway sensitivity and there is no interfering transmissions, the signal can be effectively decoded by the receiver.

Interference Model: For modelling LoRa interference between simultaneous but different SF LoRa signals, the simulator in [7] employs a signal to interference plus noise ratio (SINR) based threshold matrix. The SINR threshold matrix is used to determine if another signal interferes with a reference signal at the receiver. $T_{i,j}$ is the SINR margin in dB between the reference signal with SF = i and the interfering signal with SF = j to decode the reference signal correctly. If there are many interfering signals, the reference signal must satisfy the margin for the total received power of all interfering signals for each SF [24]. The SINR threshold can be calculated as [7]:

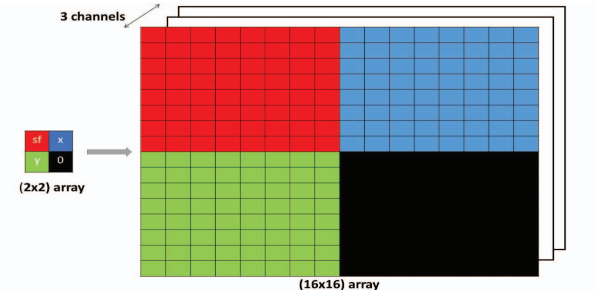


Fig. 4: An example of forming an image for CNN

$$SINR_{i,j} = \frac{P_{rc,0}}{\sum_{l \in I_j} P_{rc,l}} \tag{5}$$

where $P_{rc,0}$ is received power of reference signal and $P_{rc,l}$ is received power of interfering signal for the j^{th} SF. The packet with SF = i can survive from all interference if it satisfies the following condition for every SF= j:

$$SINR^{dB}_{i,j} > T_{i,j} \tag{6}$$

B. Evaluation Metrics

We have adopted three evaluation metrics to show the performance of the proposed models, namely the prediction accuracy, the packet delivery ratio, and the total transmit energy consumption, which are defined as follows:

Prediction Accuracy: is the ratio of the number of correct predictions of the model to the total number of samples (i.e., ground truth).

Packet Delivery Ratio (PDR): is percentage of the total number of packets successfully received by gateway, which is given by:

$$PDR = \frac{\sum \text{total packets correctly received}}{\text{total packets sent}} * 100 \tag{7}$$

Total Transmit Energy Consumption (TEC): is the total amount of energy in joule spent by the network to send packets. It depends on SF, BW, CR, and TP. The correct choice of transmission parameters leads to decrease the transmit energy consumption. As number of nodes increase, the energy consumption increases. TEC is mathematically given as follows:

$$TEC = \sum_{i=0}^{n} T_P * T_S \tag{8}$$

where T_P is transmission power in watts, T_s is the transmission duration, and n is number of nodes.

C. Simulation Results

In this subsection, we discuss the achievable results for the traditional ML algorithms (SVM, DT) from [7], in addition to another traditional ML algorithm we investigated (the Random Forest ”RF”) and the proposed DL algorithms (FCNN and CNN).

Table II summarizes the prediction accuracy of all algorithms for three different area radii of 3000, 5000, and 7000 meters away from the gateway. For each radius, we investigated different number of deployed LoRa nodes (100, 500, and 1000 nodes). The given results are the average of three runs for each radius/number of nodes combinations (i.e., 3X3X3X5 = 135 runs in total, where 5 is the number of classifiers). Figure 5 shows that FCNN outperforms the others techniques in the prediction accuracy at different network radii and different number of nodes. The results show that CNN achieves the worst accuracy among all classifiers. This is because normally CNNs work best on images that have spatial relationships (i.e. face images where eyes, nose, mouth, etc have spatial relations onto the face). In our case, however, there is no spatial correlation, therefore the training of CNN becomes less efficient.

