

Article

Interference Management for a Wireless Communication Network Using a Recurrent Neural Network Approach

Mohammad Abrar Shakil Sejan ^{1,2}, Md Habibur Rahman ^{1,2}, Md Abdul Aziz ^{1,2}, Rana Tabassum ^{1,2},
Young-Hwan You ^{2,3}, Duck-Dong Hwang ⁴ and Hyoung-Kyu Song ^{1,2,*}

¹ Department of Information and Communication Engineering, Sejong University, Seoul 05006, Republic of Korea; sejan@sejong.ac.kr (M.A.S.S.); habibur@sju.ac.kr (M.H.R.); aziz@sju.ac.kr (M.A.A.); tanvi@sju.ac.kr (R.T.)

² Department of Convergence Engineering for Intelligent Drone, Sejong University, Seoul 05006, Republic of Korea; yhyou@sejong.ac.kr

³ Department Computer Engineering, Sejong University, Seoul 05006, Republic of Korea

⁴ Department of Electronics and Communication Engineering, Sejong University, Seoul 05006, Republic of Korea; duckdonh@sejong.ac.kr

* Correspondence: songhk@sejong.ac.kr

Abstract: Wireless communication technologies have profoundly impacted the interconnectivity of mobile users and terminals. Nevertheless, the exponential increase in the number of users poses significant challenges, particularly in interference management, which is a major concern in wireless communication. Machine learning (ML) approaches have emerged as powerful tools for solving various problems in this domain. However, existing studies have not fully addressed the problem of interference management for wireless communication using ML techniques. In this paper, we explore the application of recurrent neural network (RNN) approaches to address co-channel interference in wireless communication. Specifically, we investigate the effectiveness of long short-term memory (LSTM), bidirectional LSTM (Bi-LSTM), and gated recurrent unit (GRU) network architectures in two different network settings. The first network comprises 10 connected devices, while the second network involves 20 devices. Our experimental results demonstrate that Bi-LSTM outperforms LSTM and GRU in terms of mean squared error, normalized mean squared error, and sum rate. While LSTM and GRU produce similar results, LSTM exhibits a marginal advantage over GRU. In addition, a combined RNN approach is also studied, and it can provide better results in dense networks.

Keywords: interference management; wireless network; deep learning; recurrent neural network

MSC: 94A13



Citation: Sejan, M.A.S.; Rahman, M.H.; Aziz, M.A.; Tabassum, R.; You, Y.-H.; Hwang, D.-D.; Song, H.-K. Interference Management for a Wireless Communication Network Using a Recurrent Neural Network Approach. *Mathematics* **2024**, *12*, 1755. <https://doi.org/10.3390/math12111755>

Academic Editors: Farag M. Sallabi and Mohammad Hayajneh

Received: 13 May 2024

Revised: 28 May 2024

Accepted: 4 June 2024

Published: 5 June 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Present-day wireless communication is the cornerstone of our interconnected world, facilitating seamless connectivity and information exchange across a multitude of devices and applications. From smartphones and tablets to Internet of Things (IoT) sensors and autonomous vehicles, the proliferation of wireless technologies has revolutionized how we communicate, work, and interact with our surroundings [1]. The fifth generation (5G) communication technology allows wireless communication by employing massive multiple-input-multiple-output (MIMO), ultra-dense network, and device-to-device communication [2]. However, amidst this exponential growth in wireless connectivity and high-density data connectivity, the efficient management of interference within wireless networks has emerged as a paramount challenge, dictating the quality and reliability of wireless experiences. Interference, in the context of wireless communication, refers to the phenomenon in which signals from multiple devices or networks overlap and disrupt each other, leading to signal degradation, reduced throughput, and impaired performance [3,4]. This interference can arise from various sources, including co-channel interference, adjacent

channel interference, and inter-symbol interference, each presenting unique hurdles to overcome in the quest for seamless wireless connectivity [4]. The escalating demand for wireless services, coupled with the finite nature of the radio spectrum, underscores the urgency of devising effective strategies for interference management in wireless networks. With the advent of technologies like 5G and beyond, which promise unprecedented data rates, ultra-low latency, and massive connectivity, the importance of mitigating interference becomes even more pronounced [5]. Thus, next-generation networks introduce new complexities, such as denser deployments of devices, heterogeneous architectures for networks, and diverse quality-of-service requirements, further accentuating the need for robust interference management solutions.

The solution to interference in wireless networks necessitates a multidisciplinary approach, drawing upon insights from signal processing, information theory, network optimization, and regulatory frameworks [6]. Advanced signal processing techniques, including adaptive filtering, beamforming, and interference cancellation, play a crucial role in mitigating interference and improving spectral efficiency [7–9]. Resource allocation algorithms, such as dynamic spectrum access and cognitive radio, enable the efficient utilization of available spectrum resources, minimizing interference while maximizing network capacity [10]. Recent studies in the literature have tried to solve the interference problem. The authors of [11] used graph representation learning for interference management in wireless networks. An actor–critic graph representation learning algorithm was developed to train neural networks to construct the optimal graph representing the impact of interference in a wireless network. The study reported in [12] employed a rate-splitting multiple-access precoding method in a multiple-antenna interference channel using deep reinforcement learning. The authors of [13] presented a study on satellite–ground integrated network resource interference management by analyzing coverage overlap and beneficially scheduling both beam-domain and power-domain resources. A deep Q network was designed using LSTM to predict the channel state information of satellite links. Reconfigurable intelligent surface-aided proactive mobile network interference mitigation was proposed in [14]. This study focused on developing an advantage actor–critic-based method for the interference problem by designing action space, state space, and a reward function. To mitigate RF channel interference, a deep learning approach was proposed in [15]. The authors modified the VGGNet-16 by adding a new convolution layer with 1280 filters, which enhanced the performance of the wireless communication system. Moreover, spectrum management policies and regulatory frameworks play a pivotal role in shaping the landscape of interference management, balancing the interests of various stakeholders, and ensuring fair and equitable access to the radio spectrum. As wireless networks continue to evolve and diversify, driven by emerging technologies like IoT, edge computing, and machine learning (ML), the quest for effective interference management remains a dynamic and ongoing endeavor.

In recent years, ML has been very successful in solving many complex problems across various domains. ML can be applied efficiently in various tasks, including computer vision, image interpretation, text classification, human behavior and identity recognition, fraud detection, recommendation systems, drug discovery, visual art generation, and natural language processing [16–23]. In addition, ML has significantly impacted wireless communication, revolutionizing various aspects of network operation, optimization, and management [24–26]. In particular, ML algorithms based on deep learning offer powerful tools for extracting valuable insights from the vast amount of data generated by wireless networks. Deep learning algorithms demonstrably outperform shallow machine learning algorithms in various ways. Key advantages include feature learning, superior performance on complex tasks like image and speech recognition, natural language processing, and time series forecasting, the ability to handle increasingly large and intricate datasets with complex relationships, and the facilitation of end-to-end learning, transfer learning, and the efficient handling of sequential data [27]. By analyzing historical usage patterns and real-time network conditions, ML algorithms can dynamically allocate spectrum re-

sources, optimize transmission parameters, and mitigate interference, thereby enhancing spectral efficiency and network capacity [28,29]. Moreover, ML techniques enable intelligent predictive maintenance and fault detection, allowing network operators to proactively identify and address potential issues before they affect service quality [30]. Additionally, ML-based approaches facilitate the design of adaptive and self-optimizing networks that can autonomously reconfigure themselves to adapt to changing environmental conditions and user requirements. ML offers an efficient and autonomous approach to determine the nonlinear mapping relationship between fatigue life and numerous variables, leveraging available experimental data [31]. Overall, the integration of machine learning techniques into wireless communication holds immense promise for improving network performance, reliability, and user experience [32]. A class of neural networks known as recurrent neural networks (RNNs) was created to process sequential input efficiently by preserving memory over time steps [33]. Unlike traditional feed-forward neural networks, RNNs possess feedback connections that allow information to persist and flow through the network's hidden states. This inherent memory capability makes RNNs particularly well-suited for tasks involving sequential data, such as natural language processing, time series analysis, and speech recognition [34]. The architecture of an RNN consists of recurrent connections between neurons, enabling the network to capture dependencies and temporal patterns within sequences. However, traditional RNNs suffer from the vanishing gradient problem, limiting their ability to capture long-term dependencies. To solve this problem, various extensions of RNNs have been proposed, including long short-term memory (LSTM), bidirectional LSTM, and gated recurrent unit (GRU) networks. These variants incorporate sophisticated gating mechanisms that regulate the flow of information through the network, allowing them to effectively capture and propagate information over long sequences. RNNs have demonstrated remarkable success in a wide range of applications, including language modeling, time series prediction, machine translation, and sentiment analysis [35,36]. Their ability to model sequential data dynamics and capture contextual information makes them indispensable tools in the field of deep learning.

