Comparative Analysis of Deep Learning Models for Detecting Jamming Attacks in Wi-Fi Network Data

Fatima tu Zahra
VeNIT Lab.
Marmara University
Istanbul, Turkey
Fatima.zahra@venit.org

Yavuz S. Bostanci
VeNIT Lab.
Dept. of Computer Engineering
Marmara University
Istanbul, Turkey
yavuz.bostanci@venit.org

Mujdat Soyturk
VeNIT. Lab.
Dept. of Computer Engineering
Marmara University
Istanbul, Turkey
mujdat.soyturk@marmara.edu.tr

***Abstract*—Jamming attacks presents a significant challenge to the security and reliability of wireless communication networks, especially in the context of IoT applications. This study introduces a novel approach to detecting jamming attacks by utilizing the upper-layer network parameters from the application and transport layers. An experimental testbed is developed consisting of a Wi-Fi-based IoT server-client application to collect data. The network parameters are gathered from both noiseless and noisy environment conditions to examine the performance variations of different deep-learning models in diverse environments. The performance of various deep learning models is systematically compared, employing evaluation metrics such as accuracy, F1 score, precision, recall, model complexity, and training time. The findings of this research contribute to the development of effective techniques for jamming detection. Moreover, this study provides valuable insights into the selection and adaptation of appropriate models based on system requirements and specifications, enabling efficient detection and mitigation of jamming attacks in wireless communication systems.**

***Index Terms*—IoT, jamming attacks, wireless communication, WiFi, wireless network, deep learning**

I. Introduction

The rapid advancement of wireless communication networks and the widespread adoption of Internet of Things (IoT) applications have benefitted numerous sectors, including smart homes, connected cities, advanced transportation systems, innovative healthcare, modern agriculture, and cutting-edge industries [1]. These applications facilitate system-wide monitoring, task simplification, process automation, and seamless communication among subsystems and devices. However, these developments have also exposed wireless networks to a range of security threats, with jamming attacks being one of the most significant challenges. Jamming disrupts the normal functioning of wireless networks intentionally or unintentionally by generating interference, thereby degrading the network performance and compromising its security [2].

Traditional jamming detection techniques primarily focus on the lower layers such as the physical layer and MAC layer of the protocol stack [3], [4]. The network parameters, such as signal strength, bit error rate, and channel occupancy are used to study the jamming effects [5]. While these approaches can be practical, they may not be sufficiently comprehensive to detect more sophisticated jamming attacks or adapt to the evolving nature of wireless networks [6]. Moreover, most studies rely on simulation data rather than actual network data and typically overlook the implications created by noisy environments [7]. The performance of an IoT system is significantly influenced by noise, which can originate from multiple sources, including other electronic equipment, radio frequency disruptions, and physical barriers due to the interference caused by surface reflections. A high level of noise reduces the signal-to-noise ratio of the incoming signal, detrimentally affecting the system's overall performance. The introduction of jamming attacks compounds this issue, adding further noise and interference to the system. This makes the situation even worse by lowering network throughput, extending delays and bit error rates, and increasing the likelihood of packet drops. Thus, there is a pressing need to explore novel techniques for jamming detection that can provide a more robust and versatile defense against such attacks in both noisy and noiseless environments [8].

This paper introduces an innovative approach to detecting jamming attacks in wireless networks by leveraging the upper-layer network parameters. Upper-layer network parameters are not employed to detect jamming attacks before in the literature. The performance of multiple deep learning models is evaluated and compared for data collected in both noiseless and noisy environments. The findings of this research contribute to the development of efficient techniques for jamming detection and appropriate model selection, ultimately enhancing the overall security of wireless networks.

II. Related Work

The detection of jamming attacks on wireless networks is a well-studied problem, and numerous techniques have been proposed over the years. Traditional approaches rely on lower-layer parameters like signal strength, bit error rate, and channel occupancy to detect jamming attacks. Xu et al. [3] emphasized the importance of robust jamming detection mechanisms in

This work is supported in parts by the InSecTT and Beyond5 projects. InSecTT and Beyond5 have received funding from the Electronic Component Systems for European Leadership Joint Undertaking (ECSEL-JU) and TUBITAK under agreement no. 876038 and 876124 respectively

wireless IoT networks by demonstrating how vulnerable they are to jamming attacks. Jameel et al. [4] conducted a comprehensive survey on cooperative relaying and jamming strategies to enhance physical layer security, addressing the role of cooperative relays in mitigating the impact of jamming. In [9], jamming attacks were detected using parameters like PDR, inter-packet delay, and throughput collected in the monitoring mode of the Wi-Fi network interface card.

