Signal Detection in NOMA Systems using DNN with Bidirectional LSTM

Bibekananda Panda, Poonam Singh
Department of Electronics and Communication Engineering,
National Institute of Technology Rourkela, Odisha, India
Email: bibekananda_panda@nitrkl.ac.in, psingh@nitrkl.ac.in

***Abstract*—Deep and machine learning-based algorithms are two new methodologies to solve time series prediction challenges. Traditional regression-based modeling has been found to provide less accurate findings than these techniques. A deep learning-assisted method for detecting signals in non-orthogonal multiple access systems with orthogonal frequency division (OFDM-NOMA) is described. The deep neural network (DNN) with a bi-directional long short-term memory (Bi-LSTM) is used to detect signals using different deep learning-based optimizers such as Sgdm, RMSprop, and Adam. The combination that detects most accurately is determined by comparing neural networks and optimizers. The simulations show that the deep learning technique can outperform the conventional successive interference cancellation (SIC) method and demonstrate that the Bi-LSTM-based deep learning algorithm may effectively detect signals in NOMA system scenarios in the Long-Short-Term Memory (LSTM) model. As a result, deep learning is a reliable and essential method for detecting NOMA signals.**

***Index Terms*—NOMA, LSTM, Bi-LSTM, Optimizers**

I. INTRODUCTION

In fifth-generation (5G) wireless networks, the NOMA schemes have been identified as a potential multiple access strategy for improving spectrum efficiency and system throughput. By multiplexing frequency resources in the code or power domain, NOMA allows concurrent access to identical frequency resources for several users. NOMA will enable users with bad channel conditions to distribute subcarriers to those with good channel conditions simultaneously, allowing bandwidth resources to be used efficiently. Eliminating inter-user interference is crucial for proper decoding since receivers in NOMA systems receive a combination of information from several users. Based on channel state information (CSI), the SIC technique decodes data from several users in descending order of signal power [1]. NOMA presents challenges due to pilot symbols interfering with other users' signals during the channel estimation. Traditional channel estimation algorithms may suffer significant performance degradation. Deep learning has recently gained much traction in the wireless communications industry. Several works have examined deep learning to improve channel estimation and signal identification methods [2].

Machine learning specifically, deep learning-based methods are developing neural network-based data analysis methodologies. This is because artificial intelligence (AI) and machine-learning-based technologies elevate data analysis to a new level, with data-driven models instead of model-driven. The optimum learning model for the underlying application domain may be trained. DNNs are more suited for modeling problems like time-series data analysis. Generally, a standard neural network cannot remember previous data. The deep learning approach feeds these models into forward-based learning methods. LSTM networks are a unique neural network model that simulates the associations between input and output data. These LSTM models, also known as feedback-based models, may learn from previous data by incorporating many gates into their network design to recall the past data and construct a future model based on the past and present data [3]- [4]. As a result, the input data is only scanned once (from the left (input) to the right (output)). This work examines the performance of bidirectional LSTMs (Bi-LSTM) to see if adding more layers of training to the design of an LSTM enhances its prediction. In a Bi-LSTM model, input data is deployed twice during the training process (forward and backward direction). Cognitive analysis of these two techniques is proposed when their models are trained. DNNs are designed to learn sequential data and perform well in various time series problems [5]. For fast time-varying NOMA systems, a DNN with a Bi-LSTM architecture with different deep learning-based optimizers is described for signal detection.

The remainder of the paper is structured as follows. The suggested system overview is described in Section II. Section III contains the simulation findings. Section IV includes the conclusion and potential future research options.

II. OVERVIEW OF THE PROPOSED SYSTEM

A. System Model

This paper explores a downlink two-user NOMA scenario in the OFDM system shown in Fig. 1. NOMA is one of the most efficient ways of improving system performance and accommodating huge quantities of connections. Compared to conventional orthogonal multiple access (OMA) systems, which can only allow one user per resource block, several power domains, frequencies, and codes are given to various users simultaneously to improve spectrum efficiency. Superposition coding enables several users to send data simultaneously with equal freedom but at distinct power levels. NOMA has been shown to support enormous connections, improving spectral efficiency and lowering communication latency. OFDM-NOMA outperforms OFDM-based OMA (OFDM-OMA) in power efficiency, spectrum efficiency, and fairness [6]. NOMA is enabled by superposition coding on the transmitter and successive interference cancellation (SIC) on the receiver.