ML techniques have already been widely adopted in previous studies to address interference-related problems in wireless communication. For instance, the authors of [37] applied a convolutional neural network (CNN) and LSTM to form a convolutional LSTM autoencoder to retrieve corrupted data. The proposed model demonstrated improved symbol error rate performance in interference-affected signals after neural network training. Similarly, [38] utilized field-programmable gate arrays (FPGAs) as wireless receivers and employed a neural-network-based model for data demodulation. In [39], a CNN-based interference reduction method was introduced to classify received signals, achieving a high true positive rate with the deployment of two convolutional layers. Two convolutional layers were deployed to classify the signal label with a good true positive rate. Moreover, promising results were reported in [40] regarding CNN-based techniques for denoising and interference mitigation in radar processing. The study experimented with three models, one with three layers and the other two with a bottleneck-based structure, which reduces the number of channels. To mitigate interference in radar signals, another CNN approach was proposed in [41], in which a single convolutional layer and two batch normalization layers were utilized to create the model. A single convolutional layer and two batch normalization layers were used to create the model. Additionally, RNN-based approaches have been explored in previous research. In Ref. [42], the authors applied a multi-layer GRU-based model, demonstrating high performance under various interference conditions with low processing time. Moreover, Ref. [43] utilized a derivative-free Kalman filter-based RNN to examine interference reduction methods for the global positioning system (GPS). An adaptive unscented Kalman filter-based RNN (UKF-RNN) was used to lessen pulsed, swept, continuous wave interference (CWI), and multi-tone CWI. Furthermore, Ref. [42] suggested an RNN with self-attention to reduce radar interference caused by frequency-modulated continuous waveforms (FMCWs) and orthogonal frequency-division multiplexing (OFDM). Previous studies in the literature focused on diverse problems in

wireless communication, and only a few studies considered the mitigation of interference for wireless communication. Additionally, the results produced by previous studies can be improved by applying new machine learning approaches. RNNs can be a promising candidate for examining and reducing interference in the channel. There is a research gap in considering the performance of the RNN techniques in interference management, and this gap needs further research.

In this paper, we model a co-channel interference network and simulate a real-time environment for data extraction. We utilize RNN machine learning approaches to measure the performance of the channel under interference. Each of the techniques, LSTM, BiLSTM, GRU, and a combined approach are employed to enhance the performance of interference-based communication.

The remainder of the paper is organized as follows: Section 2 details the wireless network interference structure, Section 3 describes RNN approaches, Section 4 outlines the research results and discussion, and Section 5 provides concluding remarks.

2. Wireless Network Interference

In any network system, the communication system is susceptible to signal or noise interference. Figure 1 depicts a typical wireless communication scenario. As shown in the figure, region A contains numerous users, while region B has fewer users. This discrepancy suggests the potential for interference in region A, which could diminish the user experience.

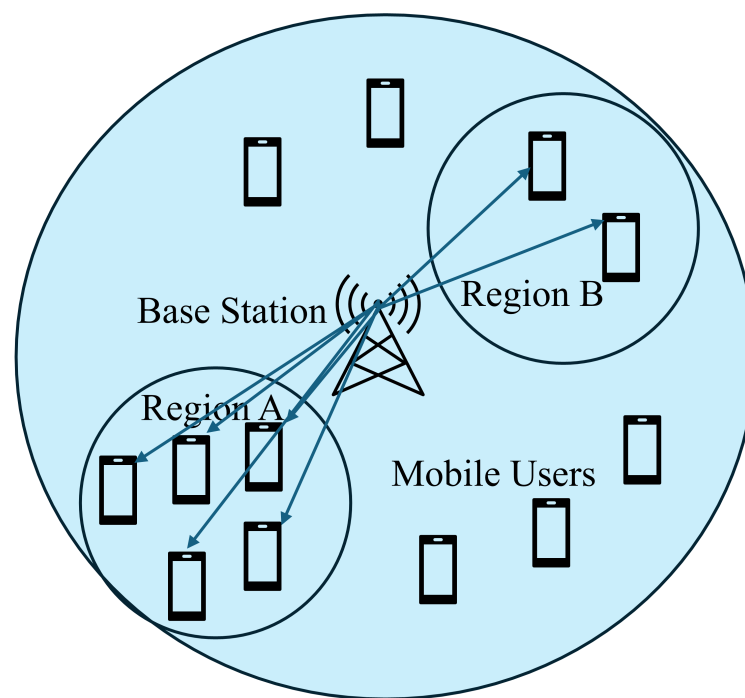


Figure 1. Interference example in a typical network between the base station and mobile users, where interference is prominent in region A and less interference in region B.

To solve the interference problem, we can consider the following optimization problem:

$$\begin{aligned} & \text{minimize} && p(e; z), \\ & \text{subject to} && e \in E \end{aligned} \quad (1)$$

where $e : \mathbb{R}^n \rightarrow \mathbb{R}$ is a continuous nonconvex objective function, e is the problem parameter, $E \in \mathbb{R}^n$ is the feasible region, and $z \in \mathbb{R}^n$ is a vector of the problem parameter. The original signal, which is received by the receiver, is added with other signals that cause signal interference. The communication links between different users and the base station

cause interference. The throughput expression using Shannon's formula can be expressed as follows:

$$T_i = B \log_2(1 + SINR_i) \quad (2)$$

where T_i is the throughput for the i -th pair of communication, B is the bandwidth of the channel, and $SINR_i$ is the signal-to-interference-plus-noise ratio. Again, the $SINR_i$ can be expressed as follows [44]:

$$SINR_i = \frac{|h_{ii}|^2 p_i}{\sum_{i \neq j} |h_{ij}|^2 p_j + \delta_i^2}, \quad (3)$$

where $h_{ii} \in \mathbb{C}$ is the direct channel between transmitter i and receiver i , $h_{ij} \in \mathbb{C}$ is the interference channel from transmitter i to receiver j , p_i is the power of the transmitter, p_j is the other transmitter power in the network, and δ_i^2 is the noise power at the receiver. Our goal is to retrieve the original signal from the corrupted signal by interference and solve Equation (1).

3. Method with RNN for Interference Management

3.1. Dataset Generation

We consider a network with N devices, each connected to a base station. The total signal received by a device is the sum of the direct signal and the interference signal. Initially, we calculate the direct signal using the diagonal parts of the channel matrix. Subsequently, the interference signal is determined by computing the cross channel, excluding the diagonal parts of the channel matrix. In the training data, we apply the interference signal to reveal the original signal; in turn, this can minimize interference. To simulate interference, we conducted experiments using two different network sizes. The first network comprised 10 devices operating simultaneously, while the second network involved 20 devices operating concurrently. We opted for these network sizes to mitigate network complexity. Initially, a simple RNN structure was employed, which can later be scaled up for a larger number of devices. Each device generated 20,000 samples, with a noise variance assumed to be 1 for this experiment. Additionally, each device generated 20 different feature values, which served as input for the RNN algorithm. Furthermore, each device had a unique label for classification purposes. The input data dimensions for the training sets were 20,000 samples $\times$ 10 features $\times$ 20 dimensions per feature (for the 10-device network) and 20,000 samples $\times$ 20 features $\times$ 20 dimensions per feature (for the 20-device network). To evaluate the models, we generated separate test sets of 2000 samples each, maintaining the same network configurations (10 or 20 features, 20 dimensions per feature) as the training data. The parameters for the considered network are listed in Table 1. After the data are received through the channel, they are utilized for training purposes. Algorithm 1 outlines the process of data generation for training and testing.

Table 1. Parameter list for data generation.

Parameter	Value
Number of devices	20, 10
Training samples	20,000
Noise variance	1
Feature of each device	20
Label for each device	1

Algorithm 1 Training Data Generation Process

- 1: Initialize data: number of users N , number of features q , number of labels for each user s , total number of data generation t .
- 2: Calculate h_{ii} and h_{ij} for each user.
- 3: Employing simulated channel the training data is passed. $y = xh_{ii} + xh_{ij} + n$.
- 4: The received data y is saved for training.
- 5: Separate data is generated for testing process.

