Deep Learning-based LOS/NLOS Classification for Reliable Communication in Industrial IoT

Annapurna Pradhan
Department of Electrical Engineering
National Institute of Technology
Rourkela, India
pradhan.annapurna37@gmail.com

Susmita Das
Department of Electrical Engineering
National Institute of Technology
Rourkela, India
sdas@nitrkl.ac.in

Gyana Ranjan Mati
Department of Electrical Engineering
National Institute of Technology
Rourkela, India
gyana.mati@gmail.com

Ankit Oscar Xalxo
Department of Electrical Engineering
National Institute of Technology
Rourkela, India
ankitoscar911@gmail.com

***Abstract*—The optimal channel selection for data transmission is essential for reliable communication in an indoor environment like industrial IoT (IIoT). Due to the presence of complex objects in the indoor factory environment, signals might get reflected. This leads to reliability loss and degradation of transmitted signal quality by increasing signal outage. Again, the optimal channel selection for efficient scheduling of the heterogeneous data packets generated by delay-sensitive ultra-reliable low latency (URLLC) service and delay-tolerant broadband service in IIoT demand for accurate identification of wireless link status. Therefore, the identification of wireless channel status like Line-of-Sight (LOS), None-Line-of-Sight (NLOS), and Multi-Path (MP) between the transmitter and the receiver becomes essential to reduce packet error probability in IIoT. In this regard, we propose a deep learning-based classifier model using Convolutional Neural Network (CNN) and Long Short Term Memory (LSTM) to identify the LOS/NLOS or MP signals and enhance the reliability of the signal by improving system throughput and accurate positioning. We compare the performance of the proposed model with various machine learning classifier models to evaluate the performance of the proposed CNN-LSTM model. We have used an open-source dataset collected from two different indoor industrial sites to be used for the training and testing of the classifier models. We have evaluated the performance based on the accuracy and time complexity of the proposed classifier model, which shows superiority in comparison to baseline machine learning models. Additionally, the results show that the system Bit Error Rate (BER) improved significantly with the optimal channel selection during scheduling of heterogenous data type using the proposed CNN-LSTM model.**

***Index Terms*—LOS, NLOS, multipath, URLLC, CNN-LSTM, CNN, MLP, IIoT, 5G.**

I. Introduction

THE rapid advancement of Internet-of-Things (IoT) technology and wireless communication facilitates various futuristic smart applications. The introduction of the fifth-generation (5G) wireless systems brings innovative services that have the potential to enable mission-critical applications like industrial automation, autonomous driving, smart grid, and smart city operations with the help of reliable data transmission among IoT devices while improving the overall user experience [1]. Maintaining the reliability of data transmission becomes one of the essential requirements for communication. In these large-scale IoT applications, billions of smart devices are interconnected to transmit data wirelessly.

For instance, applications like industrial automation in industrial IoT (IIoT) are a classic example of large-scale IoT network. Generally, IIoT contains multiple access points and receiver devices simultaneously transmitting and receiving data packets in an indoor environment [2]. In such a case, the presence of a large number of objects in the indoor environment of the IIoT increases the complexity of signal transmission without getting affected by the channel condition, which is a challenging task. The characteristics of the materials in the indoor area hugely affect the data transmission while introducing blockage and scattering [3]. Moreover, IoT users have heterogeneous Quality-of-Service (QoS) requirements like low latency, high data rate, and reliable data transmission. Therefore, identification of channel status becomes crucial for providing reliable data transmission and determining optimal scheduling policy for these incoming data traffic in such scenarios [4].

Accurate identification of the type of channel is one of the essential requirements for reliable communication. The research work for developing enabling techniques to classify the type of wireless channel is gaining attention. In this regard, the work in [5] used a deep learning-based technique to classify the type of channel using the channel impulse response data as input. Here, the authors used a Convolutional Neural Network (CNN) and Long Short Term Memory (LSTM) model to classify the LOS/NLOS channel from accurate indoor positioning in an ultra-wideband (UWB) system. Similarly, the work in [6] and [7] use the CNN-based classifier for LOS/NLOS classification. Specifically, the work in [6] proposed a hybrid scheme containing both deep learning and transfer learning to classify NLOS paths in an unmeasured environment. Likewise, the research work in [7] utilizes a CNN-based LOS/NLOS classification model in an urban 3D massive MIMO system. In another work in [8], the authors have proposed a Morlet wave transform and CNN classifier model to identify LOS/NLOS.