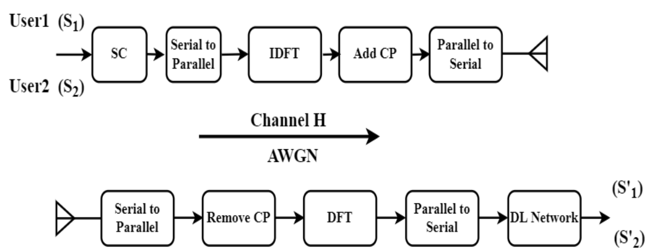

Fig. 1. System model of OFDM-NOMA

The power coefficient a_i is assigned to each user, and P_t is the total power. The total power allocation coefficient is added together into unity, i.e. $\sum_{i=1}^{L} a_i = 1$.
The transmission sequence may be expressed as [7]- [9]

$$x(n) = \sum_{i=1}^{L} \sqrt{a_i P_t} S_i(n) \tag{1}$$

Here, $S_i(n)$ is the n^{th} transmitted sample of user i. Inverse discrete Fourier transform (IDFT) transforms the superposition coding symbols from serial to parallel. Applying a cyclic prefix reduces inter-symbol interference (ISI). The maximum delay spread of the channel cannot be smaller than the length of the cyclic prefix. The signal received may be expressed as

$$y(n) = h(n) \otimes \sum_{i=1}^{L} \sqrt{a_i P_t} S_i(n) + w(n) \tag{2}$$

where $w(n)$ is the AWGN and $\otimes$ is the circular convolution.

The conventional SIC method will be applied to detect the signal and estimate the channel state information. By considering the OFDM system, the models are trained. The training data for the channel models are obtained through simulations. Each simulation generates a random data sequence as the transmitted symbols. The OFDM-based data is generated with the help of fixed pilot symbols during the deployment and training stages. The received pilot and data blocks of the OFDM system with NOMA networks are fed into the deep learning model.

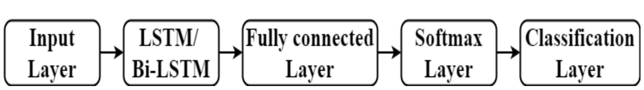

Fig. 2. Different layers of DL-network

A DNN approach is utilized to retrieve the transmitted signals with both users in a one-shot technique for the Deep learning-based NOMA receiver. The DNN is trained offline using model parameters produced with a specific channel characteristic to learn channel features implicitly. The objective of the online deployment step is to integrate the received signal with the appropriate sending symbol. The different DNN layers are illustrated in Fig. 2. These layers are the input layer, LSTM/ Bi-LSTM layer, fully connected layer, softmax layer, and classification layer. The DNN's essential layer is the LSTM/ Bi-LSTM layer, a form of RNN that can exploit data time dependencies to identify sequence and time-series data. Between time steps, these networks can learn information from sequence data while maintaining essential data. Focusing along one time-step module inside the DNN layer may be programmed to distinguish any subcarrier.

B. LSTM vs Bi-LSTM

The LSTM and Bi-LSTM layers are discussed in this section, and the separate layer concepts are described in Fig. 3 and Fig. 4, respectively.

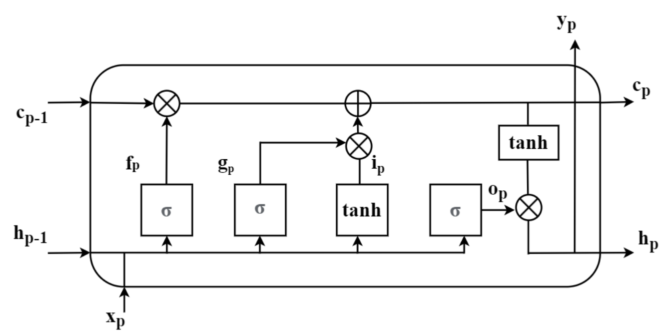

Fig. 3. The LSTM architecture

1) LSTM: The LSTM is a particular type of recurrent neural network. Its architecture was created mainly to deal with vanishing and increasing gradients. Furthermore, this network is better at preserving long-distance connections and identifying the association between values at the start and end of a series. It is organized into three gates. These are the input gate (g_p), the forget gate (f_p), and the output gate (o_p). The input gate determines whether the memory cell is updated or not. It also controls the amount of data an existing memory cell receives from a possible new memory cell. The information that the memory cell gets from the memory cell in the previous step is stored in the forget gate. The value of the next hidden state is governed by the output gate. Mathematically, the LSTM [10] block is described as