However, these lower-layer detection methods may not be adequate to counter sophisticated jamming attacks. Vadlamani et al. [2] underlined this concern, leading to the exploration of detection methods utilizing upper-layer network parameters. In a survey by Mpitziopoulos et al. [7], jamming attacks and countermeasures in wireless sensor networks were discussed, focusing on parameters such as delay, packet drop rate, and throughput at the network and application layers. The researchers in [10] conducted a study on the effects of jamming attacks on network throughput and delay, highlighting the need to monitor these upper-layer parameters. The impact of jamming on TCP performance was analyzed in [11], revealing significant degradation in throughput and an increase in packet retransmission. These works underline the importance of considering upper-layer parameters for comprehensive jamming detection. In [12], the authors presented the effects of jamming attacks on transport and application layer parameters on wireless IoT networks. It is also demonstrated through experimental setup how noisy conditions impacted the wireless communications and jamming attacks have more pronounced effects in the noiseless environment as compared to the noisy environment.

More recently, the application of machine learning and deep learning techniques has been explored for jamming detection and mitigation. It is proved from the literature that machine learning and deep learning algorithms can be extremely effective and highly accurate in detecting jamming attacks in wireless networks [13]. The authors in [14] claim that deep learning models can effectively detect abnormal patterns in network traffic with high accuracy. They evaluated the performance of various deep learning models to detect DDoS attacks and found random forests and CNN to perform best. The paper [15] contributes to the development of trust models for mitigating jamming attacks in IoT networks using deep learning algorithms. In [16], a novel malicious traffic detection method for IoT devices based on an improved TCN called Multi-class S-TCN is proposed. A complete IoT traffic security detection procedure based on deep packet inspection and protocol analysis is implemented using TCN. An end-to-end anti-jamming target detection method based on CNN is proposed in [17]. In [18], the authors implemented two jamming detection systems based on deep CNN and deep RNN. They chose LSTM as their RNN model and found out that LSTM performs better as compared to CNN in detecting network anomalies and jamming attacks. LSTM and GRU models have been extensively used for time series prediction and sequential data modeling due to their ability to handle long sequences effectively [19]. More recently, the Transformer model, introduced by Vaswani et al. [20], has shown remarkable performance in handling sequential data.

Despite these advancements, jamming detection methods still present opportunities for improvement. For instance, the impact of noisy environments on the performance of detection models is frequently disregarded. Moreover, most studies rely on simulated data, which may not accurately represent the complexity of real-world network traffic [21]. The proposed study addresses these gaps by collecting real network data from an IoT system under both noiseless and noisy conditions and systematically comparing the performance of various deep learning models for jamming detection. This systematic comparison aids in choosing the most appropriate model according to the unique needs of the wireless system. Additionally, by taking into account network parameters from both lower and upper layers, the study provides a thorough method for detecting jamming. This holistic approach improves the detection capability, making it more adaptable to evolving wireless network characteristics.

III. Methodology

This section elaborates on the methodology employed in this study, detailing the data collection process, simulation of jamming attacks, data labeling, and the training, evaluation, and comparison of various deep learning models.

A. Data Collection

Data collection constituted the initial step of this research. A wireless IoT server-client application using a Ubuntu server and Raspberry Pi was set up to generate network traffic. The experimental setup is shown in Fig. 1. The application leverages the HTTP protocol for communication, an application layer protocol that depends on the Transmission Control Protocol (TCP) at the transport layer to ensure reliable, sequenced, and error-checked data transmission. The server and client machines recurrently exchange data packets across a wireless network using IEEE 802.11n protocol (also known as WiFi 4), This standard operates on 2.4GHz frequency, and our specific IoT setup was operating on channel 11 (2.462GHz), mimicking the standard communication flow seen in real-world IoT systems. The distance between the devices and the jammer's location was kept constant throughout all the experiments to maintain consistency in the experimental setup and minimize extraneous factors that could potentially introduce variability in our data. By doing so, we aimed to isolate the effects of jamming and ensure that any changes observed in the network parameters were due to the presence or absence of jamming, rather than other variables such as fluctuations in signal strength due to distance changes.