A. Motivation and Contributions

The accurate channel type identification is helpful in assigning specific sub-channels to users with heterogeneous QoS requirements in an IIoT scenario [8]. Most of the recent works in this area utilize supervised learning models to accurately classify the channel type. However, the wireless environment is unpredictable. Therefore, the authors in [9] have proposed an unsupervised learning classifier that selects the useful channel features to classify LOS/NLOS. In [10], the authors proposed a classifier based on RSSI (Received Signal Strength Indicator) data to identify LOS/NLOS channels. The proposed classifier uses the statistical features extracted over multiple channel measurements and decides the threshold level according to the indoor environment. Most of the researchers utilized the LOS/NLOS classification outcome for finding accurate positioning in an indoor environment only while limiting the utilization of these results for efficient scheduling [11]. However, none of the work used the LOS/NLOS and Multi-Path (MP) classification simultaneously to address the scheduling challenges of heterogeneous data traffic to the best of our knowledge.

Motivated by this, we propose a CNN-LSTM-based LOS/NLOS and MP channel status identifier, which will be helpful for scheduling the incoming mixed data traffic proactively in an IIoT system. We know that the performance of deep learning-based classifiers has been superior to many ML-based models. Therefore, in this work, we proposed a CNN and LSTM deep learning-based classifier model for LOS/NLOS and MP detection and classification. We have utilized the channel classification results to accurately allocate the channels to delay-sensitive and delay-tolerant services for reliable communication.

B. Paper Organization

In this paper, Section II presents the system model of the proposed CNN-LSTM model for LOS NLOS or MP classification. Then, in Section III, we propose an optimal scheduling scheme based on incoming traffic delay constraints and the channel type. We provide the simulation setup and performance analysis of the CNN-LSTM classifier and proposed scheduler in section IV. Finally, Section V includes the conclusion and future scope of our work.

II. System Model

We consider an IIoT scenario, where multiple transmitters transmit data packets to multiple IoT devices in an indoor environment. The transmitting access points(AP) and receiving devices are equipped with single antennas. Let the IIoT system contain I single antenna device nodes. A downlink communication scenario is considered where the channel bandwidth is denoted as B, which is divided into R Physical Resource Blocks (PRBs) to accommodate user requirements.

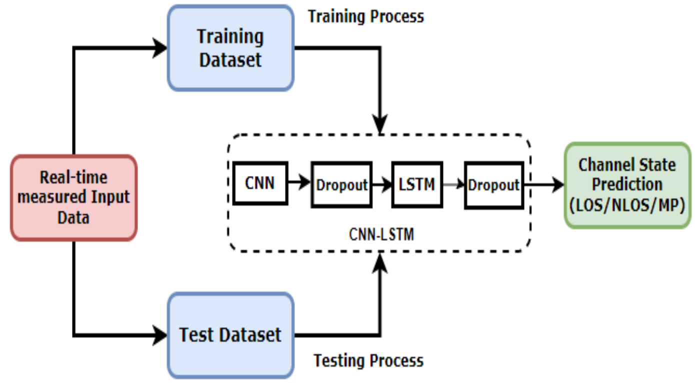

Fig. 1. Proposed CNN-LSTM Deep learning model

A. Wireless traffic model

The considered IIoT system has a mixed data traffic of both delay tolerant signal, i.e., enhanced mobile broadband (eMBB) signal and delay-sensitive ultra-reliable low latency communication (URLLC) signal. Both types of signals have different QoS requirements. Generally, eMBB traffic requires a high data rate of transmission with delay-tolerant service requirements. However, URLLC traffic needs highly reliable signal transmission within a strict delay constraint. If the packet transmission delay exceeds the deadline constraints, then the packet will be dropped, which reduces the reliability by increasing the error probability. Therefore, efficient scheduling of these mixed traffic types in an indoor IIoT environment is challenging. In such a scenario, accurate channel state information is desired for efficient scheduling of these heterogeneous data types in the suitable channel to satisfy the QoS requirements of individual users.