It is noteworthy that the simulation parameters have a considerable influence on the distribution of different labeled data points in the dataset and generate unbalance dataset. As an example, for a large topology radius, number of under sensitivity transmission labels increases, while at small topology with large number of nodes, the number of interfered transmission labels increase. When the training dataset is visualized in 3D space (feature space), where x, y and z-axes correspond to features x, y, and SF respectively, it looks very evident how the entanglement of differently labeled samples increases as number of nodes gets higher. The discriminability between samples becomes then very complicated and the ability of the classifier to create a proper decision boundary gets lower when more nodes are introduced. That gives an interpretation of why the prediction accuracy decreases with the increase of number of nodes and how the classifier is significantly affected by the quality of the dataset

Regarding the packet delivery ratio (PDR), we have made more investigations by using different number of deployed nodes from 50 to 1000 nodes with a step of 50 (i.e., 20 different cases), which leads to 900 simulation cycles (i.e., 3X20X3X5). The results show that ML algorithms outperforms both DL algorithms in terms of the network PDR, despite the fact that the prediction accuracy of smart FCNN is the highest. We have the following observation:

- The PDR is not affected directly by the prediction accuracy since successful prediction of interfered transmissions may not increase PDR. As mentioned in simulation environment section, the classifiers predict only the transmission result label for every transmission, while the simulator assigns SF value for transmission.
- Increasing number of nodes will increase the interference and therefor PDR decreases.

Finally, the results in Fig. 7 show that DL approaches save energy consumption, where both smart FCNN and CNN outperform other techniques (SVM - DT - RF), which is a major advantage for low-power IoT applications.

TABLE II
Prediction Accuracy of Different Classifiers

Radius(m)	No.of nodes	SVM	DT	FCNN	CNN	RF
3000	100	80.9	85.0	95.6	84.3	86.2
	500	71.4	68.8	84.6	62.8	76.8
	1000	71.3	69.9	77.9	60.1	72.6
5000	100	83.4	86.5	95.7	79.6	87.7
	500	69.0	66.9	84.6	62.9	75.0
	1000	70.7	68.9	77.5	65.4	71.8
7000	100	81.7	87.1	92.3	79.9	88.3
	500	71.1	67.6	81.2	69.2	75.7
	1000	71.5	69.9	72.0	55.0	72.5

IV. Conclusion

In this work, we present smart SF assignment techniques using DL abd ML algorithms. FCNN and CNN models are presented, where the results are compared with different classifiers including SVM, DT, and RF. The simulation results show that FCNN outperforms other techniques in prediction accuracy and saving power while CNN save power only. On the other hand, DT achieves the highest packet delivery ratio. This fact make DL algorithms more suitable for LoRaWAN network.

References

[1] D. M. Ibrahim, “Internet of things technology based on lorawan revolution,” in *2019 10th International Conference on Information and Communication Systems (ICICS)*, 2019, pp. 234–237.

[2] M. Bor and U. Roedig, “Lora transmission parameter selection,” in *2017 13th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, 2017, pp. 27–34.

[3] R. Marini, K. Mikhaylov, G. Pasolini, and C. Buratti, “Lorawansim: A flexible simulator for lorawan networks,” *Sensors*, vol. 21, no. 3, 2021.

[4] E. Sallum, N. Pereira, M. Alves, and M. Santos, “Improving quality-of-service in lora low-power wide-area networks through optimized radio resource management,” *Journal of Sensor and Actuator Networks*, vol. 9, no. 1, 2020.

[5] J. de Carvalho Silva, J. Rodrigues, A. Alberti, P. Šolić, and A. Aquino, “Lorawan - a low power wan protocol for internet of things: a review and opportunities,” 07 2017.

[6] A. Farhad, D.-H. Kim, and J.-Y. Pyun, “Resource allocation to massive internet of things in lorawans,” *Sensors*, vol. 20, no. 9, 2020.