Two distinct datasets were collected under different environmental conditions: one in a noiseless environment and the other in a noisy environment with the same experimental setup and configuration. In these experiments, the "noiseless" environment is defined as one where minimal interference exists, with no additional Wireless devices or physical obstacles present. In contrast, a "noisy" environment was established

with additional interference, such as other Wireless devices or potential radio frequency interference from various electronic devices. This strategy was employed to investigate the models' capability to detect jamming attacks under diverse network conditions.

Throughout the experiments, network traffic from the client machine was recorded in *pcap* files using a network sniffing tool. Network performance metrics were extracted from these pcap files to train the models. The metrics used in this study include the no. of packets transmitted and received during communication, network throughput, HTTP delay at the application layer, TCP Round Trip Time (RTT), and, TCP retransmissions at the transport layer as research have established the adverse impact of jamming on these parameters in both noisy and noiseless environments [12]. These parameters were later employed to train the deep learning models to identify jamming attacks.

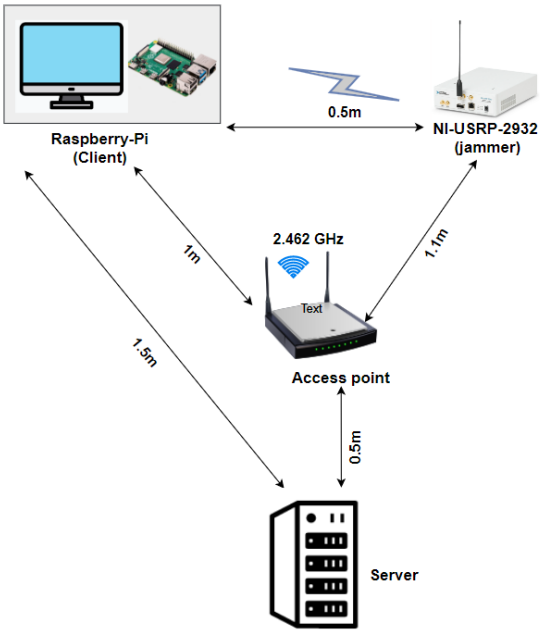

Fig. 1. Experimental setup

B. Simulation of Jamming Attacks

The Universal Software Radio Peripheral (USRP) 2932 along with the gnu radio application was utilized to simulate various types of jamming attacks on the wireless network. The study implemented four distinct types of jamming attacks: constant, random, barrage, and pulse jamming attacks as displayed in Fig. 2 during the experiments while collecting data.

Across all experiments, the jamming power and distance were held constant. We set the jammer's power to influence the system but not completely block the communication. Specifically, the maximum power of jamming signals was set to 10 dBm for both constant and pulse jamming attacks. During random and barrage attacks, the power and amplitude of the jamming signal varied, but the maximum power remained at 10 dBm. This level was chosen to ensure the continuity of communication between IoT devices, as any power exceeding 10 dBm risked breaking the connectivity.

All the jamming signals implemented in this study have a central frequency of 2.462 GHz and 20 MHz bandwidth except for the barrage attack where the bandwidth was set to be 40 MHz. Each type of attack was executed under both noiseless and noisy conditions. The data with various jamming attacks were gathered over a two-hour time period, sampling the network parameters at a frequency of one sample per second. Both the noiseless and noisy datasets were designed to be similar in terms of the nature and duration of the attacks. Additionally, the number of data points collected and the period of data collection were also kept consistent. The same dataset was used to train all the deep-learning models in this study. This strategic alignment was crucial in maintaining consistency in the data, thereby strengthening the accuracy of the results. This approach also ensured uniformity across the datasets, thereby enhancing the reliability of the study. Several experiments were carried out, generating an ample amount of data to effectively train the deep learning models.