Each user has a corresponding queue, and the queue length is dependent on the delay constraint of the packets. Here the delay-sensitive traffic is scheduled on the mini-slot, and the delay-tolerant traffic is scheduled at the start of a time slot. Let the total end-to-end delay constraint be D_e, which is equivalent to two mini-slot time periods. Then, the reliability of the transmission for the user i can be defined as the probability of the packet satisfying the delay deadline constraint as given below:

$$P\{D_i < D_{max}\} \geq D_e, \tag{1}$$

where D_{max} is the maximum end-to-end delay constraint of a packet. In order to reduce packet drop probability, we need to schedule the arrived traffic and allocate the PRBs according to the QoS requirement of the incoming traffic. Therefore, to minimize packet error probability and enhance the reliability of signal transmission, identifying the ideal channel condition is very much essential. Hence, we employ a CNN-LSTM sequence prediction model to classify and predict the channel status accordingly.

B. Proposed CNN-LSTM Network

CNN has the ability to automate the feature extraction process to learn the patterns without manual feature engineering. It uses multiple layers of filters to extract the features, which makes it resilient to noise. The accuracy of CNN is higher compared to other deep learning models. However, it requires a larger dataset to train and achieve the desired accuracy. The CNN contains multiple hidden layers comprised of convolution layers, pooling layers, and fully connected layers. Basically, the feature extraction and high-level abstraction of input data are done at the convolution layer using multiple kernel filters. The pooling layers then reduce the dimensionality of the output feature map obtained from the convolution layer. In this work, we have used the MaxPooling function to reduce the computation load and time complexity. The convolution and MaxPooling layers together extract the features and reduce the dimensionality of the input data. Then, a fully connected layer uses the softmax function to select the probability of classes. The output of a fully connected layer is the class with the highest probability.

We have used a CNN-LSTM architecture to classify the link between transmitter and receiver as a LOS/NLOS, or multipath channel. As the data is a time series data, we have used LSTM to predict the channel status based on the incoming data pattern. Long Short term memory (LSTM) is an advanced version of recurrent neural network (RNN) that can provide long-term dependencies to model chronological sequences. Basically, LSTM is composed of three gates as forget gate (f), the input gate (i), and the output gate (y) [5]. Forget gate updates the cell state from the previous epoch. In the forget gate, the input vector x_t is fed to the sigmoid function. Then, the output of the sigmoid function is the output vector y_t, which is multiplied by the state vector C_{t-1} from the previous epoch. The output vector can be represented as:

$$f_t = \sigma(w_f.[h_{t-1}, x_t] + b_y) = \sigma\phi_f \tag{2}$$

where w_f is the weight vector, b_y is the bias vector, and σ is the sigmoid function. The output vector f_t decides the degree to remember or forget the previous state vector C_{t-1}. Then, the input gate contains sigmoid and tanh functions to process input data x_t. The input gate function is given as:

$$i_t = \sigma(w_i.[h_{t-1}, x_t] + b_i) \tag{3}$$

$$\hat{C}_t = tanh(w_c.[h_{t-1}, x_t] + b_c) \tag{4}$$

where w and b are the weight and bias vectors. At last, the output gate decides the output of the LSTM network. For improving the generality and prediction accuracy, parameter optimization is required. This requires a proper arrangement to address the overfitting problem in the deep learning models. In our work, we use dropout after the fully connected layer. A dropout rate is defined in the proposed CNN-LSTM classifier to ignore some neurons and nodes during model training to counter overfitting. We have not used the bidirectional LSTM here because it requires a large number of parameters to be determined during training which ultimately increases the training time and degrades clarification accuracy.

