Deep Bi-LSTM Detection for FTN-GFDM in Underserved Communication Scenarios

Mariana Baracat de Mello, Karine Barbosa Carbonaro e Luciano Leonel Mendes

***Abstract*— This work proposes a deep Bi-LSTM for symbol detection in FTN-GFDM systems. The estimator learns the non-linear mapping between matched filter outputs and transmitted symbols, handling ISI and colored noise. The network is trained offline using synthetically generated data and evaluated over AWGN and TIFS channels. Results show that the Bi-LSTM achieves competitive BER performance compared to the SD, while offering fixed and low complexity during inference, making it suitable for real-time and resource-constrained applications.**

***Keywords*— FTN-GFDM, Bi-LSTM, sequence detection, rural areas**

I. Introduction

Future sixth generation (6G) mobile networks are expected to incorporate a dedicated service category targeting fundamental connectivity demands in underserved areas [1], [2]. The Enhanced Remote Area Communications (eRAC) scenario must not be conceived as a mere downscaled version of urban 6G infrastructure; instead, it requires tailored design considerations that account for the distinct propagation conditions, low user density, and infrastructure limitations of remote and rural environments [3], [4].

In this scenario, sub-1 GHz frequency bands are particularly advantageous due to their favorable propagation characteristics, supporting long-range, non-line-of-sight communication. However, licensed spectrum in this range is often economically restrictive, necessitating novel frequency reuse schemes optimized for sparsely populated and geographically challenging areas. Therefore, the opportunistic use of unused spectrum in the Ultra High Frequency (UHF)/Very High Frequency (VHF) television bands, known as TV White Space (TVWS), stands out as a cost-effective alternative, offering excellent coverage and penetration capabilities [4].

However, despite their coverage advantages, sub-1 GHz frequency bands inherently limit the adoption of advanced technologies such as massive Multiple-Input Multiple-Output (MIMO) and Ultra-Dense Network (UDN), which are key enablers of spectral efficiency in urban fifth generation (5G) and 6G scenarios [5]. At these frequencies, longer wavelengths lead to physically larger antenna arrays, reducing spatial multiplexing gains, while limited bandwidth constrains capacity. Additionally, low user density and scarce infrastructure make ultra-dense deployments economically unfeasible in remote and rural areas. As a result, alternative approaches are required to enhance spectral efficiency without relying on dense infrastructure or large antenna systems [6].

Recent studies have suggested that integrating Faster-than-Nyquist (FTN) signaling with Generalized Frequency Division Multiplexing (GFDM) is a promising approach for this purpose in remote and rural scenarios [6], [7]. FTN increases symbol rates beyond the Nyquist limit by introducing controlled Inter-Symbol Interference (ISI), allowing more data to be transmitted within a given bandwidth [8], [9]. GFDM, in turn, is a flexible multicarrier modulation scheme whose configurable structure supports operation in fragmented spectrum and adaptation to diverse bandwidth and latency requirements [10]. The combined waveform, known as FTN-GFDM, reduces the spacing between subsymbols and subcarriers to increase spectral efficiency, albeit at the cost of both ISI and Inter-Carrier Interference (ICI) [11]. Nonlinear detectors, such as Maximum Likelihood (ML), can optimally mitigate this interference within the Mazo limit, but their computational complexity is prohibitive for practical systems [6], [12].

Therefore, this paper proposes the use of a Recurrent Neural Network (RNN) [13], [14], specifically a Bi-directional Long Short-Term Memory (Bi-LSTM) [15] network, to model and mitigate the ISI inherent in the FTN-GFDM scheme. RNNs are well suited for processing sequential data, as they retain contextual information through hidden states, enabling the learning of temporal dependencies. The Bi-LSTM architecture enhances this capability by processing input sequences in both forward and backward directions, allowing the network to capture dependencies from both past and future symbols. This bidirectional structure improves the model's ability to represent complex temporal patterns, leading to more accurate symbol detection in the presence of ISI [16]. A key advantage of the proposed neural network-based detection approach is its fixed and low computational complexity during inference. Once trained, the Bi-LSTM model performs a constant number of operations for each input sequence, resulting in predictable latency and making it well suited for real-time or resource-constrained applications.

