2025 IEEE 36th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC) | 979-8-3503-6323-4/25/$31.00 ©2025 IEEE | DOI: 10.1109/PIMRC62392.2025.11275123

Jamming Detection in Cell-Free MIMO with Dynamic Graphs

Ali Hossary, Laura Crosara, and Stefano Tomasin
Dept. of Information Engineering (DEI), University of Padova, Italy
email: {ali.hossary@, laura.crosara@, stefano.tomasin@}unipd.it

***Abstract*—Jamming attacks pose a critical threat to wireless networks, particularly in cell-free massive MIMO systems, where distributed access points and user equipment (UE) create complex, time-varying topologies. This paper proposes a novel jamming detection framework leveraging dynamic graphs and graph convolution neural networks (GCN) to address this challenge. By modeling the network as a dynamic graph, we capture evolving communication links and detect jamming attacks as anomalies in the graph evolution. A GCN-Transformers-based model, trained with supervised learning, learns graph embeddings to identify malicious interference. Performance evaluation in simulated scenarios with moving UEs, varying jamming conditions and channel fadings, demonstrates the method's effectiveness, which is assessed through accuracy and F1 score metrics, achieving promising results for effective jamming detection.**

***Index Terms*—Jamming Detection, Dynamic Graphs, and Graph Neural Networks.**

I. Introduction

Wireless communication increasingly adopts cell-free architectures to enhance connectivity and spectral efficiency. Cell-free multiple-input multiple-output (MIMO) relies on access points (APs) that jointly serve user equipments (UEs) without predefined cell boundaries. This paradigm shift introduces new challenges related to network dynamics and security [1].

As reliance on wireless services continues to grow, security threats have become a major concern. Wireless networks, due to the shared nature of the radio spectrum, are particularly vulnerable to jamming [2]. In MIMO wireless networks, traditional jamming detection methods rely on statistical models, which struggle to adapt to the complexities of dynamic wireless environments [3]. In contrast, deep learning (DL) techniques can be applied using a data-driven approach [4]. In [5], a jammer detection method for massive MIMO systems is proposed, utilizing unused pilots during the training phase, assuming that the jammer lacks prior knowledge of the pilot patterns. The base station detects the presence of a jammer by analyzing the received signal on these unused pilots and employing a generalized likelihood ratio test (GLRT). Recent advancements have introduced new solutions, including neural networks (NNs) for jamming detection [6]. DL approaches, such as convolutional neural networks (CNNs), have been employed in [7], [8] to analyze spectrogram images for jamming detection, outperforming conventional feature-based methods. Recent advances are tailored to the characteristics of 5G networks [9]–[12]. In [13], a low-overhead intermittent jamming detection scheme for IoT networks is proposed, leveraging anchor nodes along with signal strength and multipath profile features. Furthermore, federated learning has been investigated for distributed jamming detection in flying ad-hoc networks [14]. However, all these solutions are agnostic of the network structures and are not suited for cell-free communications where synchronization is looser.

When users are mobile and channel conditions vary, modeling network behavior is crucial. *Dynamic graphs* offer a powerful representation for the evolving topology of wireless networks [15], where nodes correspond to APs and UEs, and edges represent communication links based on signal strength and interference levels. To process and analyze dynamic graphs data, *graph neural networks (GNNs)* provides a powerful framework. Inspired by CNNs, GNNs are designed to operate on graph structures, enabling tasks such as node classification, link prediction, and other graph-related learning problems [16].

In this paper, we propose a novel framework to model cell-free massive MIMO communication, exploiting dynamic graphs to capture the time variability of the communication scenario. Then, we present a novel approach for jamming detection, leveraging dynamic graphs and GNN-based architectures. Our approach identifies jamming attacks by learning latent representations of network states and monitoring deviations from expected patterns. We evaluate the proposed method using simulations that model mobility, connectivity, and interference scenarios, demonstrating its effectiveness.

The rest of this paper is organized as follows. Section II presents the cell-free MIMO system model. Section III presents the GNN-based jamming detection framework. Section IV evaluates detection performance through simulations. Finally, Section V draws the conclusions.