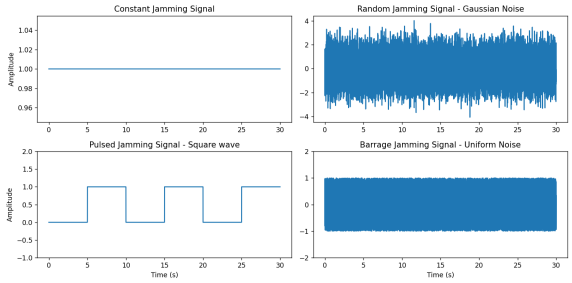

Fig. 2. Jamming signals used in the experiments

C. Data Pre-Processing and Labeling

Following data collection and jamming simulation, the data was cleaned and labeled. Each data instance was categorized as either normal (no jamming) or abnormal (jamming present). The data was labeled using a boolean. Extreme care was taken while preprocessing to ensure a balanced dataset, preventing skewness in representation and enhancing model training effectiveness. The labeling in the dataset was binary - indicating the presence or absence of jamming - rather than specifying the type of jamming attack. Given this specific labeling scheme, this analysis mainly assessed the performance of different deep learning models in identifying whether a jamming attack was taking place, rather than differentiating or classifying among various types of attacks.

D. Training and Evaluation of Deep Learning Models

In this study, all deep learning models were trained using the same set of network performance metrics extracted from pcap files recorded during the experiment. These metrics include the number of packets transmitted and received, network throughput, HTTP delay at the application layer, TCP Round

Trip Time (RTT), and TCP retransmissions. These parameters were chosen based on existing research [12] highlighting their sensitivity to jamming attacks in both noisy and noiseless environments. The labeled dataset was utilized to train five different deep-learning models, including CNN, TCN, LSTM, GRU, and the Transformer model.

To maintain consistency and ensure unbiased comparison, all models were single-layered and were individually trained on both noiseless and noisy datasets under identical conditions. This included an 80:20 data split for training and testing, with a random state set to 42, guaranteeing the same data distribution for training and evaluation across all models. Every model was trained for 50 epochs using a batch size of 32, employing the Rectified Linear Unit (ReLU) activation function. The models were compiled using the Adam optimizer and employed the Mean Squared Error (MSE) as the loss function. The training and validation loss for all models for both datasets is displayed in Fig. 3. It can be observed from the figure that the training and validation loss curves stability is less and magnitude is higher for noisy data. This shows that noise adds extra variability that the model needs to account for, which can slow down the learning process and increase the loss value.

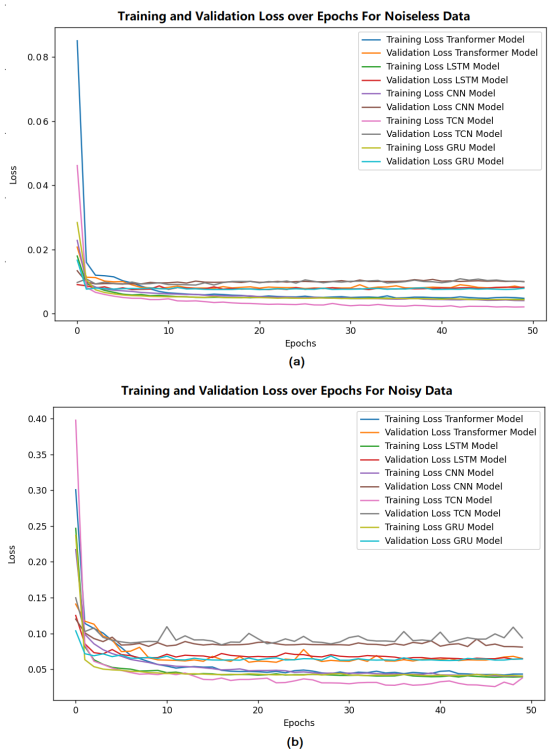

Fig. 3. Training and validation loss of all models **(a)** Models trained on Noiseless Dataset **(b)** Models trained on Noisy Dataset

By using this consistent methodology across all models, we ensured that any differences in performance could be attributed to the model architectures themselves, rather than differences in training conditions. This rigorous approach allows us to compare the efficiency, accuracy, and complexity of different model architectures with confidence. Post-training, the models were evaluated based on their ability to detect jamming attacks from the respective datasets.

IV. Results and Evalution

The deep learning models - TCN, CNN, LSTM, GRU, and Transformer - were trained using the collected data sets separately from both noiseless and noisy environments, and their performance was evaluated. The evaluation metrics used for comparing the performance of these models included accuracy, F1 score, precision, recall, model complexity in terms of training parameters, and training time of the model. The application of these metrics to both the noiseless and noisy datasets allowed for a detailed comparative analysis of each model's performance under different network conditions.

A. Model Performance on Noiseless Dataset

The performance of the models on the noiseless test data is summarized in Table I.