II. FTN-GFDM System

Consider a FTN-GFDM system employing Quadrature Amplitude Modulation (QAM) modulation, where $N = KM$

Mariana Baracat de Mello and Luciano Leonel Mendes, National Institute of Telecommunications (Inatel), Santa Rita do Sapucaí - MG, e-mail: {mariana.baracat, lucianol}@inatel.br, Karine Barbosa Carbonaro, Faculty of Electrical Engineering, Federal University of Uberlândia, Patos de Minas - MG, e-mail: karine.carbonaro@ufu.br. This work has received partial funding from the project XGM-AFCCT-2024-2-15-1, supported by xGMobile – EMBRAPII-Inatel Competence Center on 5G and 6G Networks, with financial resources from the PPI IoT/Manufacturing 4.0 program of MCTI grant number 052/2023, signed with EMBRAPII. Additionally, this work was partially supported by the Ciência para Elas project (Fapemig, APQ-04523-23), the SEMEAR project (FAPESP, 22/09319-9), the Brasil 6G project (RNP/MCTI, 01245.010604/2020-14), Fapemig (PPE-00124-23), and CNPq-Brasil.

data symbols are transmitted across K subcarriers and M subsymbols. The prototype filter $\mathbf{g}$, circularly shifted in both time and frequency, shapes each symbol. FTN signaling introduces controlled ISI and ICI by reducing $\mathcal{K}$ and $\mathcal{M}$, which correspond to the subsymbol and subcarrier spacing, respectively. The time overlapping factor is defined as $\tau = \mathcal{S}/\mathcal{K}$, and the frequency overlapping factor as $\beta = \mathcal{P}/\mathcal{M}$, where $\tau < 1$ and/or $\beta < 1$ lead to ISI and ICI, respectively [12]. Here, $\mathcal{S}$ denotes the number of samples per period of the prototype filter, and $\mathcal{P}$ represents the total number of periods.

The transmitted FTN-GFDM signal in discrete-time can be expressed as

$$x[\nu] = \sum_{k=0}^{K-1} \sum_{m=0}^{M-1} s_{k,m} g_{k,m}[\nu], \tag{1}$$

where $s_{k,m}$ is the data symbol at the k-th subcarrier and m-th subsymbol, and the time-frequency shifted prototype filter is defined as

$$g_{k,m}[\nu] = \sqrt{\frac{\mathcal{N}}{N}} g\left[\langle \nu - m\tau\mathcal{S} \rangle_{\mathcal{N}}\right] \exp\left(j2\pi \frac{k\beta}{\mathcal{S}} \nu\right), \tag{2}$$

with $\nu = [0, 1, 2, \ldots, \mathcal{N}-1]$, $\mathcal{N}$ denoting the total number of samples, and $\langle \cdot \rangle_{\mathcal{N}}$ the modulo-$\mathcal{N}$ operation ensuring circularity.

To simplify signal modeling and analysis, the transmission can be written in matrix form as

$$\mathbf{x} = \mathbf{A}\mathbf{s}, \tag{3}$$

where $\mathbf{x}$ is the transmit signal vector, $\mathbf{s}$ is the vectorized data symbol stream, and $\mathbf{A}$ is the modulation matrix composed of all circularly time-frequency shifted versions of the prototype filter $g_{k,m}$, i.e.,

$$\mathbf{A} = \left[\mathbf{g}_{0,0}, \mathbf{g}_{1,0}, \ldots, \mathbf{g}_{K-1,0}, \mathbf{g}_{0,1}, \ldots, \mathbf{g}_{K-1,M-1}\right]. \tag{4}$$

Assuming the signal passes through a multipath fading channel represented by matrix $\mathbf{H}$, the received signal is given by

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{w}, \tag{5}$$

where $\mathbf{w} \sim \mathbb{C}\mathcal{N}(0, N_0\mathbf{I})$ is the additive white Gaussian noise vector.