II. System Model

We consider a cell-free massive MIMO network [17] with M APs and M UEs, focusing on the downlink transmission. Each UE is equipped with a single antenna, and each AP is equipped with N_{A} antennas. APs are static, while UEs are moving. We adopt a discrete-time model with sampling

This work is supported by the project ISP5G+ (CUP D33C22001300002), which is part of the SERICS program (PE00000014) under the NRRP MUR program funded by the EU-NGEU" and by the European Commission through the Horizon Europe/JU SNS project ROBUST-6G (Grant Agreement no. 101139068).

interval T, considering the network state at time instants nT, with $n \in \mathbb{Z}$. Each AP is associated with a single UE, and uses maximal ratio precoding to transmit data to its served UE, we may have more UEs than AE, but still at any given time only one UE is connected to each AP. Moreover, we account for the presence of a jammer that aims at corrupting the communication between APs and UEs. Each AP is transmitting with unitary power to each UE.

Channel Model: Let $\boldsymbol{h}(k,m,n)$ denote the $N_{\mathrm{A}} \times 1$ vector of the narrowband baseband equivalent channel between the k-th UE and m-th AP at time nT. We consider a Rician fading channel; thus, the channel vector is modeled as

$$\boldsymbol{h}_{k,m}(n) = \beta\sigma_{k,m}(n) + \sqrt{1-\beta^2}\boldsymbol{g}_{k,m}(n), \tag{1}$$

with $\beta = \sqrt{\frac{K}{K+1}}$ a constant (and K is the Ricean K-factor) and $\boldsymbol{g}_{k,m}(n)$ being a $N_{\mathrm{A}} \times 1$ random matrix having i.i.d. zero-mean complex Gaussian entries. The variance of each entry of $\boldsymbol{g}_{k,m}(n)$ is determined by the path-loss model, which characterizes the received signal power as a function of the distance $d_{k,m}(n)$ between the k-th UE and the m-th AP at time nT, i.e.,

$$\sigma^2_{k,m}(n) = \frac{d_0^2}{d^2_{k,m}(n)}, \tag{2}$$

with $d_0 = 100\,\mathrm{m}$ representing the distance at which the channel has unitary variance. With $\beta = 1$ we obtain a deterministic model, while varying $\beta \in [0,1]$ we configure the randomness of the fading channel. We assume that reception is affected by additive white Gaussian noise (AWGN) with variance σ^2 per antenna.

Signal-to-noise-plus-interference Ratio: The transmitter applies maximal ratio (MR) precoding to steer the transmitted signal towards the intended user, and, in the absence of jamming, a connection is established from the AP m to the UE k at time nT if the signal-to-interference-plus-noise ratio (SINR)

$$\Gamma_{k,m}(n) = \frac{||\boldsymbol{h}_{k,m}(n)||^4}{\sigma^2 + \sum_{m' \neq m} |\boldsymbol{h}^H_{k,m}(n)\boldsymbol{h}_{k,m'}(n)|^2}, \tag{3}$$

is above a threshols Γ_0, i.e.,

$$\Gamma_{k,m}(n) > \Gamma_0. \tag{4}$$

Note that the formula includes the interference from other APs.

Mobility Model: We consider a system with UEs and APs ditributed within a square area of edge length L. The coordinates of each AP, indexed by m, are positioned at fixed locations that cover the area. At $n = 0$, the UEs are uniformly distributed within the square $[0, L]$. The coordinates of the position of user k at time nT are

$$\begin{aligned} x_k(n+1) &= x_k(n) + (v_{x,k} + w_{x,k}(n+1))T, \\ y_k(n+1) &= y_k(n) + (v_{y,k} + w_{y,k}(n+1))T, \end{aligned} \tag{5}$$

where $v_{x,k}$ and $v_{y,k}$ are the reference velocities of user k, uniformly distributed in the interval $[0, v_{\max}]$. The terms $w_{x,k}(n+1)$ and $w_{y,k}(n+1)$ are zero-mean Gaussian components with variance σ^2_w. If a user reaches the boundary of the square, its position is reset to a new location, uniformly sampled within the square, and assigned a new reference velocity. We assume that each user maintains a minimum distance $d_{\min}$ from any AP.

A. User Assignment Rule

We adopt the following rule for the assignment of UE to its serving AP. We proceed iteratively. We start with the full list of APs and UEs, and select the UE k and AP m that have the minimum distance among all pairs in the list. We assign UE k to AP m, then we remove a couple of devices from the list. The next iteration identifies the next AP-UE couple among the non-assigned APs and UEs.