3.2. Details of RNN Models

In this section, we describe the RNN models considered for inference reduction. After generating the training data, we employ the models for training.

3.2.1. LSTM

LSTM networks incorporate specialized memory cells with gating mechanisms, allowing them to selectively store, update, and retrieve information over extended sequences. This unique architecture enables LSTM networks to overcome the limitations of standard RNNs by facilitating the learning of complex temporal patterns and dependencies. The key components of an LSTM unit include the input gate, forget gate, output gate, and cell state, which are each responsible for controlling the flow of information and preserving relevant information over multiple time steps. By effectively managing the flow of information through the network, LSTM networks can capture intricate dependencies and context within sequential data, making them powerful tools for tasks requiring long-range temporal modeling and prediction. The gates of LSTM can be expressed as follows:

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \quad (4)$$

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \quad (5)$$

$$g_t = \tanh(W_{xg}x_t + W_{hg}h_{t-1} + b_g) \quad (6)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot g_t \quad (7)$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \quad (8)$$

$$h_t = o_t \odot \tanh(c_t), \quad (9)$$

where x_t is the input at time stop t , h_{t-1} is the hidden state from the previous time step, c_{t-1} is the cell state from the previous time step, i_t is the input gate vector, f_t is the forget gate vector, g_t is the update gate vector, o_t is the output gate vector, σ is the sigmoid activation function, $\odot$ represents element-wise multiplication, W is the weight matrix, and b is bias vector. The architecture of the LSTM-based machine learning model is depicted in Figure 2. The LSTM layer is connected to the input layer, after which the data pass through a dropout layer and a fully connected layer. Finally, a classification layer is utilized to predict the correct class of the input data. This architecture allows the model to effectively process sequential data, extract relevant features, and make accurate predictions.

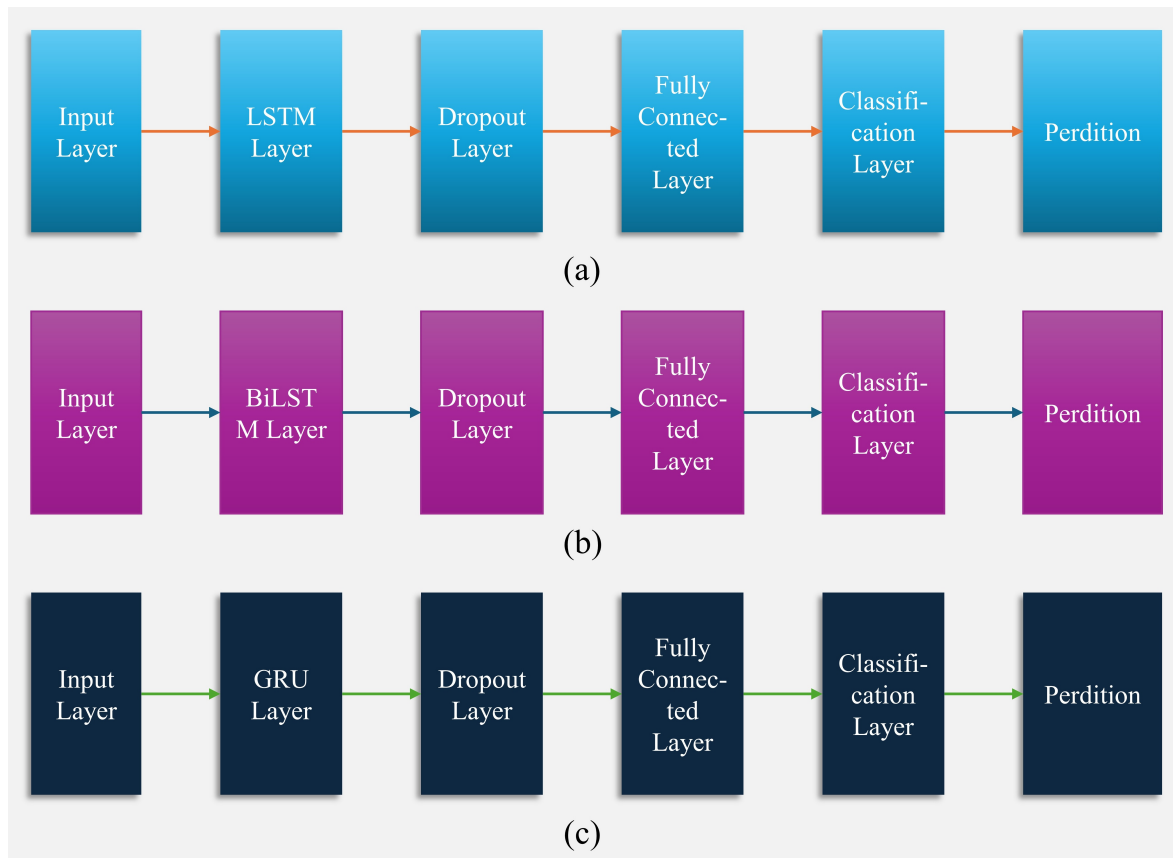


Figure 2. Model architecture for training data; (a) LSTM ML model architecture for interference reduction; (b) the BiLSTM ML model architecture for interference reduction; (c) the GRU ML model architecture for interference reduction.

3.2.2. BiLSTM

BiLSTM networks are an extension of the traditional LSTM architecture, designed to capture dependencies in both the forward and backward directions within sequential data. Unlike standard LSTMs, which process input sequences sequentially from past to future, BiLSTMs incorporate two separate LSTM layers: one processing the input sequence in the forward direction and the other processing it in the backward direction. This bidirectional processing allows the network to capture contextual information from both past and future contexts, enabling a more comprehensive understanding of the input sequence. In a BiLSTM, the output of each LSTM layer is concatenated at each time step, effectively merging information from both directions. This concatenated representation contains information about the input sequence as a whole, incorporating both past and future contexts. BiLSTMs are particularly effective in tasks where context from both directions is crucial, such as part-of-speech tagging, named entity recognition, and sentiment analysis. The bidirectional nature of BiLSTMs makes them powerful tools for capturing long-range dependencies and contextual information, leading to improved performance in various sequence modeling tasks. However, it is worth noting that BiLSTMs may introduce additional computational complexity due to the need to process input sequences in both directions. Nonetheless, their ability to leverage bidirectional context makes them a popular choice in many natural language processing and sequence prediction applications. The mathematical expression of BiLSTM can be expressed as follows [45]:

$$\vec{h}_f = \sigma(W_f S_t + W_f h_{t-1} + b_f), \quad (10)$$

$$\overleftarrow{h}_r = \sigma(W_r S_t + W_r h_{t+1} + b_r), \quad (11)$$

where σ is the activation function, the time steps of forward and backward represent $t - 1$ and $t + 1$, S_t is the transmitted signal, the hidden state of previous and the next are h_{t-1} and h_{t+1} , the weights and learnable bias of both directions are Wf and Wr and bf and br , and $hf \rightarrow$ and $h3r \leftarrow$ are the forward- and backward-direction LSTM network outputs, respectively. The model architecture for BiLSTM to tackle interference is illustrated in Figure 2b. The input data are fed into the BiLSTM layer, followed by a dropout layer. Subsequently, a fully connected layer and a classification layer are employed to classify the input data. This architecture enables the model to capture both past and future context information, thus enhancing its ability to mitigate interference effectively.