After channel equalization, the demodulator applies a Matched Filter (MF) operation $\mathbf{A}^{\mathrm{H}}$ to obtain

$$\mathbf{r} = \mathbf{A}^{\mathrm{H}}\mathbf{H}^{-1}\mathbf{H}\mathbf{A}\mathbf{s} + \mathbf{A}^{\mathrm{H}}\mathbf{H}^{-1}\mathbf{w} = \mathbf{G}\mathbf{s} + \bar{\mathbf{w}}, \tag{6}$$

where $\mathbf{G} = \mathbf{A}^{\mathrm{H}}\mathbf{A}$ is the interference matrix encapsulating both ISI and ICI components. The noise term $\bar{\mathbf{w}} \sim \mathbb{C}\mathcal{N}(0, N_0(\mathbf{A}^{\mathrm{H}}(\mathbf{H}^{\mathrm{H}}\mathbf{H})^{-1}\mathbf{A}))$ becomes colored due to the non-orthogonal nature of FTN signaling.

The condition number of $\mathbf{G}$ significantly impacts the system's performance. Ill-conditioned matrices degrade the effectiveness of linear detectors such as Zero-Forcing (ZF) and Minimum Mean Square Error (MMSE), motivating the use of nonlinear strategies [6]. Maximum Likelihood Sequence Estimation (MLSE) achieves optimal Bit Error Rate (BER) performance but at prohibitive complexity. Hence, practical alternatives like Sphere Decoding (SD), Successive Symbol-by-Symbol Sequence Estimation (SSSSE), and Successive Symbol-by-Symbol go-back K Sequence Estimation (SSSgbKSE) offer a balance between complexity and BER performance [12].

III. Recurrent Neural Networks

RNNs are a class of neural networks designed to model sequential data by maintaining a form of internal memory through recurrent connections [17]. Their architecture introduces temporal dependencies by feeding the hidden state from the previous time step into the computation of the current state, thus characterizing them as discrete-time dynamic systems [18], [13]. Let $\mathbf{x}_t^{l-1}$ denotes the input to layer l at time step t, and $\mathbf{h}_t^l$ represents the hidden state at the same layer and time step. The equations governing the forward pass of a standard RNN layer l are given by [16], [19]

$$\mathbf{h}_t^l = \phi\left(\mathbf{W}_{\mathrm{hh}}^l \mathbf{h}_{t-1}^l + \mathbf{W}_{\mathrm{xh}}^l \mathbf{x}_t^{l-1} + \mathbf{b}_{\mathrm{h}}^l\right), \tag{7}$$

$$\mathbf{y}_t^l = \mathbf{W}_{\mathrm{hy}}^l \mathbf{h}_t^l + \mathbf{b}_{\mathrm{y}}^l, \tag{8}$$

where $\phi(\cdot)$ is a nonlinear activation function (typically *tanh* or ReLU); $\mathbf{W}_{\mathrm{xh}}$, $\mathbf{W}_{\mathrm{hh}}$, and $\mathbf{W}_{\mathrm{hy}}$ are weight matrices corresponding to the input-to-hidden, hidden-to-hidden (recurrent), and hidden-to-output transformations, respectively; and $\mathbf{b}_{\mathrm{h}}^l$ and $\mathbf{b}_{\mathrm{y}}^l$ are the corresponding bias vectors. The parameters of the network are shared across time steps, enabling the RNN to learn temporal patterns and dependencies in the data. This parameter sharing also allows the network to generalize across sequence lengths.