Note that this procedure generates the assignment between APs and UEs, while an effective communication link (connection) between each couple is obtained only if condition (4) is satisfied.

B. Jammer Behavior

We consider the presence of a jammer that intermittently affects the communication between UEs and APs. Time is divided into F frames, each of duration T_{F}. Within each frame, the jammer remains active for a duration $\tau \in [0, T_{\mathrm{F}}]$. The jammer is equipped with a single antenna since its target is to disrupt any communication around it. When the jammer is *active*, the resulting SINR for a transmission from AP m to UE k at time nT becomes

$$\Gamma_{k,m}(n) = \frac{||\boldsymbol{h}_{k,m}(n)||^4}{\sigma^2 + P_J + \sum_{m' \neq m} |\boldsymbol{h}^H_{k,m}(n)\boldsymbol{h}_{k,m'}(n)|^2}, \tag{6}$$

where σ^2_{J} is the jammer transmit power, $P_J = \sigma^2_{\mathrm{J}}|S_k(n)|^2$, and $S_k(n)$ is the complex scalar channel from the jammer to UE k at time nT, according to the Rician model (1).

III. Jamming Detection By Dynamic Graph

We model the cell-free massive MIMO network as a dynamic connection graph $\{G(n)\}$, where $G(n)$ is the connection graph at time nT and T is the sampling time of the graph representation. In particular, each graph $G(n)$ has $N = 2M$ nodes (in the set $V(n)$), corresponding to both the APs and the UEs. The edges (collected in the set $E(n)$) represent the connections between APs and UEs. Specifically, an edge exists between UE k and AP m when (4) is satisfied. Each edge from AP m to UE k is labeled with the vector $\boldsymbol{w}_{k,m}(n) = [\alpha d_{k,m}, \zeta\gamma_{k,m}]$, where α and ζ are normalization factors that ensure proper scaling between distance and SINR values. The edge weights encode key connectivity metrics:

- *connection distance* $d_{k,m}(n)$, which defines the physical distance between an AP and a UE,
- *link quality* $\Gamma_{k,m}(n)$, quantified by the SINR, captures the reliability and performance of the communication link.

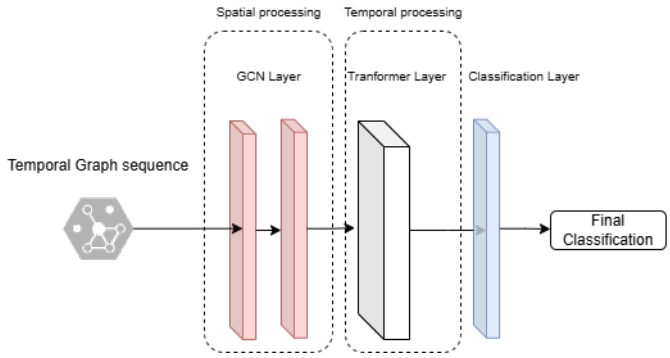

Fig. 1. Architecture of the proposed jamming detection model.

A. Jamming Detection Technique

Graph neural networks (GNNs) are neural models that capture the dependence of graphs via message passing between the nodes of graphs. In recent years, variants of GNNs such as graph convolutional network (GCN), graph attention network (GAT), and graph recurrent network (GRN) have demonstrated ground-breaking performances on many deep learning tasks [18]. We propose a novel jamming detection framework to identify jamming attacks in wireless networks, based on the dynamic graph representation. The architecture leverages the dynamic graph $\{G(n)\}$, graph convolution, and attention mechanisms to capture the distinctive patterns of connectivity disruptions caused by signal jammers.

The proposed jamming detection system consists of:

1) **Feature Extraction**, Each static graph $G(n)$ is constructed from the network topology and connectivity data between nodes.
2) **Spatial processing module (GCN layer):** Utilizes two stacked Gated Graph Convolutional layers to process each network snapshot independently and extract meaningful node-level representations (embeddings).
3) **Temporal processing module (Transformer layer):** Applies a multi-head self-attention mechanism across a sequence of graphs to detect temporal patterns that are indicative of jamming.
4) **Classification module:** Outputs a binary decision indicating whether the input sequence contains a jamming attack.

Fig. 1 illustrates the overall architecture. The system processes sequences of K network graphs $\mathcal{G}(t) = \{G(t), G(t+1), \ldots, G(t+K-1)\}$, where each sequence represents a specific network condition over time, to provide a binary decision on whether jamming activity is present within the sequence.