3.2.3. GRU

GRU is a type of RNN architecture that addresses some limitations of traditional RNNs, such as the vanishing gradient problem and the inability to capture long-term dependencies effectively. GRUs are designed to selectively update and reset their internal state, allowing them to retain relevant information over longer sequences while avoiding unnecessary memory consumption. The GRU architecture consists of several key components, including the update gate and the reset gate. These gates regulate the flow of information through the network, determining which information to retain and which to discard at each time step. GRUs also feature a candidate state computation step, which generates a new candidate state based on the current input and the reset gate. This candidate state is then combined with the update gate's output to produce the final hidden state for the current time step. One of the key advantages of GRUs over traditional RNNs is their ability to capture long-term dependencies more effectively while requiring fewer parameters. This makes GRUs particularly well suited for tasks involving sequential data, such as language modeling, machine translation, and time series prediction. Additionally, GRUs are generally faster to train than LSTMs due to their simpler architecture, making them a popular choice in many applications requiring recurrent neural networks. The GRU computation procedure is explained using the formulas as follows [46]:

$$U_t = \sigma(W_U X_t + V_U H_{t-1} + b_U) \quad (12)$$

$$R_t = \sigma(W_R X_t + V_R H_{t-1} + b_R) \quad (13)$$

$$\hat{H}_t = \tanh(W_H X_t + V_H (R_t \otimes H_{t-1}) + b_H) \quad (14)$$

$$H_t = (1 - U_t) \otimes H_{t-1} + U_t \otimes \hat{H}_t, \quad (15)$$

where the biases are b_U , b_R , and b_H . The gate activation function is calculated using the sigmoid function as $\sigma(c) = (1 + e^{-c})^{-1}$. Again, W_U , W_R , W_H , V_U , V_R , and V_H are weight matrices. While the input state X_t and the output of the hidden layer at the previous instant are merged to generate $\hat{H}_t$, the output of the hidden layer at the current instant is represented by H_t . The hyperbolic tangent function known as ($\tanh$) is in charge of calculating the state activation function. As depicted in Figure 2c, the GRU-based machine learning model architecture is connected to the input data. The same architecture is followed as previously used in LSTM and BiLSTM for the sake of the best comparison. This consistency allows for a fair evaluation of the performance of different RNN approaches in interference management.

4. Experiment Result and Discussion

4.1. Training and Testing of the Models

We have selected two different network sizes to test various RNN approaches. The first network comprises 20 active devices, while the second network has 10 active devices. The parameter list for machine learning is provided in Table 2. To minimize model complexity, we opted for a single hidden layer in each RNN architecture. To determine the optimal number of hidden layers for each approach, we started with 10 hidden units and gradually increased this number to 100. However, for hidden units below 50, we

observed significant fluctuations in validation accuracy and loss, leading to unstable results. Conversely, when the number of hidden units exceeded 50, validation accuracy began to decrease. Consequently, 50 hidden units were chosen as they provided stable results along with high validation accuracy. Additionally, we only considered one optimizer and one loss function for the experiment. To describe the internal structure and complexity of LSTM, BiLSTM, and GRU, we can refer to their formulations. The complexity of LSTM is expressed as $(t_s \times t_h \times (4t_i + 4t_h + 3))$ [46,47], where t_i represents the number of features in the input vector, t_s denotes the size of the input time sequence, and t_h indicates the number of hidden units. For BiLSTM, the complexity is represented as $(t_s \times t_h \times 2(4t_i + 4t_h + 3))$. Similarly, the complexity of GRU is expressed as $(t_s \times t_h \times (3t_i + 3t_h + 3))$.

Table 2. Parameter list for machine learning.

Parameter	Value
RNN approach	LSTM, BiLSTM, GNN, Combined
Layers	Single layer
Hidden units LSTM	50
Hidden units BiLSTM	50
Hidden units GRU	50
Training epochs	150
Learning rate	0.001
Number of iterations	200
Learning rate decay	None
Optimizer	Adam
Loss function	Mean squared error

Figure 3 illustrates the training and validation accuracy, as well as loss, for different RNN approaches. Specifically, Figure 3a displays the performance of LSTM for the 10-device network. The training accuracy for the LSTM algorithm reaches 81.74%, with a validation accuracy of 80.28%. Additionally, the training loss decreases to 0.089, and the validation loss reaches 0.1041 after 150 episodes. The proposed LSTM network's training is completed in 150 episodes, after which the model's learning accuracy converges and remains stable. Therefore, the training episodes were kept at 150 for all algorithms to ensure a fair comparison. Figure 3b depicts the training accuracy and loss for BiLSTM in the 10-device network. The training accuracy for BiLSTM reaches 86.12%, and the validation accuracy reaches 82.62%. In contrast, the training loss reduces to 0.053, and the validation loss decreases to 0.1040. This indicates that BiLSTM has a higher performance in terms of both accuracy and loss reduction compared to LSTM. BiLSTM demonstrates slightly better performance in training accuracy compared to LSTM in terms of validation accuracy. In the case of GRU, the training accuracy and loss are presented in Figure 3c. GRU has shown a training accuracy of 81.31% and a validation accuracy of 79.79% after 150 episodes. The training loss for GRU is 0.093, and the validation loss is 0.1051, indicating a lower performance compared to BiLSTM and LSTM. In the subsequent phase, a 20-device network was considered for training. Figure 3d illustrates the training progress of LSTM, where both the accuracy and loss converge after 20 episodes. The training accuracy for the 20-device network is 79.35%, and the validation accuracy is 78.14% after 150 episodes. The training loss for the LSTM in the 20-device network is 0.087, and the validation loss is 0.096. Similarly, in the case of BiLSTM, as shown in Figure 3e, the training accuracy reaches 81.18% and the validation accuracy is 78.48% after 150 episodes. The training loss for BiLSTM decreases to 0.0723, and the validation loss is 0.0958 at the end of 150 episodes. For GRU, the training accuracy is 78.79% and the validation accuracy is 77.99%, as depicted in Figure 3f. Additionally, the training loss for GRU is 0.091 and the validation loss is 0.0966 at the end of training.

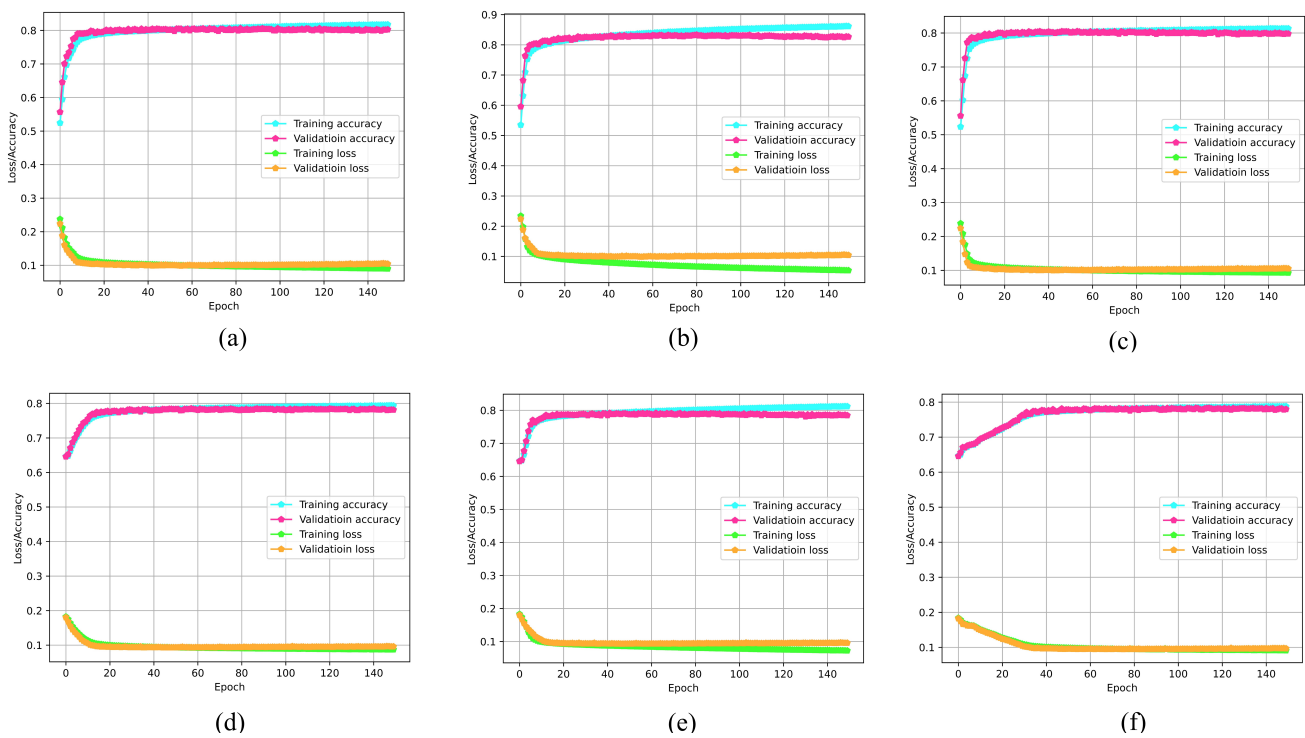


Figure 3. Accuracy and loss curve for different RNN techniques during model training; (a) LSTM training accuracy and loss with 10-device network; (b) BiLSTM training accuracy and loss with 10-device network; (c) GRU training accuracy and loss with 10-device network; (d) LSTM training accuracy and loss with 20-device network; (e) BiLSTM training accuracy and loss with a 20-device network; and (f) GRU training accuracy and loss with 20-device network.