In examining the performance of the deep learning models on noiseless data, the LSTM model stands out with the highest accuracy of 95.78%, precision of 92.52%, recall of 97.37%, and an F1 score of 94.84%. It demonstrates superior performance among other models despite its moderate model complexity and training time. The CNN model, while having the lowest model complexity and shortest training time, still manages to achieve a respectable accuracy of 90.36% demonstrating its efficiency. The TCN model, despite having the highest model complexity of 23,233 parameters, does not outperform the other models, indicating that a higher number of parameters does not necessarily lead to better performance. GRU stands after LSTM in performance. Both the GRU and Transformer models show similar performance metrics, balancing accuracy and complexity well.

B. Model Performance on Noisy Dataset

The performance analysis of the models using the noisy dataset is presented in Table II.

Upon analyzing the performance of the deep learning models on noisy data, the LSTM model once again proves to be the most robust, achieving an accuracy of 90.88%, the highest among all models. The CNN model, with its low complexity and short training time, still manages to achieve an accuracy of 83.13%. On the contrary, the TCN model, despite its high complexity, has the lowest accuracy and F1 score, suggesting that it may be less robust to noise compared to the other models.

C. Model Performance Evaluation and Comparison

A comparative analysis of the performance of deep learning models on noiseless and noisy datasets shows evident differences in accuracy, precision, recall, F1 Score, and training time. The bar chart showing the accuracy of models for both noiseless and noisy data sets is presented in Fig. 4. As expected, all models perform better on noiseless data, with the LSTM model achieving the highest accuracy of 95.78%

TABLE I
PERFORMANCE OF THE DEEP LEARNING MODELS ON THE NOISELESS DATA

Model	Accuracy (%)	Precision(%)	Recall(%)	F1 Score (%)	Training Time(s)	Model Complexity
TCN	91.16	89.22	89.22	89.22	10.86	23233
CNN	90.36	88.24	88.24	88.24	**3.72**	**1217**
LSTM	**95.78**	**92.52**	**97.37**	**94.84**	6.83	12451
GRU	94.58	90.99	97.12	93.95	6.04	9501
Transformer	93.17	89.72	94.12	91.87	8.46	3618

TABLE II
PERFORMANCE OF THE DEEP LEARNING MODELS ON THE NOISY DATA

Model	Accuracy (%)	Precision(%)	Recall(%)	F1 Score (%)	Training Time(s)	Model Complexity
TCN	81.93	80.37	78.18	79.26	11.10	23233
CNN	83.13	80.36	81.82	81.08	**4.01**	**1217**
LSTM	**90.88**	88.75	**89.59**	**87.17**	7.18	12451
GRU	87.57	84.35	88.18	86.22	6.47	9501
Transformer	86.75	**89.69**	79.09	84.06	8.92	3618

compared to 90.88% on noisy data. In the case of the TCN model, the performance drops significantly when introduced to noisy data, with an accuracy decrease from 91.16% to 81.93%. The CNN, GRU, and Transformer models also exhibit a decrease in performance when moving from noiseless to noisy data as presented in Fig. 4.

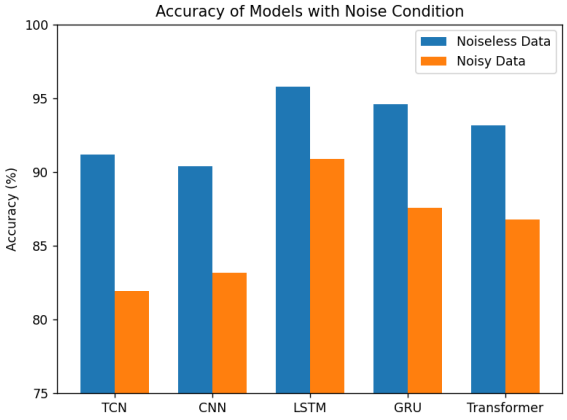

Fig. 4. Accuracy of models with noiseless and noisy dataset

The comparison of model training time is exhibited in Fig. 5 and the model complexity is presented in Fig. 6. The model complexity which is defined by the number of trainable parameters, is determined by the model's architecture and not the data it's trained on. Hence, it remains the same for both noisy and noiseless datasets.