Training is commonly performed using the Backpropagation Through Time (BPTT) algorithm, which extends the standard backpropagation method to sequential data by unrolling the network over time [20]. The total gradient of the loss function with respect to the trainable parameters is computed by accumulating the contributions from each time step as follows [19]

$$\frac{\partial \mathcal{L}}{\partial \theta} = \sum_{t=1}^{T} \frac{\partial \mathcal{L}_t}{\partial \theta}. \tag{9}$$

Here, $\mathcal{L}$ denotes the total loss over the input sequence of length T, $\mathcal{L}_t$ is the loss at time step t, and θ represents the set of trainable parameters of the network, such as weight matrices and biases.

However, BPTT is prone to the vanishing and exploding gradient problems when processing long sequences, which can hinder the learning of long-term dependencies [14], [20]. To address these limitations, more robust recurrent architectures, such as the Long Short-Term Memory (LSTM) [21] and the Gated Recurrent Unit (GRU) [22], were introduced. These models incorporate gating mechanisms that improve gradient flow and enable the network to capture temporal dependencies over longer time spans more effectively.

IV. Neural Network-Based Detection for FTN-GFDM Systems

As previously discussed, symbol detection in FTN-GFDM systems poses a significant challenge due to the presence of

ISI, which requires high-complexity nonlinear detectors to achieve a BER performance close to that of ML detection. The detection problem can be reformulated to address this complexity as a supervised regression task, employing a neural network to capture the temporal dependencies within the transmitted symbol sequence.

The neural network is trained to learn the nonlinear relationship between the sequence at the MF output $\mathbf{r}$ and the original transmitted symbols $\mathbf{s}$, addressing both the interference pattern, characterized by the matrix $\mathbf{G}$, and the colored noise resulting from the non-orthogonality of the modulation matrix $\mathbf{A}$ and the channel equalization process.

The network is defined as a function $f(\mathbf{r};\theta)$, where θ represents the set of trainable weights and biases. The parameters are optimized by minimizing a loss function that quantifies the discrepancy between the predicted output and the reference symbol vector, expressed as

$$\min_{\theta} \mathbb{E}\left[\mathcal{L}(\mathbf{s}, f(\mathbf{r};\theta))\right], \tag{10}$$

where $\mathbb{E}[\cdot]$ denotes the expectation operator.

A. Training Dataset Generation

The neural network is trained offline using a synthetically generated dataset constructed from the FTN-GFDM system model. Each training sample comprises a randomly generated symbol vector $\mathbf{s}^{(i)}$ and its corresponding received vector $\mathbf{r}^{(i)}$, obtained from the MF output as described in (6). Therefore, the dataset is defined as

$$\mathcal{S} = \left\{ (\mathbf{r}^{(i)}, \mathbf{s}^{(i)}) \right\}_{i=1}^{\mathcal{T}_{\text{train}}}, \tag{11}$$

where $\mathcal{T}_{\text{train}}$ denotes the total number of training samples. The neural network is trained to learn the temporal mapping from $\mathbf{r}$ to $\mathbf{s}$, capturing the effects of ISI the correlation introduced in the noise by channel equalization.

B. Proposed Bi-LSTM Architecture

In the FTN-GFDM system, the estimation of each symbol must consider interference from both preceding and succeeding symbols within the block. To model such dependencies, LSTM networks are employed instead of conventional RNNs, as they provide enhanced capability to retain relevant information across long sequences.

A Bi-LSTM architecture can be adopted to improve performance further, allowing the network to process the input sequence in both forward and backward directions. This bidirectional structure enables the network to exploit the ISI/ICI pattern within the FTN-GFDM signal, effectively capturing the mutual influence among overlapping symbols and improving detection accuracy.

Let $\mathbf{r} = [\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_N]$ denote the real-valued input sequence, where each $\mathbf{r}_t \in \mathbb{R}^2$ contains the real and imaginary parts of the t-th received sample at the MF output, with $t = 1, 2, \ldots, N$. This sequence is processed by the Bi-LSTM network over N time steps, one for each symbol in the block. At each step t, the LSTM cell generates a hidden state vector $\mathbf{h}_t \in \mathbb{R}^U$, where U denotes the number of hidden units in the layer. This defines the dimensionality of the hidden representation used to model interference among symbols across the block.