4.2. Model Performance in Wireless Network

After passing the data through a noisy channel, we recovered the data using the RNN approach. To evaluate the quality of the received signal, we calculated the mean squared error (MSE) and normalized mean squared error (NMSE) in the initial phase. Subsequently, in the next phase, we calculated the sum rate for the received signal to assess the performance of each of the RNN algorithms. Initially, we analyzed the results of the mean squared error for the three RNN approaches across two different network sizes.

Figure 4 displays the MSE for BiLSTM, LSTM, and GRU against the episode number during training for the 10-device network. The MSE for LSTM decreases as the number of episodes increases, with a mean and standard deviation of 0.00522 ± 0.0007 . GRU shows a similar trend to LSTM with a slightly higher MSE, with a mean and standard deviation of 0.00534 ± 0.00063 . BiLSTM exhibits better performance compared to both LSTM and GRU, with a mean and standard deviation for MSE of 0.0043 ± 0.00067 . BiLSTM outperforms LSTM and GRU, having the lowest mean value among the three and showing a consistent downward curve in MSE, as illustrated in Figure 4.

Figure 5 illustrates the MSE performance for the 20-device network. Here, all approaches exhibit a decreasing trend in MSE as the number of episodes increases. LSTM achieves a mean and standard deviation of 0.00488 ± 0.0062 , with a slight upward trend towards the end of the episodes. GRU demonstrates a similar trend to LSTM, with a slightly larger mean MSE of 0.00523 ± 0.000989 . According to the results presented in Figure 5, the MSE mean and standard deviation for BiLSTM are 0.00469 ± 0.000563 . For the 20-device network, the MSE trend is similar for all three approaches, but BiLSTM exhibits comparatively better results than LSTM and GRU, as shown in Figure 5. Although the difference in MSE values is small, it can have a performance impact in delay-sensitive networks.

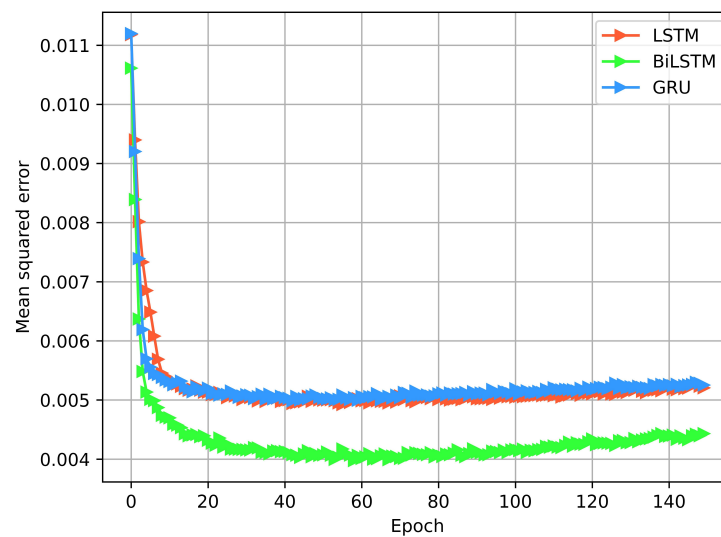


Figure 4. Mean squared error for a 10-device network.

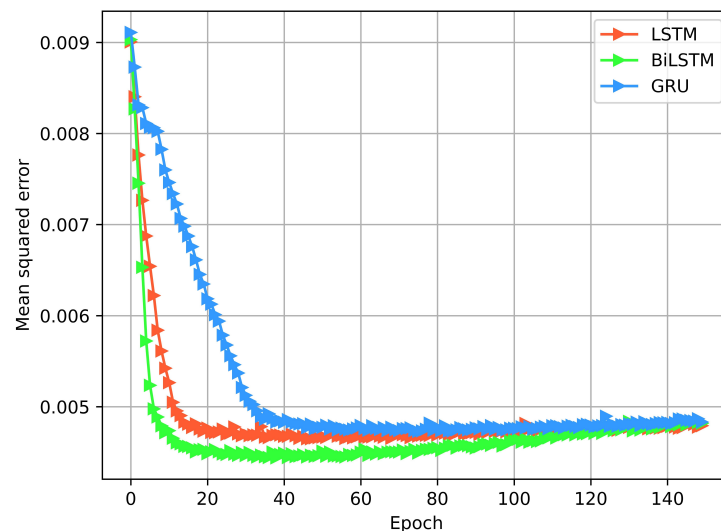


Figure 5. Mean squared error for 20-device network.

Additionally, we calculated the NMSE for both networks. The NMSE is calculated as follows:

$$NMSE = \frac{(\sum_{i=1}^M y_{estimated}(i) - y_{desired}(i))^2}{\sum_{i=1}^M y_{desired}(i)}, \quad (16)$$

where $y_{estimated}(i)$ is the i -th estimation from the ML model and $y_{desired}(i)$ is the ground truth.

The corresponding results are depicted in Figure 6 for the 10-device network. In the case of NMSE, LSTM and GRU show similar results, as depicted in Figure 6. The mean and standard deviation for the NMSE of LSTM are 0.278 ± 0.037 . For GRU, the mean and standard deviation are 0.279 ± 0.033 , which are similar to those of LSTM. However, BiLSTM exhibits better performance than LSTM and GRU, as indicated by its curve being closer to the lowest NMSE value. For BiLSTM, the mean and standard deviation are 0.231 ± 0.036 . Thus, for the 10-device network, BiLSTM demonstrates the best performance among the three approaches, as shown in Figure 6.

Figure 7 shows the NMSE results for the 20-device network. In this case, the mean and standard deviation for LSTM are 0.4156 ± 0.52 and for GRU are 0.4452 ± 0.0828 . BiLSTM again shows better performance, with a mean and standard deviation of 0.4010 ± 0.04673 .

For the 20-device network, BiLSTM shows overall better results compared to GRU and LSTM, as shown in Figure 7. While LSTM and GRU exhibit similar results, LSTM has slightly higher performance. In both the 10-device and 20-device network sizes, the NMSE exhibits a trend similar to that of MSE, reinforcing the superior performance of BiLSTM in reducing the error in interference-prone wireless communication channels.

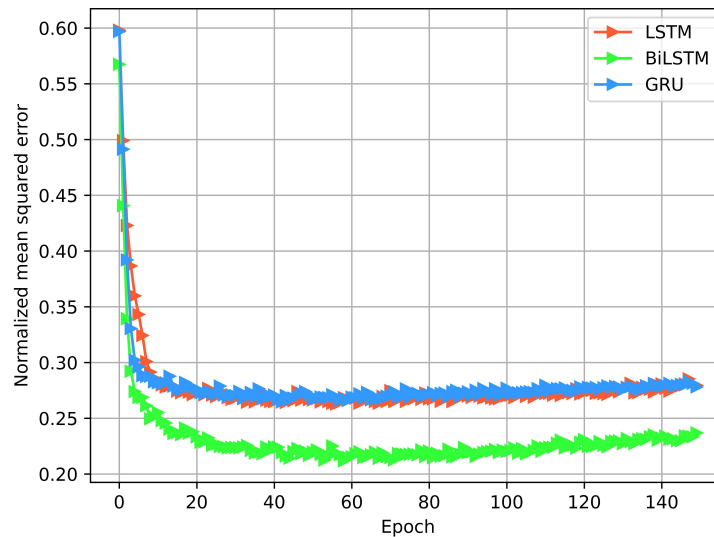


Figure 6. Normalized mean squared error for the 10-device network.