The system processes a sequence of N_{steps} consecutive network graphs:

$$\mathcal{G}(n) = \{G(n), G(n+1), \ldots, G(n+N_{\text{steps}}-1)\},$$

where each $G(n)$ represents the state of the wireless network at time nT.

Feature Extraction and Graph Construction: Each static graph $G(n)$ is constructed from the real-time network topology and connectivity data. From each graph, we extract the following features:

- **Node-level features:**
  - *Degree centrality* $d_v(n)$: the number of connections of node v at time n;
  - *Node type* $\tau_v \in \{0, 1\}$: where 0 denotes Access Points (APs) and 1 denotes User Equipments (UEs);
  - *Position coordinates* $(x_v(n), y_v(n))$: the physical location of node v in 2D space.
- **Edge-level features:**
  - *SINR* $\Gamma_{u,v}(n)$: the signal-to-interference-plus-noise ratio between nodes u and v;
  - *Distance* $d_{u,v}(n)$: Euclidean distance between nodes u and v, computed as:

$$d_{u,v}(n) = \sqrt{(x_u(n) - x_v(n))^2 + (y_u(n) - y_v(n))^2}.$$

These features are extracted from the dynamic graph object, which stores node types, positions, and connection weights between APs and UEs. After conducting ablation experiments by selectively removing features and measuring the resulting performance, we found the above features to be the most critical for detecting jamming events.

After experimenting with removing features and measuring performance degradation, the above-mentioned features are the most impactful for the jamming detection process.

Spatial Processing Module: The extracted node and edge features are fed into a Graph Neural Network to compute node embeddings. These embeddings encode both the local structure (who a node is connected to) and attributes (such as position and type). Specifically, for each node v at time n, we compute:

$$h_v(n) = \text{GNN}(G(n), \boldsymbol{\xi}_v(n)),$$

where $\boldsymbol{\xi}_v(n)$ is the feature vector of node v. The GCN aggregates information from neighboring nodes and edges, enabling each node to "learn" a summary of its local neighborhood and behavior.

Temporal Attention and Jamming Classification: The sequence of node embeddings from each graph is passed to a Transformer layer. This layer uses temporal self-attention to identify patterns across time, specifically, it can emphasize graphs that exhibit abnormal behavior (such as sudden drops in SINR or rapid topology changes) and downweight normal periods. This is essential because jamming effects may not be constant but occur intermittently across the sequence.

Classification Module: The final detection is performed by a single linear layer that classifies the aggregated representation. The Transformer outputs a temporal representation $T_o(n)$, which is passed through a fully connected classification layer. The final output is the probability of jamming at the sequence level:

$$p(n) = \text{Softmax}(\text{LayerNorm}(W \cdot T_o(n))). \tag{7}$$

where LayerNorm denotes layer normalization, and $\boldsymbol{W}$ is the weight matrix of the classification layer. The decision is based on whether the probability of the *jammer* class exceeds a fixed threshold. This design allows the model to integrate spatial and temporal information effectively, improving robustness and interpretability in jamming detection

B. Model Training

The model is trained using the cross-entropy loss in a supervised manner using labeled datasets containing examples of nominal and jamming scenarios. During training, sequences of graph snapshots are presented to the model along with binary labels indicating the presence or absence of jamming activity. This supervised approach enables the model to learn discriminative patterns that distinguish normal network fluctuations from intentional jamming interference. The weights are optimized using the Adam optimizer, implementing early stopping when validation performance plateaus.

IV. Numerical Results

A. Dataset Generation

To evaluate the proposed jamming detection approach, we generate a dataset of dynamic network graphs simulating wireless communications with and without jamming interference.

We consider a $L \times L$ area with $L = 1$ km, containing 5 fixed APs and 10 mobile UE nodes. The fixed APs are positioned at strategic locations covering the area: four at the corners, with coordinates $(0.2, 0.2)$, $(0.8, 0.2)$, $(0.2, 0.8)$, and $(0.8, 0.8)$, and one at the center $(0.5, 0.5)$ (all in km unit). Mobile UEs move according to a controlled random walk model with velocity components drawn from a uniform distribution in $[-v_{\max}, v_{\max}]$, where $v_{\max} = 6$ km/h. We consider $T = 1$ s and $T_F = 10$ s. Connectivity between an AP and UE is established when the SINR exceeds the threshold $\Gamma_0 = 5$ dB. The noise power is $\sigma^2 = 0.001$. The jammer affects UEs within 0.35 km radius, and it is located in a different random position for each simulation. The number of network static graphs per sequence $\mathcal{G}(t)$ is $N = 80$.