$$g_p = \sigma(\mathbb{W}_g.[h_{p-1}, x_p] + \mathcal{B}_g) \tag{3}$$

$$f_p = \sigma(\mathbb{W}_f.[h_{p-1}, x_p] + \mathcal{B}_f) \tag{4}$$

$$o_p = \sigma(\mathbb{W}_o.[h_{p-1}, x_p] + \mathcal{B}_o) \tag{5}$$

$$c_p = f_p \otimes c_{p-1} + g_p \otimes i_p \tag{6}$$

$$i_p = tanh(\mathbb{W}_i.[h_{p-1}, x_p] + \mathcal{B}_i) \tag{7}$$

The immediate input sequence, the preceding long-term state, and the previous hidden layer state are represented here as x_p, c_{p-1}, and h_{p-1}, respectively. Additionally, the weight matrices for the input gate, output gate, and forget gate are represented by $\mathbb{W}_g$, $\mathbb{W}_o$, and $\mathbb{W}_f$, respectively. $\mathcal{B}_g$, $\mathcal{B}_o$, $\mathcal{B}_f$, and $\mathcal{B}_i$ also stand for the biased parameters of the input gate, output gate, forget gate, and temporary reverse cell state, respectively.

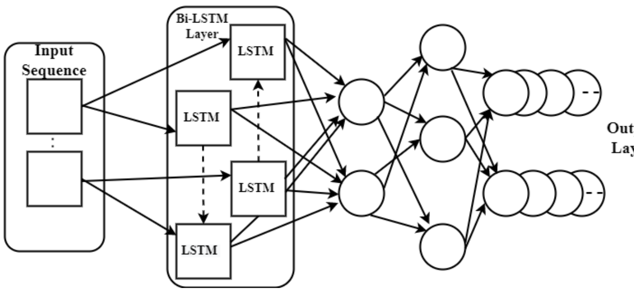

Fig. 4. The Bi-LSTM architecture

2) Bi-LSTM: The Bi-LSTM [11] expands earlier LSTM models that analyze the input data using two LSTMs. In the first round, an LSTM is given as an input sequence (i.e., the forward layer). In the second cycle, the LSTM model is fed the reverse version of the input sequence (i.e., the backward layer). When the LSTM is used twice, it increases the model's accuracy and the learning of long-term dependencies. In Fig. 4, the enhanced version of LSTM (Bi-LSTM) is depicted. In the first round, the forward input sequence $(x_1, x_2, \ldots, x_p)$ is fed into the LSTM model, and the reverse input sequence $(x_p, x_{p-1}, \ldots, x_1)$ is provided in the second round. So, for the forward and the backward LSTM, we have

$$[y_1, y_2, \ldots, y_p] = \overrightarrow{LSTM}(x_1, x_2, \ldots, x_p) \tag{8}$$

and

$$[y_p, y_{p-1}, \ldots, y_1] = \overleftarrow{LSTM}(x_p, x_{p-1}, \ldots, x_1), \tag{9}$$

where LSTM $(\cdot)$ indicates the forward trained LSTM with expected output $(y_1, y_2, \ldots, y_p)$, and the backward trained LSTM $(\cdot)$ denotes the reversed LSTM with estimated output $(y_p, y_{p-1}, \ldots, y_1)$. The use of LSTM enhances the accuracy and long-term dependence significantly.

C. Deep Learning- based optimizers overview

Any deep learning model comprises an algorithm that aims to expand the data and make predictions based on previously unknown information. We need an optimization approach and an algorithm that converts input examples to output examples. An optimization technique establishes the value of the parameters (weights) that minimizes the error while converting inputs to outputs. These optimization approaches or optimizers highly influence the accuracy of the deep learning model. While training the deep learning model, the weights of each epoch must be modified, and the loss function minimized. An optimizer is a function or algorithm that alters a neural network's weights and learning rate. As a result, it assists in lowering total loss and enhancing reliability [12]- [14].
The following is the list of some optimizers in our work on MATLAB simulations:

1) Stochastic Gradient Descent with Momentum (Sgdm): The Stochastic Gradient Descent with Momentum (Sgdm) algorithm is a version of the Gradient Descent (GD) method. The derivative is determined one point at a time using the SGDM method. SGDM-based procedures allow users to change the algorithm's parameters for substantial data sets. Faster convergence and fewer memory needs are among the advantages of this optimizer. Even after reaching global minima, model parameters with many variations might overshoot. The weight update equation is written as

$$w_{p+1} = w_p - \eta \bigtriangleup w_p \tag{10}$$

Here, w_p, w_{p+1} , and η are the previous weight (old weight), new weight, and learning rate respectively.
The weight update rule for SGDM is described as follows.

$$update_p = \gamma \cdot update_{p-1} + \eta \bigtriangleup w_p$$

$$w_{p+1} = w_p - update_p \tag{11}$$

$$\begin{aligned}
&update_0 = 0\\
&update_1 = \gamma \cdot update_0 + \eta \bigtriangleup w_1 = \eta \bigtriangleup w_1\\
&update_2 = \gamma \cdot update_1 + \eta \bigtriangleup w_2 = \eta \bigtriangleup w_2 + \gamma\eta \bigtriangleup w_1\\
&\qquad\vdots\\
&update_l = \gamma \cdot update_{l-1} + \eta \bigtriangleup w_l
\end{aligned}$$

Generally, it depends on the current direction and the fraction of the direction which is pointed previously.

2) Root Mean Squared Propagation (RMSProp): Among deep learning enthusiasts, the RMSprop is a necessary optimizer. In RMSprop, an Adagrad adaptation, the learning rate is reduced by an exponentially decaying average of squared gradients. RMSprop adjusts the learning rate automatically, and each parameter has a variable learning rate. This exponentially momentum-based value is defined as,

$$v_p = \alpha * v_{p-1} + (1 - \alpha)\left(\bigtriangleup w_p\right)^2$$

The weight update formula for RMSprop is

$$w_{p+1} = w_p - \frac{\eta}{\sqrt{v_p + \epsilon}} * \bigtriangleup w_p \tag{12}$$

3) Adaptive Moment Estimation (Adam): The Adam optimizer is a well-known gradient descent optimization technique. It is a technique for determining adaptive learning rates for each parameter. The advantages of both the RMSprop and Adadelta approaches are combined in this optimizer. It is faster and converges quickly with many variations, and also adjusts the vanishing learning rate. The exponential moving average m_p and the cumulative history same as RMSprop v_p are required to calculate the weight update for Adam.

$$m_p = \alpha_1 * m_{p-1} + (1-\alpha_1) * \triangle w_p$$
$$v_p = \alpha_2 * v_{p-1} + (1-\alpha_2) * (\triangle w_p)^2$$
$$\widehat{m_p} = \frac{m_p}{1-\alpha_1^p}$$
$$\widehat{v_p} = \frac{v_p}{1-\alpha_2^p}$$

The weight update formula for Adam is

$$w_{p+1} = w_p - \frac{\eta}{\sqrt{\widehat{v_p}+\epsilon}} * \widehat{m_p} \tag{13}$$

III. Simulation Results

The performance of the OFDM-NOMA on DNN layers with LSTM and Bi-LSTM layer learning-aided detection is investigated in this section.