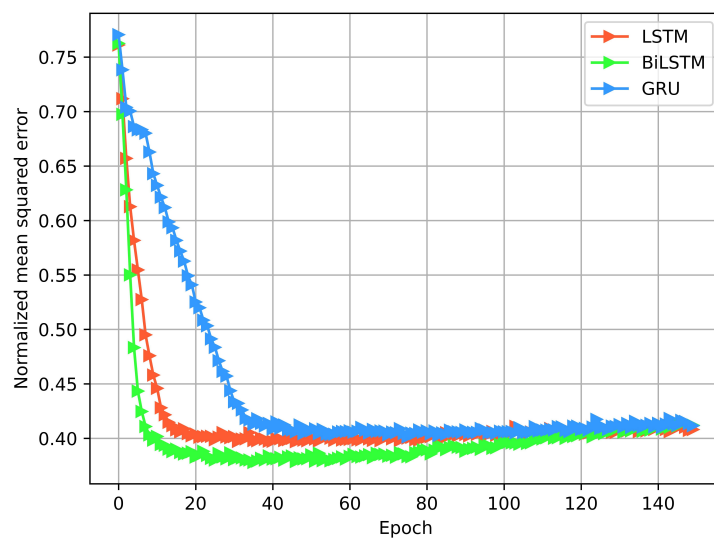


Figure 7. Normalized mean squared error for the 20-device network.

To assess the performance of complex RNN structures, we designed a model incorporating an LSTM layer, a GRU layer, and a BiLSTM layer. The input data first undergo processing by the LSTM layer, followed by a dropout layer with a retention rate of 85% (1–0.15). Similarly, the output from the LSTM layer feeds into the GRU layer, again followed by a dropout layer. Finally, the BiLSTM layer receives the processed data from the GRU layer, with another dropout layer applied. The output of the final dropout layer connects to a fully connected layer. As illustrated in Figure 8, for the 10-device network, the combined approach outperforms both the single-LSTM and -GRU models. However, the BiLSTM model still achieves superior performance compared to the combined model. Interestingly, when the number of devices in the network increases, the combined approach demonstrates significant improvement, surpassing the performance of single-layer models in terms of MSE, as shown in Figure 9.

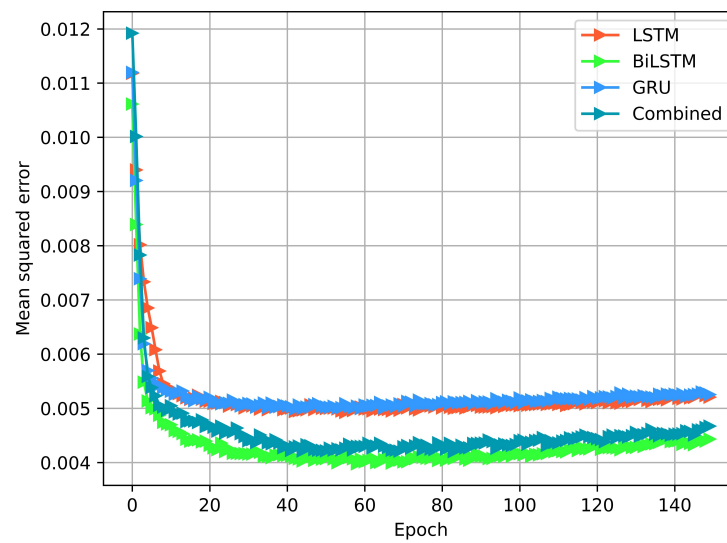


Figure 8. MSE performance comparison of a combined model (LSTM + GRU + BiLSTM) with a single-layer model for the 10-device network.

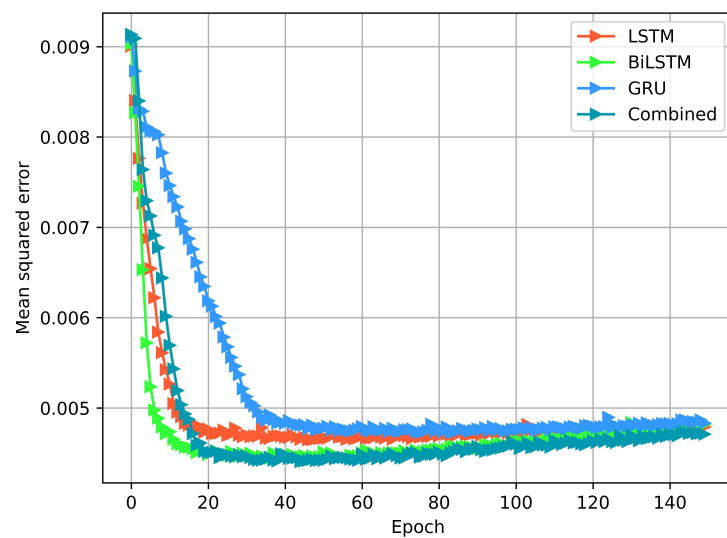


Figure 9. MSE performance comparison of a combined model (LSTM + GRU + BiLSTM) with a single-layer model for the 20-device network.

Figure 10 presents the maximum achievable sum rate for the 10-device network. It is evident from the figure that BiLSTM achieves a higher sum rate compared to the other two approaches, while LSTM and GRU demonstrate similar sum rates. Specifically, LSTM slightly outperforms GRU, with mean sum rates of 1.7289 and 1.7264, respectively. In the case of the 20-device network shown in Figure 11, the overall performance of all approaches improves, and each approach approaches the maximum sum rate. In this experiment, we observed the better performance of BiLSTM, as it is closer to the maximum sum rate. LSTM secures the second position, with its curve being above that of GRU.

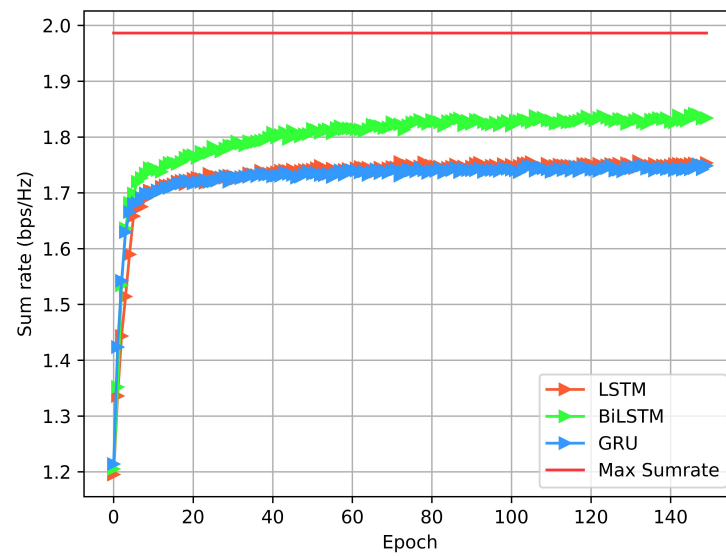


Figure 10. Sum rate achieved by different RNN approaches for the 10-device network.

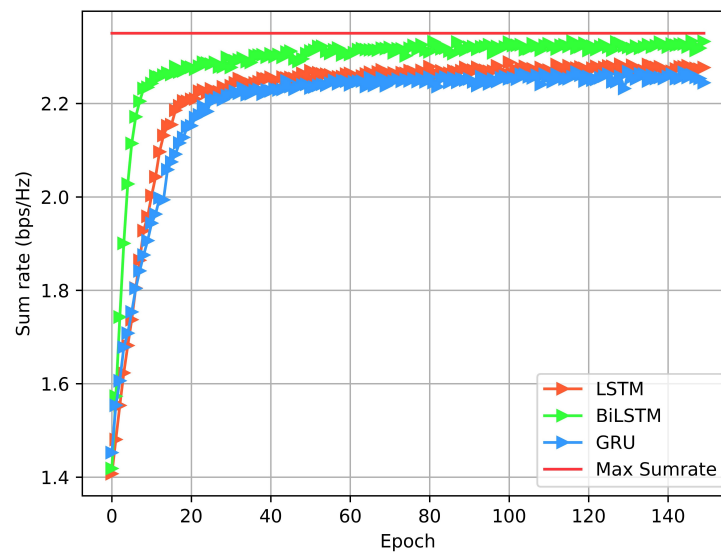


Figure 11. Sum rate achieved by different RNN approaches for the 20-device network.

Figure 12 compares the MSE of the proposed RNN approach with the study conducted in [44,48]. In the study by Sun et al. [44], deep neural networks were adopted to reduce interference in the wireless channel. Additionally, Chun et al. [48] proposed a deep learning model based on successive interference cancellation. The comparison reveals that the proposed RNN approach achieves a lower MSE in the validation set compared to the approaches described in [44,48].