We analyze two distinct scenarios. In the *deterministic scenario*, we set $\beta = 1$, resulting in a fixed channel matrix $\boldsymbol{h}_{k,m}(n)$. In the *fading scenario*, we set $\beta = 0$, such that $\boldsymbol{h}_{k,m}(n)$ models a Rayleigh fading channel.

B. GNN Implementation

The architecture was implemented using PyTorch and PyTorch Geometric. We used a GCN layer for each snapshot of the dynamic graph that consists of 2 Gated Graph convolution layers with 64 hidden units. The Transformer encoder consists of 4 encoder layers, each with 16 attention heads, 64 hidden units, and a feed-forward dimension of 128. We use GELU activation in the feed-forward networks and apply layer normalization with batch-first processing. Since graph sequences have inherent temporal ordering, we add learned positional encodings to capture temporal relationships. A single linear layer with an intermediate dimension of 32 is used for binary classification. The model was trained for 30 epochs using the Adam optimizer with a learning rate of 1.2×10^{-4}, weight decay of 10^{-6}, and batch size of 8. We applied a dropout of 0.03 in the Transformer layers and 0.05 overall to prevent overfitting. The dataset has 2200 dynamic graphs for each scenario, training was performed on 70% of the dataset, while 10% of the dataset was used for validation and 20% for testing.

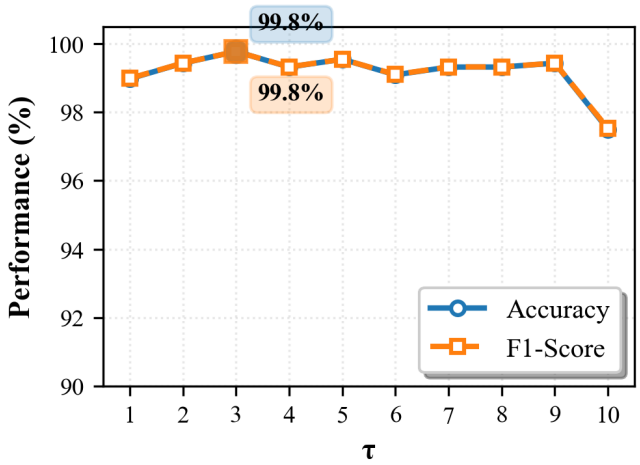

Fig. 2. Accuracy and F1 score vs τ, for the deterministic scenario. Training performed with a dataset having $\tau = 10$.

C. Performance Metrics

Let TP be the number of True Positives, TN be the number of True Negatives, FP be the number of False Positives, and FN be the number of False Negatives. The accuracy is

$$a = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}, \tag{8}$$

F1 score is

$$F_1 = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}}. \tag{9}$$

D. Simulation Results

This section presents a comprehensive experimental evaluation of our dynamic graph-based jammer detection system under two primary training scenarios: (1) mixed-τ training using data from all jammer persistence patterns $\tau \in \{1, 2, ..., 10\}$, and (2) $\tau = 10$ specialist training using only continuous jammer scenarios. The parameter τ represents the jammer activation frequency within each temporal sequence, where $\tau = 1$ indicates sporadic jamming (active for only 1 out of 10 timesteps), $\tau = 5$ represents moderate persistence (active for 5 out of 10 timesteps), and $\tau = 10$ denotes continuous jamming (active throughout the entire sequence). All experiments were conducted with 80-timestep sequences on cell-free MIMO networks, evaluating performance under both fading and non-fading channel conditions.