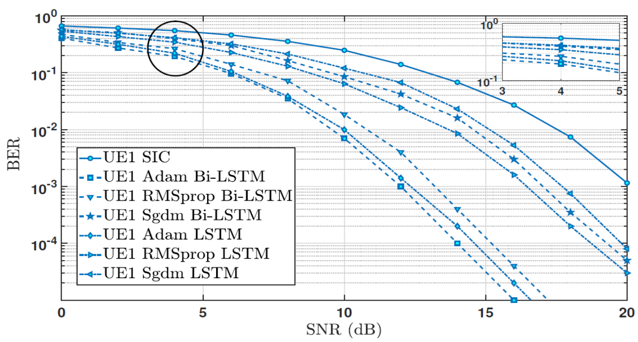

Fig. 5. BER curves for User 1 with different optimizer

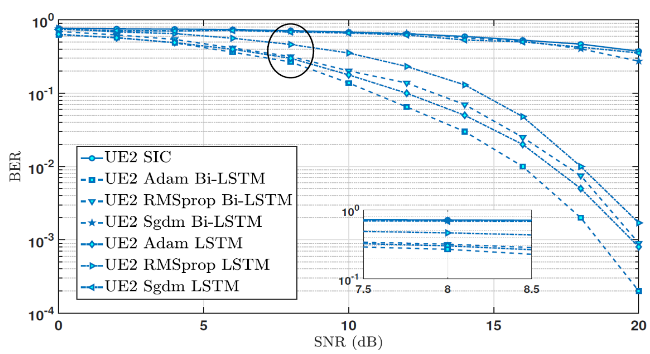

Fig. 6. BER curves for User 2 with different optimizer

For our simulation, we use various cyclic prefixes (CP = 16, 8). Table I lists the key simulation parameters. Fig. 5 and 6 demonstrate the performance of Bi-LSTM and LSTM-based OFDM-NOMA in terms of bit error rate (BER) for users 1 and 2 with cyclic prefixes of 16 and 8, respectively. The simulation graphs for these results are also compared with the conventional SIC method. A cyclic prefix is usually introduced in succeeding symbols to ignore the inter-symbol interference (ISI) and retain the orthogonality of the subcarriers. The DL receiver still performs effectively and can outperform the SIC receiver for both users under the severe impacts of ISI (CP = 8).

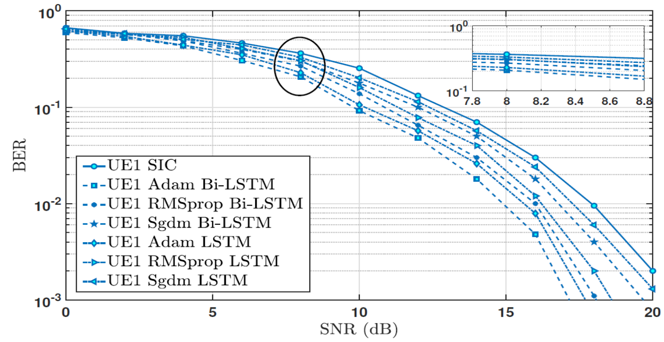

Fig. 7. BER curves for User 1 with different optimizer

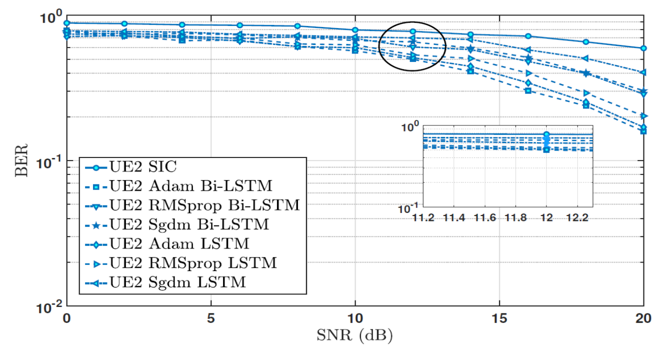

Fig. 8. BER curves for User 2 with different optimizer

Fig. 7 and 8 show the performance of the Bi-LSTM and LSTM learning-based OFDM-NOMA systems for cyclic prefix 8, respectively. Table II and III show the validation loss and accuracy of the performance of the LSTM, and Bi-LSTM layers with different optimizers for CP 16 and 8, respectively.

Adopting Bi-LSTM improves the accuracy of the system model. Table II shows the validation loss and accuracy of the performance of the LSTM and Bi-LSTM layers with different optimizers for CP 16. Adopting Bi-LSTM improves the accuracy of the system model. Similarly, the Bi-LSTM layer output gives improved accuracy in high ISI (CP=8), as seen in Table III. The Bi-LSTM models outperform standard unidirectional LSTMs, as shown by the results. By exploring input data twice (both forward and reverse direction), Bi-LSTMs retain the underlying context better. For specific types of data, such as signal processing and prediction of the

TABLE I
Simulation Parameters

Parameters	Value	Parameters	Value
No. of subcarriers	64	DNN Layer	5
Pilot length	64	Batch Size	4000
Channel Length	16	Epochs	100
Number of Users	2	Learning rate	0.01
Length of CP	16, 8	Optimizer	Sgdm, Adam, RMSprop

TABLE II
OPTIMIZER PERFORMANCE FOR CYCLIC PREFIX (CP=16)

Optimizers	Cyclic Prefix	Validation Accuracy (%)	Validation Loss (%)
Sgdm (LSTM)	16	52.97	1.32
Sgdm (Bi-LSTM)	16	60.00	0.81
RMSProp (LSTM)	16	94.92	0.24
RMSProp (Bi-LSTM)	16	96.95	0.19
Adam (LSTM)	16	97.65	0.17
Adam (Bi-LSTM)	16	99.25	0.02

symbols in the input phrase, Bi-LSTM's superior performance over ordinary unidirectional LSTM is reasonable.