At each time step t, the forward and backward layers produce hidden state vectors $\overrightarrow{\mathbf{h}}_t$ and $\overleftarrow{\mathbf{h}}_t$, respectively, which are concatenated to form the Bi-LSTM output, given by

$$\mathbf{h}_t = \left[\overrightarrow{\mathbf{h}}_t, \overleftarrow{\mathbf{h}}_t\right]. \tag{12}$$

Each LSTM cell maintains a hidden state and a cell state, which are updated at each time step through standard gating mechanisms involving input, forget, and output gates, as described in [21].

A second Bi-LSTM layer is stacked on top of the first to enhance the network's ability to capture symbol dependencies. The first layer outputs a sequence of hidden vectors $\{\mathbf{h}_t\}_{t=1}^{N}$, which is processed by the second layer to produce a compact representation of the input sequence.

The output of the second Bi-LSTM layer is then passed to a fully connected (dense) layer responsible for estimating the transmitted symbol vector. A hyperbolic tangent activation function is applied to ensure that the outputs are constrained to the $(-1, +1)$ interval, which matches the Binary Phase Shift Keying (BPSK) modulation used. The network output can be expressed as

$$\hat{\mathbf{s}} = \tanh(\mathbf{W}\mathbf{h} + \mathbf{b}), \tag{13}$$

where $\mathbf{W}$ and $\mathbf{b}$ denote the weight matrix and bias vector of the dense layer, respectively, and $\mathbf{h}$ is the final hidden representation produced by the second Bi-LSTM layer.

This architecture allows the network to learn a nonlinear mapping from the received sequence $\mathbf{r}$ to the transmitted symbols $\mathbf{s}$, handling interference and colored noise caused by the non-orthogonality of the FTN-GFDM signal.

V. Performance Evaluation

The proposed architecture described in Section IV was implemented using a two-layer Bi-LSTM network followed by a fully connected output layer. The input to the network consists of sequences of $N = 25$ complex samples, separated into their real and imaginary components, and organized as a matrix of shape $(25, 2)$. The network architecture begins with a Bi-LSTM layer comprising 128 units in each direction, configured to return a sequence of hidden states across all time steps. The second Bi-LSTM layer, with 64 units per direction, processes the output of the previous layer and summarizes the sequence information into a single feature vector. This output is then passed to a dense layer with 25 neurons and a `tanh` activation function, which produces continuous-valued symbol estimates in the range $(-1, 1)$, appropriate for the BPSK constellation.

Figure 1 shows the Bi-LSTM network architecture, including the input/output dimensions and the number of trainable parameters per layer. The parameter count for each Bi-LSTM layer is given by $8(dh + h^2 + h)$, where d is the input size per time step and h is the number of hidden units in each direction. Thus, the proposed network contains a total of 301,721 trainable parameters.

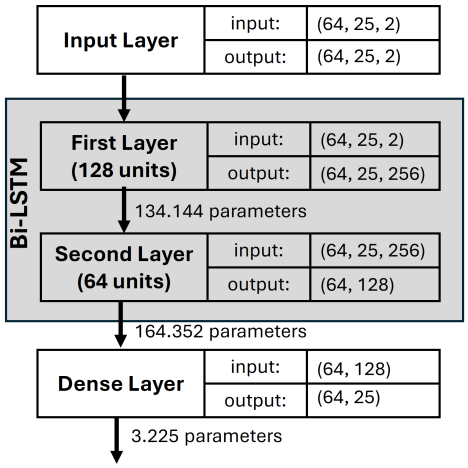


Fig. 1: Architecture of the proposed Bi-LSTM-based estimator, showing the input/output dimensions and the number of trainable parameters per layer.