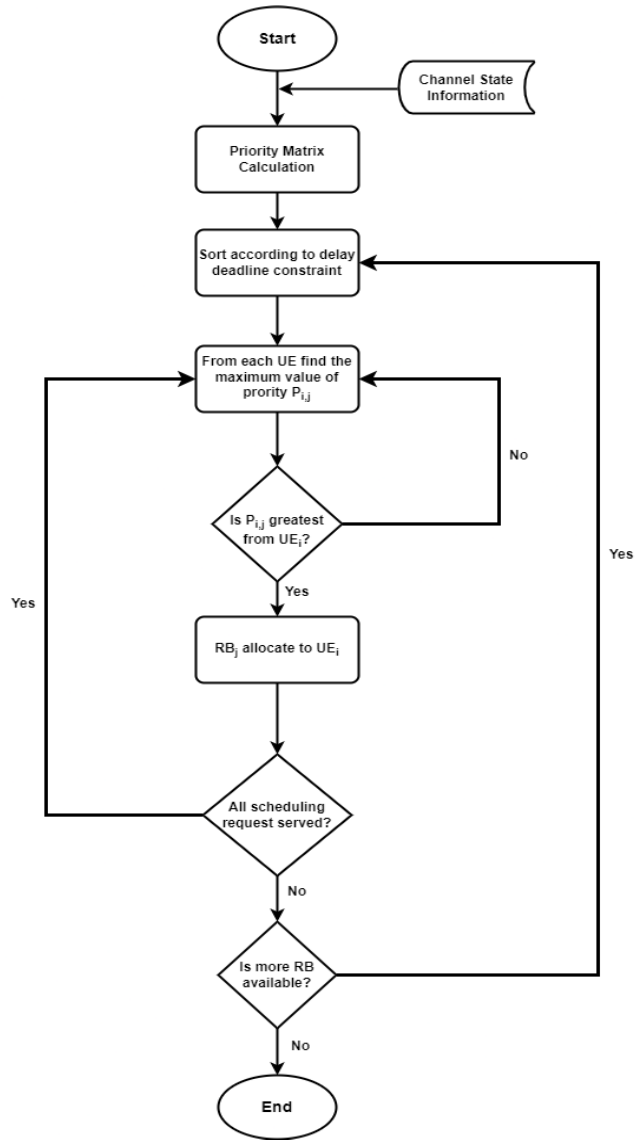

Fig. 2. Proposed Scheduling and Resource Allocation scheme

III. Proposed Scheduling scheme

We propose a scheduling scheme based on the predicted result of the CNN-LSTM network, as shown in Fig. 2. After the channel status identification, the priority matrix is calculated based on the delayed deadline, and the incoming traffic is arranged in ascending order of latency constraint. Then, the scheduling request is served, and PRBs are allocated. The delay-sensitive services are given the highest priority and scheduled within two mini-slot time periods in an LOS channel to improve reliability and reduce packet drop. On the other hand, delay-tolerant services are scheduled on either LOS or MP channel. LOS channels are given the highest priority, then the MP channel. If a channel is identified as NLOS, then the scheduled data transmission quality in this channel is hugely degraded. Therefore, the LoS link is given priority to schedule the incoming traffic instead of NLOS and MP.

We calculate the Bit Error Rate (BER) as the performance evaluation metric for the proposed scheduler. Depending on the deadline constraint of the arrived packet, we update the priority queue status. Then, the earliest deadline packet is given the highest priority to be scheduled through the LOS path. If the delay constraint is below the threshold level, i.e., two mini slots, then it is termed a delay-sensitive service; otherwise, it is a delay-tolerant service. Then, according to the scheduling policy, BS schedules the available physical resource blocks (PRB) for the incoming traffic. Depending on the channel status, delay-sensitive services are scheduled in LOS paths, and delay-tolerant services are scheduled either on LOS, NLOS, or MP channels. First, the queuing delay of each device node is calculated and normalized, which can be represented as:

$$w_i^q(t) = \frac{q_n(t)}{\sum_{i=1}^{I} q(t)}, \tag{5}$$

where $q_n(t)$ is the queuing delay of device node i at time slot t. Then, the total delay at the device node is normalized and is represented as:

$$w_i^D(t) = \frac{D_n(t)}{\sum_{i=1}^{I} D_n(t)}. \tag{6}$$

Then, considering the queuing delay and total normalized delay at an IoT device, priority-based weights are assigned to each node i at the time slot t. This can be represented as,

$$P_i(t) = w_i^q(t) + w_i^D(t), \tag{7}$$

where P_i represents the sum of the total normalized delay at a user node i.Then, the PRB allocation is done proportional to the delay constraint $P_i(t)$.

IV. Simulation Study and Performance Analysis

For simulation, we use an open-source dataset collected in two different environments as given in [12]. The LOS, NLOS, and MP measurements were present in the dataset containing the RSSI value and CSI value at different timestamps. We build the classification model using the total number of 25000 samples randomly selected from the dataset. We use 20,000 samples for training and 5000 for testing the classifier model. Samples are randomly selected to prevent model overfitting. We employ Python with TensorFlow and Keras library to simulate the deep learning models using a PC with 16 GB RAM and an I7 CPU (3.3GHz). We train the other baseline classifier models, like MLP and CNN classifier models, with the proposed CNN-LSTM model to provide a comparison of accuracy in classifying the channel type. The training of each model is done with 20 epochs. The features like RSSI, CIR, received power level, and first path power level are selected for training the deep learning models. Then, the performance of the classifiers is evaluated in terms of classification accuracy, training time, and testing time. We calculate the performance

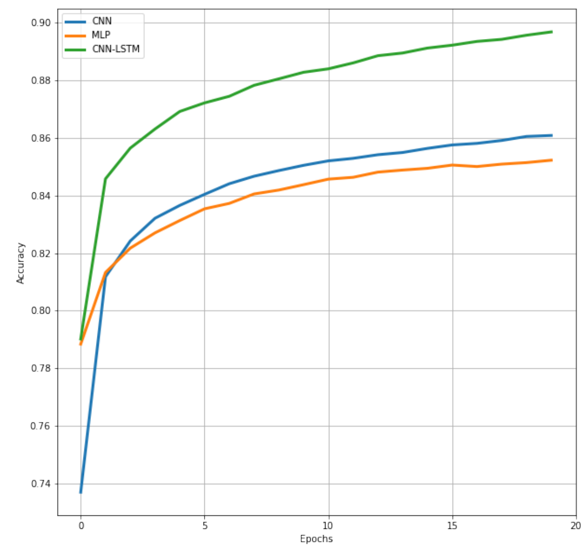

Fig. 3. Classification accuracy comparision

evaluation metrics like precision, recall, and F1 score of the classifier models as [12]:

$$Recall = TP * [1/(TP + FN)] \tag{8}$$

$$Precision = TP * [1/(TP + FP)] \tag{9}$$

$$F1\, score = 2 * \frac{Precision.Recall}{Precision + Recall} \tag{10}$$

Here, TP is the true positive; FP and FN denote false positive and false negative values, respectively. Basically, precision is the percentage of relevant or positive results, whereas recall indicates the positive cases where the classifier prediction is correct. On the other hand, the Fl-score conveys the balance between the precision and recall value of a particular classifier model.

The comparison of the corresponding precision, recall, and F1 score of the proposed CNN-LSTM with that of the CNN, and MLP models are given in Table I. Here, we can observe the superiority of the CNN-LSTM classifier performance with respect to the other baseline models. Additionally, we provide a comparison of the time taken by the deep learning models for training and testing on the given dataset in Table II. Here, we can observe that the proposed CNN-LSTM model takes more time to train in comparison to CNN and MLP models. This is due to the presence of more dense layers in the CNN-LSTM model. In Fig.3, we provide a comparative analysis of ML models in terms of their classification accuracy on the test dataset. We can observe that the proposed CNN-LSTM classifier significantly improves the classification accuracy in comparison to the CNN and MLP classifiers.