E. $\tau = 10$ Specialist Training Analysis

The $\tau = 10$ specialist results under non-fading conditions, shown in Fig. 2, achieved accuracy consistently above 99%

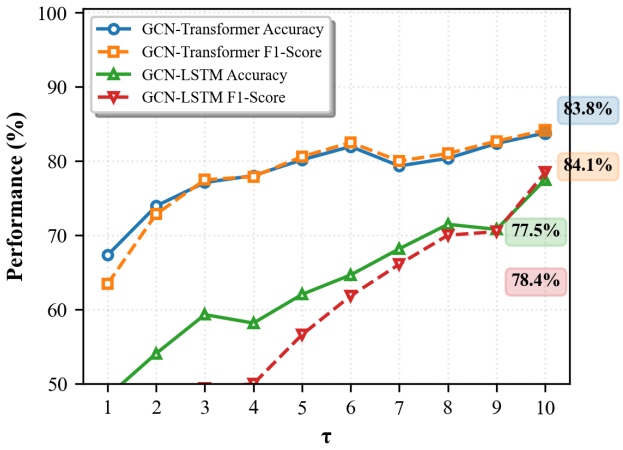

Fig. 3. Accuracy and F1 score vs τ, for the fading scenario. Training performed with a dataset having $\tau = 10$.

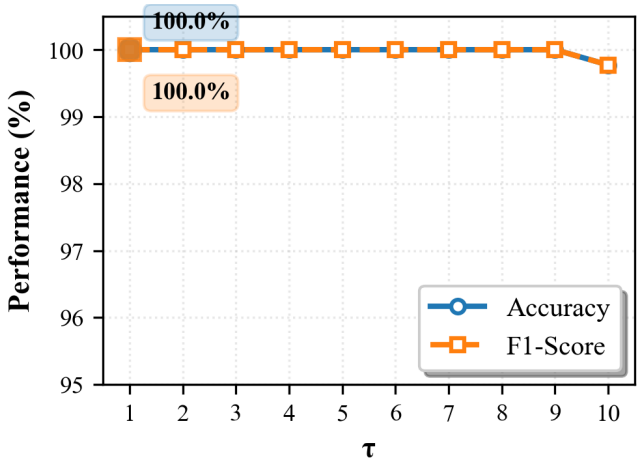

Fig. 4. Accuracy and F1 score vs τ, for the deterministic scenario. Training performed with a dataset having a mixture of attacks with different values of τ.

across $\tau = 1 - 9$, and F1-scores reaching 99.8% at $\tau = 3$. However, a notable performance degradation occurs at $\tau = 10$, where accuracy drops to 97.1% and F1-score to 97.4%. This indicates that training exclusively on continuous jammer scenarios, counterintuitively, provides excellent generalization to sporadic and rhythmic jamming patterns under non-fading channels.

In contrast, the fading scenario, shown in Fig. 3, reveals the specialist's true generalization limitations and more pronounced performance variations. While maintaining strong overall performance (accuracy range: 67.2%-83.8%), the model shows increased sensitivity to jammer persistence patterns, however, a comparison has been done on the same dataset using the known Long Short Term Memory GCN (GCN-LSTM) [19] which combines the capabilities of LSTMs to extract temporal dependencies with the feature learning power of the GCN, and as the figure shows, our model performed better in all projected jamming behaviours. The performance progression from $\tau = 1$ (67.2% accuracy) to $\tau = 8$ (83.8% accuracy) demonstrates the model's adaptation to different temporal structures, with optimal detection occurring in the rhythmic jamming domain ($\tau = 6 - 8$).

F. Mixed-τ Training Performance

Fig. 4 shows the performance of our mixed-τ training approach under non-fading channel conditions. The model exhibit 100% accuracy across $\tau = 1 - 9$, with minimal degradation to 99.7% at $\tau = 10$.

In the fading scenario, presented in Fig. 5, the obtained accuracy ranges from 75.6% at $\tau = 1$ to 89.7% at $\tau = 8$, before decreasing to 79.4% at $\tau = 10$. The monotonic improvement from $\tau = 1$ to $\tau = 8$ (73.2% to 89.5% F1-score) suggests that the model learns increasingly effective detection strategies as jammer persistence increases, until reaching the domain boundary at $\tau = 9 - 10$.