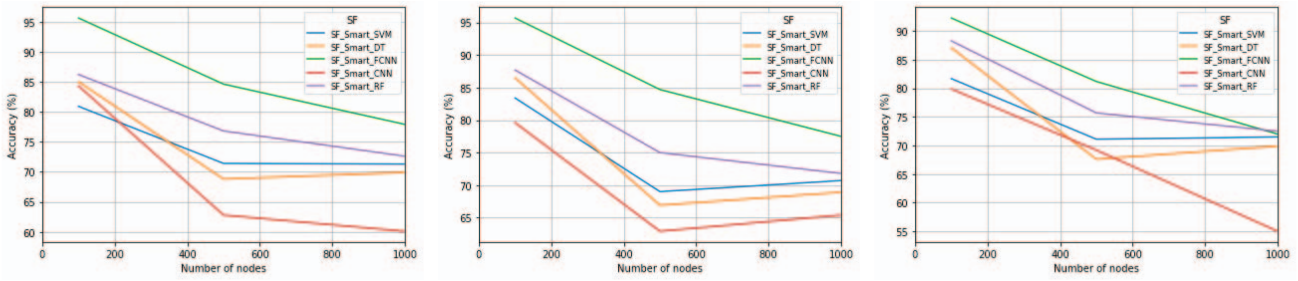
[7] T. Yatagan and S. Oktug, “Smart spreading factor assignment for lorawans,” in *2019 IEEE Symposium on Computers and Communications (ISCC)*, 2019, pp. 1–7.

[8] A. B. Zaky, J. Z. Huang, K. Wu, and B. M. ElHalawany, “Generative neural network based spectrum sharing using linear sum assignment problems,” *China Communications*, vol. 17, no. 2, pp. 14–29, 2020.

[9] A. A. Aziz El-Banna, A. B. Zaky, B. M. ElHalawany, J. Zhexue Huang, and K. Wu, “Machine learning based dynamic cooperative transmission framework for iout networks,” in *2019 16th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, 2019, pp. 1–9.

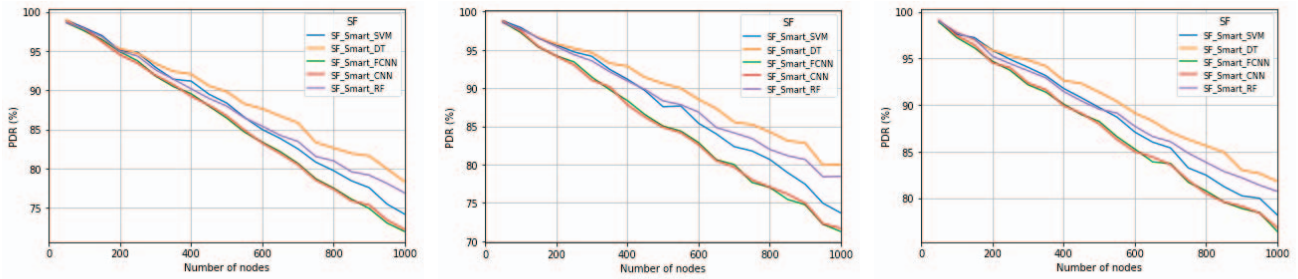
[10] A. A. A. El-Banna, B. M. ElHalawany, A. B. Zaky, J. Z. Huang, and K. Wu, “Machine learning-based multi-layer multi-hop transmission scheme for dense networks,” *IEEE Communications Letters*, vol. 23, no. 12, pp. 2238–2242, 2019.

[11] S. Hashima, B. M. ElHalawany, K. Hatano, K. Wu, and E. M. Mohamed, “Leveraging machine-learning for d2d communications in 5g/beyond 5g networks,” *Electronics*, vol. 10, no. 2, 2021.



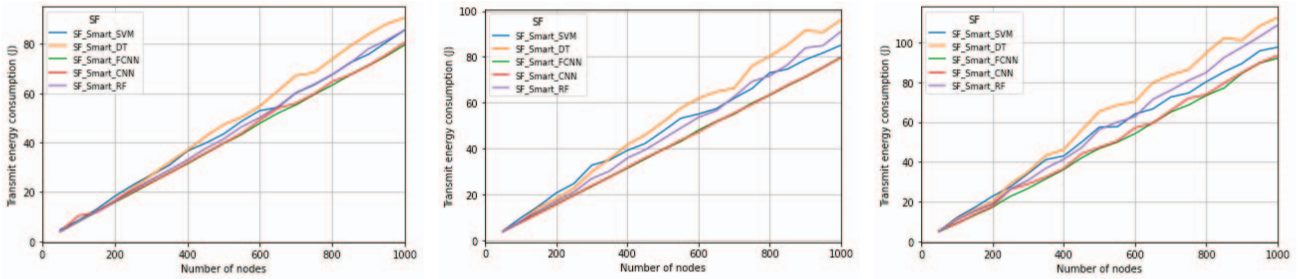
(a) (b) (c)