Three datasets were generated using a simulated FTN-GFDM system with parameters detailed in Table I[1]. For each dataset, 80% of the samples were used for training and 20% for validation. The corresponding models were evaluated using separate test datasets, generated independently from the training data. Training Signal-to-Noise Ratio (SNR) values were sampled at $0, 2, 4, 6, 8, 10$ dB, while testing covered the full range from 0 to 10 dB in 1 dB steps. The number of training and test samples per SNR is presented in Table II.

TABLE I: FTN-GFDM System Parameters Used in Dataset Generation

Parameters	Values
Samples per period of the prototype filter ($\mathcal{S}$)	5
Periods of the prototype filter ($\mathcal{P}$)	4
Total number of samples ($\mathcal{N} = \mathcal{SP}$)	20
Time overlap factor (τ)	0.8
Frequency overlap factor (ϕ)	1
Number of subcarriers per block (K)	5
Number of subsymbols per block (M)	5
Number of data symbols per block ($N = KM$)	25
Prototype filter	Dirichlet

TABLE II: Training and Testing Setup for Datasets

Dataset	Training Samples per SNR	Test Samples per SNR
Dataset 1	5000	10000
Dataset 2	25000	50000
Dataset 3	50000	100000

Training followed a supervised learning approach, in which each input sample consisted of the MF outputs paired with the corresponding transmitted symbols, as defined in (11). The models were trained using the Adam optimizer with an initial learning rate of 0.001, for 10 epochs and a mini-batch size of 64.

[1]The datasets are available at: `https://github.com/Mariana-Baracat/ftn-gfdm-datasets`

Figures 2a and 2b show the BER performance of the Bi-LSTM models trained with three different datasets evaluated over Additive White Gaussian Noise (AWGN) and Time-Invariant Frequency-Selective (TIFS) channels, respectively. The TIFS channel is characterized by an impulse response given by $\mathbf{h} = [1, 0.2, 0.1, 0.04]^{\mathrm{T}}$. The SD was used as a performance benchmark for comparison purposes. Although the SD achieves near-optimal performance for BPSK over both AWGN and TIFS channels, its computational complexity depends on the SNR and the number of transmitted symbols N, and can grow exponentially in the worst case,[2] which limits its applicability in latency-sensitive or resource-constrained scenarios.

In contrast, the Bi-LSTM detector provides competitive performance with constant inference-time complexity, since training is performed offline. This makes it an attractive alternative for scenarios requiring low computational cost while maintaining acceptable BER performance. Furthermore, the Bi-LSTM models exhibited good generalization capability, even for SNR values not included in the training set.

The results also indicate that the proposed Bi-LSTM neural network exhibits good generalization capability across different channel conditions in the FTN-GFDM signal detection task, i.e., when training and testing are performed on samples generated under distinct channel models. Specifically, the model trained on AWGN samples and tested on equalized TIFS data achieves better performance than the model trained and tested on TIFS data.

Moreover, it is observed that the BER performance improves as the number of training samples per SNR increases. The model trained with Dataset 1 exhibits inferior performance due to the limited number of examples available during training. In addition, this model was evaluated using only 10000 samples in the testing phase, which may lead to fluctuations in the BER curve, especially at high SNR values, where the absolute number of errors tends to be small and statistically less stable.

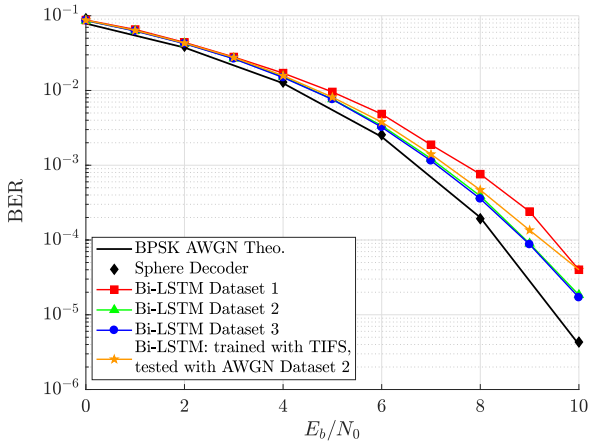
The models trained with Dataset 2 and Dataset 3 achieved better performance than the model trained with Dataset 1. Moreover, the BER curves of the models trained with Dataset 2 and Dataset 3 are very close, indicating that the model reaches its learning capacity with these dataset sizes. Increasing the training size to 50000 samples provides only a small performance gain, at the cost of increased computational complexity and training time, which may not be justifiable in resource-constrained applications.