TABLE I
CLASSIFICATION MODEL PERFORMANCE COMPARISON

Models		LOS			NLOS			MP	
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
CNN	0.82	0.88	0.85	0.91	0.82	0.86	0.86	0.88	0.87
MLP	0.78	0.9	0.83	0.9	0.79	0.84	0.89	0.85	0.87
CNN-LSTM	0.85	0.92	0.89	0.91	0.88	0.9	0.92	0.88	0.9

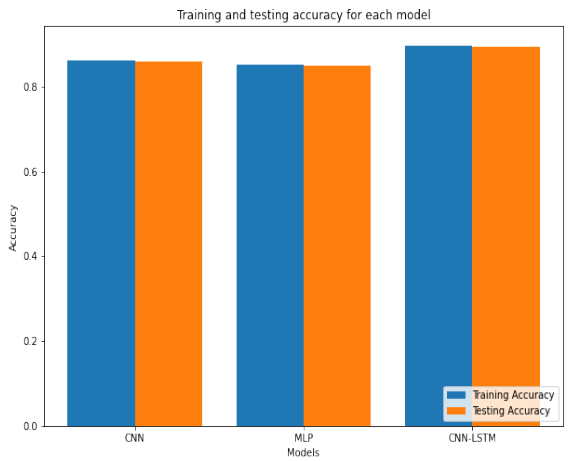

Fig. 4. Training and Testing accuracy of each model

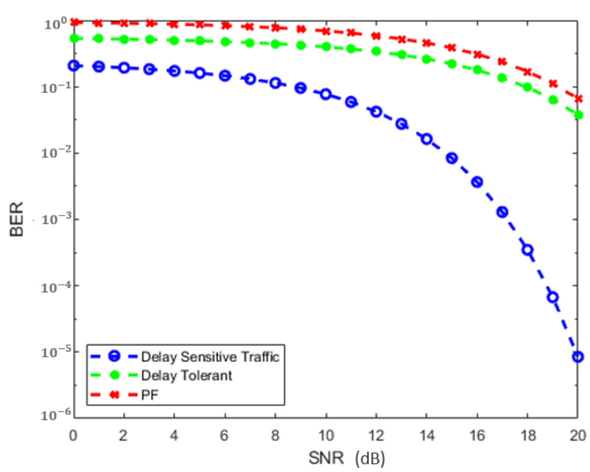

Fig. 5. BER vs. SNR (dB)

TABLE II
TRAINING TIME AND TESTING TIME OF CLASSIFIER MODELS

Model	Training Time(in s)	Testing Time(in s)
CNN	322.96	4.29
MLP	263.91	5.21
CNN-LSTM	627.88	5.17

Similarly, Fig.4 shows the accuracy of deep learning models in the training and testing phase after 10 epochs. We can observe that in both the training and testing phase, the prediction accuracy of the proposed CNN-LSTM model is better than that of the CNN and MLP models. A comparison of time taken for training and testing the deep learning models is presented in Table II. We can observe that the proposed CNN-LSTM model takes more time to train in comparison to the CNN and MLP models. This is due to the presence of more dense layers in CNN-LSTM compared to other classifier models.

Due to the superior performance of CNN-LSTM in terms of classification accuracy, we use this for identifying the channel state in an indoor environment. Then, depending on the prediction, the data packets are scheduled on the specific wireless sub-channels. The scheduler selects the LOS link for scheduling the delay-sensitive packets whose deadline is approaching by giving it the highest priority.

The performance of the proposed scheduling technique is evaluated in terms of the bit-error rate (BER) value for the specific type of service requirement. In Fig.5, we provide the BER performance of the proposed scheduler in comparison to the baseline Proportional Fair (PF) scheduler under various SNR conditions. We can observe the improvement in the BER value of the proposed scheduling scheme for both delay-sensitive and delay-tolerant services in comparison to PF. According to the proposed scheme, the delay-sensitive packets are scheduled in LOS channels, whereas delay-tolerant services can be scheduled on other sub-channels. This reduces the packet drop probability and improves the reliability significantly during the data transmission of heterogeneous service types.