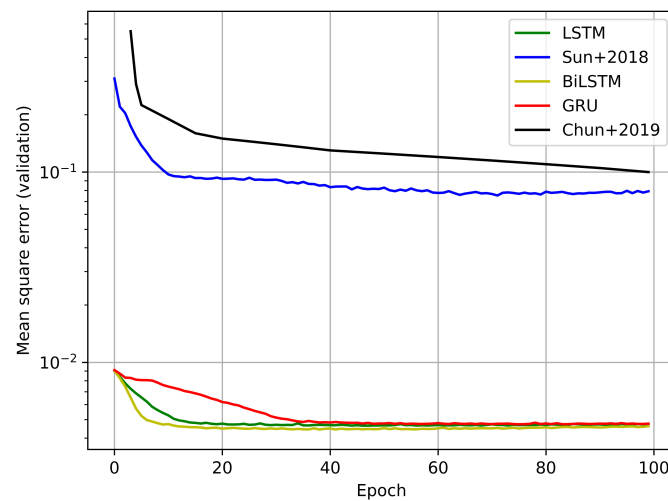


Figure 12. MSE comparison of different RNN approaches and other methods [44,48].

5. Conclusions

In this paper, we investigated the application of recurrent neural networks, specifically LSTM, BiLSTM, and GRU, to mitigate the effects of co-channel interference in wireless communication channels. We modeled two network scenarios: one with 10 devices and another with 20 devices. Data from each network were transmitted with controlled interference, and the received, corrupted data were used to train the RNN models. Our experiments revealed that BiLSTM achieved superior performance compared to LSTM and GRU, likely due to its enhanced ability to extract relevant features from the training data. Notably, LSTM and GRU exhibited similar results across both network sizes. However, a key limitation of this study is the potential performance degradation when the number of devices increased significantly. This may be attributed to the growing complexity of the interference environment and the limitations of the training data size. Future research directions can explore various avenues. As the number of devices in a network is expected to grow exponentially, the investigation of the performance of these RNN models with a massive swarm of devices would be valuable. Additionally, real-world communication channels experience dynamic interference due to environmental factors. Therefore, the incorporation of the dynamic changes in the interference into the training data could enhance the model's robustness. Furthermore, exploring more complex interference structures and new wireless communication channel models, such as intelligent reflecting surface-based communication and massive MIMO, could provide promising avenues for future research on interference mitigation using RNNs.

Author Contributions: Conceptualization, M.A.S.S. and M.H.R.; methodology, M.A.S.S., R.T. and Y.-H.Y.; software, M.A.S.S., M.H.R., M.A.A. and R.T.; validation, M.A.S.S., D.-D.H. and M.H.R.; formal analysis, M.A.S.S., M.H.R. and M.A.A.; investigation, M.A.S.S., R.T. and M.A.A.; resources, H.-K.S.; data curation, M.A.S.S. and R.T.; writing—original draft preparation, M.A.S.S.; writing—review and editing, M.A.S.S., M.H.R., M.A.A. and H.-K.S.; visualization, M.A.S.S., M.H.R. and M.A.A.; supervision, H.-K.S., Y.-H.Y. and D.-D.H.; project administration, H.-K.S.; funding acquisition, H.-K.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) under the Metaverse support program to nurture the best talents (IITP-2024-RS-2023-00254529). The grant was funded by the Korean government (MSIT) and in part by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2020R1A6A1A03038540), and it was also funded in part by the National Research Foundation of Korea (NRF) grant funded by the Korea Government (MSIT) (RS-2023-00219051).

Data Availability Statement: The data will be made available by the authors on request.

Conflicts of Interest: The authors declare no conflicts of interest.

References

- Wu, Q.; Li, G.Y.; Chen, W.; Ng, D.W.K.; Schober, R. An overview of sustainable green 5G networks. *IEEE Wirel. Commun.* **2017**, *24*, 72–80. [\[CrossRef\]](#)
- Duan, W.; Gu, J.; Wen, M.; Zhang, G.; Ji, Y.; Mumtaz, S. Emerging Technologies for 5G-IoV Networks: Applications, Trends and Opportunities. *IEEE Netw.* **2020**, *34*, 283–289. [\[CrossRef\]](#)
- Alzubaidi, O.T.H.; Hindia, M.N.; Dimyati, K.; Noordin, K.A.; Wahab, A.N.A.; Qamar, F.; Hassan, R. Interference challenges and management in B5G network design: A comprehensive review. *Electronics* **2022**, *11*, 2842. [\[CrossRef\]](#)
- Siddiqui, M.U.A.; Qamar, F.; Ahmed, F.; Nguyen, Q.N.; Hassan, R. Interference management in 5G and beyond network: Requirements, challenges and future directions. *IEEE Access* **2021**, *9*, 68932–68965. [\[CrossRef\]](#)
- Dangi, R.; Lalwani, P.; Choudhary, G.; You, I.; Pau, G. Study and investigation on 5G technology: A systematic review. *Sensors* **2021**, *22*, 26. [\[CrossRef\]](#) [\[PubMed\]](#)
- Goldsmith, A. *Wireless Communications*; Cambridge University Press: Cambridge, UK, 2005.
- Wang, J. CFAR-based interference mitigation for FMCW automotive radar systems. *IEEE Trans. Intell. Transp. Syst.* **2021**, *23*, 12229–12238. [\[CrossRef\]](#)
- Qaisar, Z.H.; Irfan, M.; Ali, T.; Ahmad, A.; Ali, G.; Glowacz, A.; Glowacz, W.; Caesarendra, W.; Mashraqi, A.M.; Draz, U.; et al. Effective beamforming technique amid optimal value for wireless communication. *Electronics* **2020**, *9*, 1869. [\[CrossRef\]](#)
- Zhang, J.; Zhang, J.; Björnson, E.; Ai, B. Local partial zero-forcing combining for cell-free massive MIMO systems. *IEEE Trans. Commun.* **2021**, *69*, 8459–8473. [\[CrossRef\]](#)
- Zambianco, M.; Verticale, G. Interference minimization in 5G physical-layer network slicing. *IEEE Trans. Commun.* **2020**, *68*, 4554–4564. [\[CrossRef\]](#)
- Gu, Z.; Vucetic, B.; Chikkam, K.; Aliberti, P.; Hardjawana, W. Graph Representation Learning for Contention and Interference Management in Wireless Networks. *IEEE/ACM Trans. Netw.* **2024**, 1–16. [\[CrossRef\]](#)
- Irkicatal, O.N.; Ceran, E.T.; Yuksel, M. Deep Reinforcement Learning Enhanced Rate-Splitting Multiple Access for Interference Mitigation. *arXiv* **2024**, arXiv:2403.05974.
- Ding, X.; Lei, Y.; Zou, Y.; Zhang, G.; Hanzo, L. Interference Management by Harnessing Multi-Domain Resources in Spectrum-Sharing Aided Satellite-Ground Integrated Networks. *IEEE Trans. Veh. Technol.* **2024**, 1–16. [\[CrossRef\]](#)
- Wang, Y.; Sun, M.; Cui, Q.; Chen, K.C.; Liao, Y. RIS-aided proactive mobile network downlink interference suppression: A deep reinforcement learning approach. *Sensors* **2023**, *23*, 6550. [\[CrossRef\]](#) [\[PubMed\]](#)
- Gul, O.M.; Kulhandjian, M.; Kantarci, B.; Touazi, A.; Ellement, C.; D'amours, C. Secure industrial iot systems via rf fingerprinting under impaired channels with interference and noise. *IEEE Access* **2023**, *11*, 26289–26307. [\[CrossRef\]](#)
- Voulodimos, A.; Doulamis, N.; Doulamis, A.; Protopapadakis, E. Deep learning for computer vision: A brief review. *Comput. Intell. Neurosci.* **2018**, *2018*, 7068349. [\[CrossRef\]](#)
- Lee, H.; Huang, C.; Yune, S.; Tajmir, S.H.; Kim, M.; Do, S. Machine friendly machine learning: Interpretation of computed tomography without image reconstruction. *Sci. Rep.* **2019**, *9*, 1–9. [\[CrossRef\]](#) [\[PubMed\]](#)
- Kadhim, A.I. Survey on supervised machine learning techniques for automatic text classification. *Artif. Intell. Rev.* **2019**, *52*, 273–292. [\[CrossRef\]](#)
- Seota, S.B.W.; Klein, R.; Van Zyl, T. Modeling e-behaviour, personality and academic performance with machine learning. *Appl. Sci.* **2021**, *11*, 10546. [\[CrossRef\]](#)
- Dornadula, V.N.; Geetha, S. Credit card fraud detection using machine learning algorithms. *Procedia Comput. Sci.* **2019**, *165*, 631–641. [\[CrossRef\]](#)
- Khanal, S.S.; Prasad, P.; Alsadoon, A.; Maag, A. A systematic review: Machine learning based recommendation systems for e-learning. *Educ. Inf. Technol.* **2020**, *25*, 2635–2664. [\[CrossRef\]](#)
- Santos, I.; Castro, L.; Rodriguez-Fernandez, N.; Torrente-Patino, A.; Carballal, A. Artificial neural networks and deep learning in the visual arts: A review. *Neural Comput. Appl.* **2021**, *33*, 121–157. [\[CrossRef\]](#)
- Haque, R.; Islam, N.; Islam, M.; Ahsan, M.M. A comparative analysis on suicidal ideation detection using NLP, machine, and deep learning. *Technologies* **2022**, *10*, 57. [\[CrossRef\]](#)
- Hu, S.; Chen, X.; Ni, W.; Hossain, E.; Wang, X. Distributed Machine Learning for Wireless Communication Networks: Techniques, Architectures, and Applications. *IEEE Commun. Surv. Tutor.* **2021**, *23*, 1458–1493. [\[CrossRef\]](#)
- Sejan, M.A.S.; Rahman, M.H.; Shin, B.S.; Oh, J.H.; You, Y.H.; Song, H.K. Machine learning for intelligent-reflecting-surface-based wireless communication towards 6G: A review. *Sensors* **2022**, *22*, 5405. [\[CrossRef\]](#) [\[PubMed\]](#)
- Rahman, M.H.; Sejan, M.A.S.; Aziz, M.A.; Tabassum, R.; Baik, J.I.; Song, H.K. A Comprehensive Survey of Unmanned Aerial Vehicles Detection and Classification Using Machine Learning Approach: Challenges, Solutions, and Future Directions. *Remote Sens.* **2024**, *16*, 879. [\[CrossRef\]](#)