This trade-off is evident in the training times reported in Table III, which increase with the number of samples per SNR. For the AWGN channel, training time increases from 4 to 49 minutes on a system equipped with an AMD Ryzen 53500U processor (2.10 3.70 GHz, 4 cores / 8 threads) as the dataset size grows from 5000 to 50000 samples. A similar trend is observed for the TIFS channel. These results highlight the increased training cost associated with larger datasets, especially given the small performance improvement beyond 25000 samples.

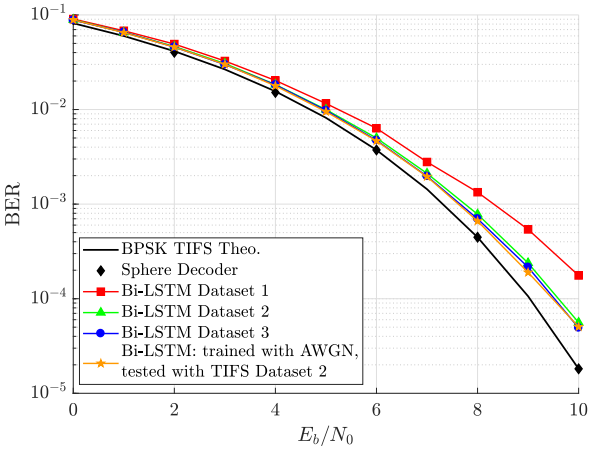
[2]Upper and lower bounds on the SD complexity can be found in [12].

TABLE III: Training Setup for Datasets

	AWGN		TIFS	
Dataset	MSE	Time	MSE	Time
1	0.082936	04m21s	0.091955	4m40s
2	0.074907	25m22s	0.081125	26m32s
3	0.075386	45m21s	0.080009	49m50s



(a) BER over AWGN channel.



(b) BER over TIFS channel.

Fig. 2: BER performance of Bi-LSTM models trained with three different datasets evaluated over different channels.

VI. Conclusões

This paper proposed a Bi-LSTM-based detector for FTN-GFDM systems, aiming to address the challenges of symbol detection under severe ISI conditions. Simulation results over AWGN and TIFS channels demonstrated that the proposed architecture achieves competitive BER performance compared to traditional nonlinear detectors, while offering a fixed and low computational complexity during inference. Additionally, the results showed that detection accuracy improves with larger training datasets; however, performance gains become marginal beyond a certain dataset size, reflecting the learning capacity of the proposed model. These results highlight the potential of Bi-LSTM-based detection as an efficient and scalable solution for FTN-GFDM systems operating in constrained communication scenarios, such as eRAC.

References

[1] K. B. Letaief, W. Chen, Y. Shi, J. Zhang, and Y.-J. A. Zhang, "The Roadmap to 6G: AI Empowered Wireless Networks," *IEEE Communications Magazine*, vol. 57, no. 8, pp. 84–90, 2019.

[2] S. Chen, Y.-C. Liang, S. Sun, S. Kang, W. Cheng, and M. Peng, "Vision, Requirements, and Technology Trend of 6G: How to Tackle the Challenges of System Coverage, Capacity, User Data-Rate and Movement Speed," *IEEE Wireless Communications*, vol. 27, no. 2, pp. 218–228, 2020.