TABLE III
OPTIMIZER PERFORMANCE FOR CYCLIC PREFIX (CP= 8)

Optimizers	Cyclic Prefix	Validation Accuracy (%)	Validation Loss (%)
Sgdm (LSTM)	8	51.40	1.52
Sgdm (Bi-LSTM)	8	56.74	1.01
RMSProp (LSTM)	8	93.60	0.27
RMSProp (Bi-LSTM)	8	95.97	0.24
Adam (LSTM)	8	96.22	0.20
Adam (Bi-LSTM)	8	98.35	0.08

IV. CONCLUSION

This paper compares and evaluates the accuracy and performance of unidirectional and bidirectional LSTM models with different deep learning-based optimizers. By comparing the training data collected from the right to left (i.e., opposite direction) and left to right (i.e., standard data training), it has been determined if it can positively and substantially impact signal detection accuracy in the OFDM-NOMA system. As per the simulation results, using an additional training layer can improve the system performance accuracy, which is suitable for modeling. It has observed an exciting aspect when investigating the behavior of unidirectional LSTM and Bi-LSTM models. The BER performance of DL-based Bi-LSTM is much superior to that of LSTM. To achieve equilibrium, it has been observed that training using Bi-LSTM is slower and needs accessing larger batches of data. This result shows that the unidirectional LSTM model cannot provide additional data characteristics since training is just one way. Still, the bidirectional LSTM model may be able to provide some further data attributes. Thus, the paper discusses Bi-LSTM instead of LSTM as a signal detection technique for OFDM-NOMA. Furthermore, different DL-based optimizers were used to evaluate performance for various cyclic prefix sequences. A potentially promising area for future research would be to extend this work to a heterogeneous NOMA network with more detailed experimentation and precise data analysis.

REFERENCES

[1] Z. Ding *et al.*, "Application of Non-Orthogonal Multiple Access in LTE and 5G Networks," *IEEE Communications Magazine*, vol. 55, no. 2, pp. 185-191, 2017.

[2] H. Ye, G. Y. Li and B. H. Juang, "Power of Deep Learning for Channel Estimation and Signal Detection in OFDM Systems," *IEEE Wireless Communications Letters*, vol. 7, no. 1, pp. 114-117, 2018.

[3] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning,"*Nature*, vol. 521, no. 7553, pp. 436-444, 2015.

[4] Y. Liu, Z. Ding, M. Elkashlan, and H. V. Poor, "Cooperative Non-Orthogonal Multiple Access with Simultaneous Wireless Information and Power Transfer," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 4, pp. 938-953, April 2016.

[5] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735-1780, 1997.

[6] J. Li, J. Hou, L. Fan, Y. Yan, X. -Q. Jiang and H. Hai, "NOMA-aided generalized pre-coded quadrature spatial modulation for downlink communication systems," *China Communications*, vol. 17, no. 11, pp. 120-130, 2020.

[7] Narengerile and J. Thompson, "Deep Learning for Signal Detection in Non-Orthogonal Multiple Access Wireless Systems," *UK/ China Emerging Technologies (UCET)*, pp. 1-4, 2019.

[8] A. Emir, F. Kara, H. Kaya, X. Li, "Deep learning-based flexible joint channel estimation and signal detection of multi-user OFDM-NOMA," *Physical Communication*, Volume 48, 2021.

[9] B. Panda and P. Singh, "Performance Analysis of NOMA Systems in Rayleigh and Rician Fading Channels," *in Proc. IEEE Advanced Communication Technologies and Signal Processing (ACTS)*, pp. 1-6, Dec. 2021.

[10] J. Sun, W. Shi, Z. Yang, J. Yang, and G. Gui, "Behavioral Modeling and Linearization of Wideband RF Power Amplifiers Using BiLSTM Networks for 5G Wireless Systems," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 11, pp. 10348-10356, 2019.

[11] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673-2681, 1997.

[12] D. Soydaner, "A Comparison of Optimization Algorithms for Deep Learning," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 34, 2020.

[13] R. Yu Sun, "Optimization for Deep Learning: An Overview," *Journal of the Operations Research Society of China*, pp. 249–294, 2020.

[14] A. Shrestha and A. Mahmood, "Review of Deep Learning Algorithms and Architectures," *IEEE Access*, vol. 7, pp. 53040-53065, 2019.