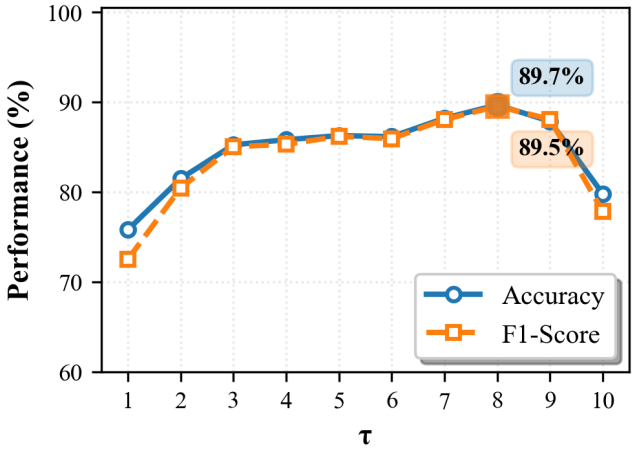

Fig. 5. Accuracy and F1 score vs τ, for the random fading scenario. Training performed with a dataset having a mixture of attacks with different values of τ.

G. Channel Fading Effects on Detection Performance

Comparing non-fading versus fading scenarios reveals significant differences in detection robustness. Under non-fading conditions, both training strategies achieve near-perfect performance across most τ values, suggesting that the absence of channel fading provides cleaner signal characteristics that enhance jammer detection reliability. The stable channel conditions appear to preserve jamming signatures without additional noise from natural channel variations.

Conversely, fading scenarios present more challenging detection environments, with performance variations of 10-15 percentage points across different τ values. This increased difficulty under fading channels indicates that channel-induced signal variations may mask jamming signatures, requiring more sophisticated detection algorithms to distinguish between fading-induced and jammer-induced signal degradations.

H. Training Strategy Effectiveness Comparison

The mixed-τ training approach demonstrates improved generalization capabilities and overall performance compared to the $\tau = 10$ specialist across both channel conditions. Under non-fading conditions, mixed-τ training achieves near-perfect performance (more than 99% accuracy) across the entire τ spectrum, while under fading conditions, it maintains reasonable performance levels (76-90% range) with more graceful degradation patterns. In contrast, the $\tau = 10$ specialist, despite showing perfect performance under non-fading conditions, exhibits significant generalization limitations under fading scenarios, with performance dropping as low as 67% at $\tau = 1$.

The mixed-τ approach's enhanced robustness across different channel conditions and jammer persistence patterns indicates that exposure to diverse jamming behaviors during training provides more generalizable feature representations. This finding supports the hypothesis that multi-domain training strategies are essential for robust jammer detection in dynamic wireless environments.

I. Baseline Shift Problem in Persistent Jamming

The performance degradation observed at $\tau = 10$ across all experimental configurations can be attributed to the fundamental baseline shift problem in persistent jamming scenarios. When jammers operate continuously, cell-free MIMO networks undergo adaptive responses. These network adaptations effectively establish a new operational baseline where continuous interference becomes the "normal" state.

V. Conclusions

This paper presented a comprehensive analysis of jammer detection in cell-free MIMO networks using dynamic graphs and specific graph neural network architecture, revealing insights into the effect of channel fading in the jamming detection process, and the multi-domain nature of temporal anomaly detection, in addition to this, our experimental evaluation across different jammer patterns ($\tau \in \{1, 2, ..., 10\}$) demonstrated that mixed-τ training achieves enhanced generalization compared to specialist approaches, with performance exceeding 99% under non-fading conditions and maintaining robustness above 75.6% even in challenging fading scenarios, higher than existing known models. The comparative analysis between fading and non-fading channels revealed that stable channel conditions significantly enhance detection reliability, while channel fading introduces additional complexity that degrades performance by 10-15 percentage points across all τ values.

A nice finding of this work is the identification of the baseline shift problem in persistent jamming scenarios ($\tau = 9-10$), where continuous jammer presence causes network adaptation responses that establish a new operational baseline, making traditional anomaly detection approaches ineffective. This phenomenon explains the characteristic performance degradation observed at high τ values across all experimental configurations, highlighting the need for detection strategies that can identify adaptation artifacts rather than direct interference signatures. The delineation of three distinct detection domains, namely sporadic ($\tau = 1 - 3$), rhythmic ($\tau = 4 - 8$), and persistent ($\tau = 9 - 10$), provides a theoretical framework for developing domain-specific architectures that address the unique challenges of each jammer behavior pattern.

References

[1] H. Q. Ngo, A. Ashikhmin, H. Yang, E. G. Larsson, and T. L. Marzetta, "Cell-free massive MIMO versus small cells," *IEEE Trans. on Wireless Commun.*, vol. 16, no. 3, pp. 1834–1850, Mar. 2017.