[3] L. L. Mendes, C. S. Moreno, M. V. Marquezini, A. M. Cavalcante, P. Neuhaus, J. Seki, N. F. T. Aniceto, H. Karvonen, I. Vidal, F. Valera, *et al.*, "Enhanced Remote Areas Communications: The Missing Scenario for 5G and Beyond 5G Networks," *IEEE Access*, vol. 8, pp. 219859–219880, 2020.

[4] A. M. Cavalcante, M. V. Marquezini, L. Mendes, and C. S. Moreno, "5G for remote areas: Challenges, Opportunities and Business Modeling for Brazil," *IEEE Access*, vol. 9, pp. 10829–10843, 2021.

[5] W. Jiang, B. Han, M. A. Habibi, and H. D. Schotten, "The Road Towards 6G: A Comprehensive Survey," *IEEE Open Journal of the Communications Society*, vol. 2, pp. 334–366, 2021.

[6] M. Mello, L. Mendes, and T. Barbosa, "Spectrum Efficient GFDM Based on Faster Than Nyquist Signaling," *JCIS*, vol. 35, pp. 349–356, Dec 2020.

[7] M. B. de Mello and L. Leonel Mendes, "Time-frequency ftn signaling for gfdm," in *2019 16th International Symposium on Wireless Communication Systems (ISWCS)*, pp. 16–20, 2019.

[8] J. E. Mazo, "Faster-than-Nyquist signaling," *The Bell System Technical Journal*, vol. 54, pp. 1451–1462, Oct 1975.

[9] F. Rusek and J. B. Anderson, "The two dimensional Mazo limit," in *IEEE Int. Symp. Inf. Theory*, pp. 970–974, Sept. 2005.

[10] L. L. Mendes, N. Michailow, M. Matthé, I. Gaspar, D. Zhang, and G. Fettweis, "GFDM: providing flexibility for the 5G physical layer," *Opportunities in 5G Networks: A Research and Development Perspective*, p. 325, 2016.

[11] J. B. Anderson, F. Rusek, and V. Öwall, "Faster-than-Nyquist signaling," *Proceedings of the IEEE*, vol. 101, pp. 1817–1830, Aug 2013.

[12] M. B. Mello and L. L. Mendes, "Low-Complexity Detection Algorithms Applied to FTN-GFDM Systems," *IEEE Access*, vol. 10, pp. 101683–101696, 2022.

[13] D. P. Mandic and J. Chambers, *Recurrent Neural Networks for Prediction: Learning Algorithms,Architectures and Stability*. USA: John Wiley & Sons, Inc., 2001.

[14] D. Mienye, T. Swart, and G. Obaido, "Recurrent neural networks: A comprehensive review of architectures, variants, and applications," *Information*, vol. 15, p. 517, 08 2024.

[15] M. A. Aziz, M. H. Rahman, M. Abrar Shakil Sejan, R. Tabassum, D.-D. Hwang, and H.-K. Song, "Bidirectional deep learning decoder for polar codes in flat fading channels," *IEEE Access*, vol. 12, pp. 149580–149592, 2024.

[16] S. Lai and M. Li, "Recurrent neural network assisted equalization for ftn signaling," in *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, pp. 1–6, 2020.

[17] A. Geron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly Media, Inc., 2nd ed., 2019.

[18] T. Mohaidat and K. Khalil, "A survey on neural network hardware accelerators," *IEEE Transactions on Artificial Intelligence*, vol. 5, no. 8, pp. 3801–3822, 2024.

[19] C. Sun, M. Song, D. Cai, B. Zhang, S. Hong, and H. Li, "A systematic review of echo state networks from design to application," *IEEE Transactions on Artificial Intelligence*, vol. 5, no. 1, pp. 23–37, 2024.

[20] P. Werbos, "Backpropagation through time: what it does and how to do it," *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1550–1560, 1990.

[21] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[22] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (A. Moschitti, B. Pang, and W. Daelemans, eds.), (Doha, Qatar), pp. 1724–1734, Association for Computational Linguistics, oct 2014.