27. Ahmed, S.F.; Alam, M.S.B.; Hassan, M.; Rozbu, M.R.; Ishtiaq, T.; Rafa, N.; Mofijur, M.; Shawkat Ali, A.; Gandomi, A.H. Deep learning modelling techniques: Current progress, applications, advantages, and challenges. *Artif. Intell. Rev.* **2023**, *56*, 13521–13617. [\[CrossRef\]](#)
28. Du, J.; Jiang, C.; Wang, J.; Ren, Y.; Debbah, M. Machine learning for 6G wireless networks: Carrying forward enhanced bandwidth, massive access, and ultrareliable/low-latency service. *IEEE Veh. Technol. Mag.* **2020**, *15*, 122–134. [\[CrossRef\]](#)
29. Aziz, M.A.; Rahman, M.H.; Sejan, M.A.S.; Baik, J.I.; Kim, D.S.; Song, H.K. Spectral Efficiency Improvement Using Bi-Deep Learning Model for IRS-Assisted MU-MISO Communication System. *Sensors* **2023**, *23*, 7793. [\[CrossRef\]](#) [\[PubMed\]](#)
30. Hsu, J.Y.; Wang, Y.F.; Lin, K.C.; Chen, M.Y.; Hsu, J.H.Y. Wind turbine fault diagnosis and predictive maintenance through statistical process control and machine learning. *IEEE Access* **2020**, *8*, 23427–23439. [\[CrossRef\]](#)
31. Gao, J.; Heng, F.; Yuan, Y.; Liu, Y. A novel machine learning method for multiaxial fatigue life prediction: Improved adaptive neuro-fuzzy inference system. *Int. J. Fatigue* **2024**, *178*, 108007. [\[CrossRef\]](#)
32. Zhu, G.; Liu, D.; Du, Y.; You, C.; Zhang, J.; Huang, K. Toward an intelligent edge: Wireless communication meets machine learning. *IEEE Commun. Mag.* **2020**, *58*, 19–25. [\[CrossRef\]](#)
33. Salehinejad, H.; Sankar, S.; Barfett, J.; Colak, E.; Valaee, S. Recent advances in recurrent neural networks. *arXiv* **2017**, arXiv:1801.01078.
34. Cossu, A.; Carta, A.; Lomonaco, V.; Bacciu, D. Continual learning for recurrent neural networks: An empirical evaluation. *Neural Netw.* **2021**, *143*, 607–627. [\[CrossRef\]](#) [\[PubMed\]](#)
35. Baliyan, A.; Batra, A.; Singh, S.P. Multilingual sentiment analysis using RNN-LSTM and neural machine translation. In Proceedings of the 2021 8th International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India, 17–19 March 2021; pp. 710–713.
36. Onan, A. Bidirectional convolutional recurrent neural network architecture with group-wise enhancement mechanism for text sentiment classification. *J. King Saud Univ.-Comput. Inf. Sci.* **2022**, *34*, 2098–2117. [\[CrossRef\]](#)
37. Zhou, Y.; Samiee, A.; Zhou, T.; Jalali, B. Deep learning interference cancellation in wireless networks. *arXiv* **2020**, arXiv:2009.05533.
38. Bhatia, A.; Robinson, J.; Carmack, J.; Kuzdeba, S. FPGA implementation of radio frequency neural networks. In Proceedings of the 2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 26–29 January 2022; pp. 0613–0618.
39. Grunau, S.; Block, D.; Meier, U. Multi-label wireless interference identification with convolutional neural networks. *arXiv* **2018**, arXiv:1804.04395.
40. Rock, J.; Roth, W.; Toth, M.; Meissner, P.; Pernkopf, F. Resource-efficient deep neural networks for automotive radar interference mitigation. *IEEE J. Sel. Top. Signal Process.* **2021**, *15*, 927–940. [\[CrossRef\]](#)
41. Rock, J.; Toth, M.; Messner, E.; Meissner, P.; Pernkopf, F. Complex signal denoising and interference mitigation for automotive radar using convolutional neural networks. In Proceedings of the 2019 22th International Conference on Information Fusion (FUSION), Ottawa, ON, Canada, 2–5 July 2019; pp. 1–8.
42. Mun, J.; Kim, H.; Lee, J. A deep learning approach for automotive radar interference mitigation. In Proceedings of the 2018 IEEE 88th Vehicular Technology Conference (VTC-Fall), Chicago, IL, USA, 27–30 August 2018; pp. 1–5.
43. Wei-Lung, M. Gps interference mitigation using derivative-free kalman filter-based rnn. *Radioengineering* **2016**, *25*, 519.
44. Sun, H.; Chen, X.; Shi, Q.; Hong, M.; Fu, X.; Sidiropoulos, N.D. Learning to optimize: Training deep neural networks for interference management. *IEEE Trans. Signal Process.* **2018**, *66*, 5438–5453. [\[CrossRef\]](#)
45. Rahman, M.H.; Sejan, M.A.S.; Aziz, M.A.; You, Y.H.; Song, H.K. HyDNN: A Hybrid Deep Learning Framework Based Multiuser Uplink Channel Estimation and Signal Detection for NOMA-OFDM System. *IEEE Access* **2023**, *11*, 66742–66755. [\[CrossRef\]](#)
46. Rahman, M.H.; Sejan, M.A.S.; Aziz, M.A.; Kim, D.S.; You, Y.H.; Song, H.K. Deep Convolutional and Recurrent Neural-Network-Based Optimal Decoding for RIS-Assisted MIMO Communication. *Mathematics* **2023**, *11*, 3397. [\[CrossRef\]](#)
47. Freire, P.; Srivallapanondh, S.; Spinnler, B.; Napoli, A.; Costa, N.; Prilepsky, J.E.; Turitsyn, S.K. Computational Complexity Optimization of Neural Network-Based Equalizers in Digital Signal Processing: A Comprehensive Approach. *J. Light. Technol.* **2024**, 1–25. [\[CrossRef\]](#)
48. Chun, C.J.; Kang, J.M.; Kim, I.M. Deep learning-based joint pilot design and channel estimation for multiuser MIMO channels. *IEEE Commun. Lett.* **2019**, *23*, 1999–2003. [\[CrossRef\]](#)

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.