V. CONCLUSION

In this paper, we propose a dynamic scheduling scheme based on predicted channel status in an IIoT scenario. We employ the CNN-LSTM deep learning model to accurately classify and identify the channel type between legitimate transmitter and receiver in IIoT i.e., LOS/NLOS or MP channel. The performance of the proposed model is evaluated in terms of classification accuracy, and the results show the superiority of the proposed CNN-LSTM model in comparison to other baseline models like CNN and MLP. The output of the CNN-LSTM model is given to the scheduler for employing the scheduling policy of the incoming signal depending on the delay constraint and the predicted channel type. The proposed scheduling scheme outperforms the baseline schemes like PF and shows better BER performance in comparison to the baseline method while improving the reliability of the delay-sensitive signal transmission. In the future, we plan to extend the work for developing intelligent and dynamic scheduling schemes using unlabeled datasets, considering the constraints of wireless channel dynamics like fading and mobility conditions in the future.

REFERENCES

[1] A. Nessa, B. Adhikari, F. Hussain, and X. N. Fernando, "A survey of machine learning for indoor positioning," *IEEE access*, vol. 8, pp. 214 945–214 965, 2020.

[2] H. Wang, Y. Bai, and X. Xie, "Dynamic resource allocation for 5g-enabled industrial internet of things system with delay tolerance," in *2022 IEEE 96th Vehicular Technology Conference (VTC2022-Fall)*. IEEE, 2022, pp. 1–6.

[3] A. Mamane, M. Fattah, M. El Ghazi, M. El Bekkali, Y. Balboul, and S. Mazer, "Scheduling algorithms for 5g networks and beyond: Classification and survey," *IEEE Access*, vol. 10, pp. 51 643–51 661, 2022.

[4] Z. Sun, K. Wang, R. Sun, and Z. Chen, "Channel state identification in complex indoor environments with st-cnn and transfer learning," *IEEE Communications Letters*, 2022.

[5] C. Jiang, J. Shen, S. Chen, Y. Chen, D. Liu, and Y. Bo, "Uwb nlos/los classification using deep learning method," *IEEE Communications Letters*, vol. 24, no. 10, pp. 2226–2230, 2020.

[6] J. Park, S. Nam, H. Choi, Y. Ko, and Y.-B. Ko, "Improving deep learning-based uwb los/nlos identification with transfer learning: An empirical approach," *Electronics*, vol. 9, no. 10, p. 1714, 2020.

[7] T. Zeng, Y. Chang, Q. Zhang, M. Hu, and J. Li, "Cnn-based los/nlos identification in 3-d massive mimo systems," *IEEE Communications Letters*, vol. 22, no. 12, pp. 2491–2494, 2018.

[8] Z. Cui, Y. Gao, J. Hu, S. Tian, and J. Cheng, "Los/nlos identification for indoor uwb positioning based on morlet wavelet transform and convolutional neural networks," *IEEE Communications Letters*, vol. 25, no. 3, pp. 879–882, 2020.

[9] A. Kirmaz, D. S. Michalopoulos, I. Balan, and W. Gerstacker, "Los/nlos classification using scenario-dependent unsupervised machine learning," in *2021 IEEE 32nd Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*. IEEE, 2021, pp. 1134–1140.

[10] F. Carpi, L. Davoli, M. Martalò, A. Cilfone, Y. Yu, Y. Wang, and G. Ferrari, "Rssi-based methods for los/nlos channel identification in indoor scenarios," in *2019 16th International Symposium on Wireless Communication Systems (ISWCS)*. IEEE, 2019, pp. 171–175.

[11] A. Poulose and D. S. Han, "Uwb indoor localization using deep learning lstm networks," *Applied Sciences*, vol. 10, no. 18, p. 6290, 2020.

[12] C. L. Sang, B. Steinhagen, J. D. Homburg, M. Adams, M. Hesse, and U. Rückert, "Identification of nlos and multi-path conditions in uwb localization using machine learning methods," *Applied Sciences*, vol. 10, no. 11, p. 3980, 2020.