[2] H. Pirayesh and H. Zeng, "Jamming attacks and anti-jamming strategies in wireless networks: A comprehensive survey," *IEEE Commun. Surveys & Tutorials*, vol. 24, no. 2, pp. 767–809, Mar. 2022.

[3] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM computing surveys (CSUR)*, vol. 41, no. 3, pp. 1–58, Sep. 2009.

[4] Y. Zhao, Z. Nasrullah, and Z. Li, "Deep learning for anomaly detection: A review," *ACM Computing Surveys (CSUR)*, vol. 54, no. 12, pp. 1–38, Mar. 2021.

[5] H. Akhlaghpasand, S. M. Razavizadeh, E. Björnson, and T. T. Do, "Jamming detection in massive MIMO systems," *IEEE Wireless Commun. Letters*, vol. 7, no. 2, pp. 242–245, Apr. 2018.

[6] P. Lohan, B. Kantarci, M. Amine Ferrag, N. Tihanyi, and Y. Shi, "From 5G to 6G networks: A survey on AI-based jamming and interference detection and mitigation," *IEEE Open Jour. of the Commun. Society*, vol. 5, pp. 3920–3974, Jun. 2024.

[7] Y. Li, J. Pawlak, J. Price, K. Al Shamaileh, Q. Niyaz, S. Paheding, and V. Devabhaktuni, "Jamming detection and classification in OFDM-based UAVs via feature- and spectrogram-tailored machine learning," *IEEE Access*, vol. 10, pp. 16 859–16 870, 2022.

[8] C. Zhang, P. Patras, and H. Haddadi, "Deep learning in mobile and wireless networking: A survey," *IEEE Commun. Surveys & Tutorials*, vol. 21, no. 3, pp. 2224–2287, Mar. 2019.

[9] M. Varotto, S. Valentin, F. Ardizzon, S. Marzotto, and S. Tomasin, "One-class classification as GLRT for jamming detection in private 5g networks," in *Proc. Int. Work. on Signal Processing Advances in Wireless Commun. (SPAWC)*, 2024, pp. 201–205.

[10] M. Varotto, F. Heinrichs, T. Schürg, S. Tomasin, and S. Valentin, "Detecting 5G narrowband jammers with CNN, k-nearest neighbors, and support vector machines," in *Proc. IEEE Int. Work. on Information Forensics and Security (WIFS)*, 2024, pp. 1–6.

[11] M. Varotto, S. Valentin, and S. Tomasin, "Detecting 5G signal jammers using spectrograms with supervised and unsupervised learning," in *Proc. IEEE Int. Conf. on Commun. Work. (ICC Work.)*, 2024, pp. 767–772.

[12] ——, "Detecting 5G signal jammers with autoencoders based on loose observations," in *Proc. IEEE Globecom Work. (GC Wkshps)*, 2023, pp. 160–165.

[13] B. Upadhyaya, S. Sun, and B. Sikdar, "Machine learning-based jamming detection in wireless IoT networks," in *Proc. IEEE VTS Asia Pacific Wireless Commun. Symposium (APWCS)*, 2019, pp. 1–5.

[14] N. I. Mowla, N. H. Tran, I. Doh, and K. Chae, "Federated learning-based cognitive detection of jamming attack in flying ad-hoc network," *IEEE Access*, vol. 8, pp. 4338–4350, 2020.

[15] J. Skarding, B. Gabrys, and K. Musial, "Foundations and modeling of dynamic networks using dynamic graph neural networks: A survey," *IEEE Access*, vol. 9, pp. 79 143–79 168, 2021.

[16] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *IEEE Trans. on Neural Networks and Learning Systems*, vol. 32, no. 1, pp. 4–24, Jan. 2021.

[17] S. Elhoushy, M. Ibrahim, and W. Hamouda, "Cell-free massive MIMO: A survey," *IEEE Commun. Surveys & Tutorials*, vol. 24, no. 1, pp. 492–523, Apr. 2022.

[18] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, "Graph neural networks: A review of methods and applications," *AI Open*, vol. 1, pp. 57–81, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2666651021000012

[19] L. García-Duarte, J. Cifuentes, and G. Marulanda, "Short-term spatio-temporal forecasting of air temperatures using deep graph convolutional neural networks," *Stochastic Environmental Research and Risk Assessment*, vol. 37, no. 5, p. 1649–1667, Dec 2022.