Fig. 5: Prediction accuracy at different radius: (a) r = 3000m (b) r = 5000m (c) r = 7000m



(a) (b) (c)

Fig. 6: Packet delivery ratio at different radius: (a) r = 3000m (b) r = 5000m (c) r = 7000m



(a) (b) (c)

Fig. 7: Transmit energy consumption at different radius: (a) r = 3000m (b) r = 5000m (c) r = 7000m

[12] A. A. A. El-Banna, K. Wu, and B. M. ElHalawany, "Application of neural networks for dynamic modeling of an environmental-aware underwater acoustic positioning system using seawater physical properties," *IEEE Geoscience and Remote Sensing Letters*, pp. 1–5, 2020.

[13] B. M. ElHalawany, K. Wu, and A. B. Zaky, "Deep learning based resources allocation for internet-of-things deployment underlaying cellular networks," *Mobile Networks and Applications*, vol. 25, no. 5, pp. 1833–1841, 2020.

[14] W. U. Khan, T. N. Nguyen, F. Jameel, M. A. Jamshed, H. Pervaiz, M. A. Javed, and R. Jäntti, "Learning-based resource allocation for backscatter-aided vehicular networks," *IEEE Transactions on Intelligent Transportation Systems, Early Access*, pp. 1–15, 2021.

[15] B. M. ElHalawany, S. Hashima, K. Hatano, K. Wu, and E. M. Mohamed, "Leveraging machine learning for millimeter wave beamforming in beyond 5g networks," *IEEE Systems Journal, Early Access*, pp. 1–12, 2021.

[16] U. S. Toro, B. M. ElHalawany, A. B. Wong, L. Wang, and K. Wu, "Machine-learning-assisted signal detection in ambient backscatter communication networks," *IEEE Network*, vol. 35, no. 6, pp. 120–125, 2021.

[17] S. Hashima, Z. M. Fadlullah, M. M. Fouda, E. M. Mohamed, K. Hatano, B. M. ElHalawany, and M. Guizani, "On softwarization of intelligence in 6g networks for ultra-fast optimal policy selection: Challenges and opportunities," *IEEE Network, Early Access*, pp. 1–9, 2022.

[18] B. Christos, G. Apostolos, S. S. A. Katsampiris, and P. Nikolaos, "Spreading factor analysis for lora networks: A supervised learning approach," 2021, pp. 344–353.

[19] A. H. R. D. S. Muhammad Asad Ullah, Junnaid Iqbal and H. Alves, "K-means spreading factor allocation for large-scale lora networks," *Sensors*, p. 1, 2019.

[20] M. Z. Alom, T. M. Taha, C. Yakopcic, S. Westberg, P. Sidike, M. S. Nasrin, M. Hasan, B. C. Van Essen, A. A. S. Awwal, and V. K. Asari, "A state-of-the-art survey on deep learning theory and architectures," *Electronics*, vol. 8, no. 3, 2019.

[21] C. Pelletier, G. I. Webb, and F. Petitjean, "Temporal convolutional neural network for the classification of satellite image time series," *Remote Sensing*, vol. 11, no. 5, 2019.

[22] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, "Convolutional neural networks: an overview and application in radiology," *Insights into imaging*, vol. 9, no. 4, pp. 611–629, 2018.

[23] A. H. Yazdani Abyaneh, A. Hossein Gharari, and V. Pourahmadi, "Deep neural networks meet csi-based authentication," 11 2018.

[24] D. Magrin, M. Centenaro, and L. Vangelista, "Performance evaluation of lora networks in a smart city scenario," in *2017 IEEE International Conference on Communications (ICC)*, 2017, pp. 1–7.