It is observed from the figures that TCN has the highest trainable parameters and the highest training time. This suggests that TCN, despite having a more complex architecture, may not be as robust to noise as the other models. TCN stands out to be the most unsuitable deep learning model for jamming detection in wireless network traffic according to the analysis. Also, CNN proved to be the fastest model to train with the least complexity which makes it a good choice for scenarios where computational resources or time are constraints. The concern with complexity starts to matter

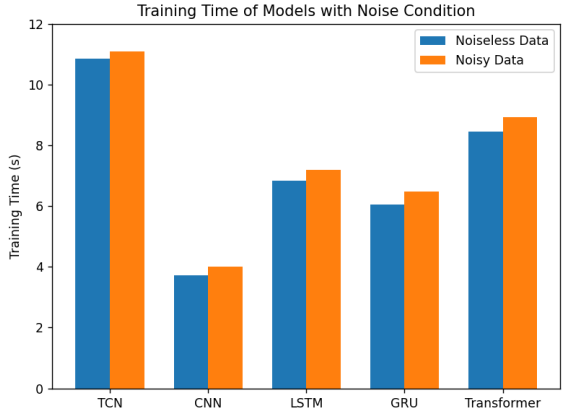

Fig. 5. Training time of models with noiseless and noisy dataset

primarily when computational resources become a limiting factor. This often occurs in real-time applications, embedded systems, or when the model is expected to run on resource-constrained devices, such as microcontrollers, or IoT devices with limited processing capabilities and power. The LSTM model with its superior performance and resilience to noise, makes it a strong candidate for real-world applications where noise is often unavoidable. The GRU and Transformer models have a balanced trade-off between performance metrics and model complexity which remained consistent across both data types.

Overall, this comparative analysis underscores the importance of evaluating deep learning models under different conditions, particularly in scenarios where noise may be present. It highlights the varying degrees of robustness of these models and the need to carefully consider the specific requirements and constraints of the application when selecting the appropriate model.

V. CONCLUSION

This paper presented a novel approach for detecting jamming attacks in wireless networks using deep learning models

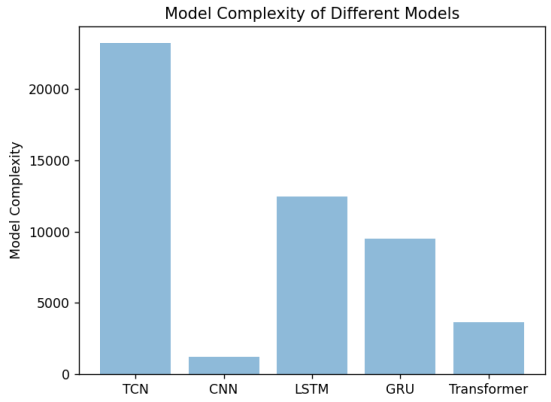

Fig. 6. Complexity of Models (Trainable Parameters)

trained on upper-layer network parameters. An experimental setup was built to collect data from a wireless IoT system in both noiseless and noisy environmental conditions. Various jamming attacks were performed while collecting network data using an SDR. The comparative analysis of five different deep learning models including TCN, CNN, LSTM, GRU, and Transformer model was conducted to evaluate their performance regarding jamming detection. LSTM emerged as the most robust model closely followed by GRU, demonstrating superior performance on both noiseless and noisy datasets. The outcome of this research contributes significantly to the development of effective strategies for jamming detection in wireless networks. By employing deep learning models trained on upper-layer network parameters, we have expanded the scope of traditional detection techniques, making them more adaptable and robust in the face of various network conditions and types of jamming attacks.

This study concludes that deep learning models offer flexibility, adaptability, and improved performance in jamming detection. However, ML isn't necessarily the only solution; The choice between ML and non-ML approaches should be guided by the specific requirements and constraints of the system, including factors like computational resources, latency requirements, the diversity of potential threats, and the need for adaptability. Future work could potentially involve refining the labeling in our dataset to identify specific types of attacks, which would enable a more detailed analysis of the performance of each model against different types of jamming. It is also aimed to optimize these models further to reduce computational complexity without sacrificing performance. Furthermore, to validate the practical applicability of these models, their implementation and assessment in real-world scenarios are also planned.

REFERENCES

[1] W. Ejaz et al., "Internet of Things (IoT) in 5G Wireless Communications," in IEEE Access, vol. 4, pp. 10310-10314, 2016, doi: 10.1109/ACCESS.2016.2646120.

[2] S. Vadlamani, B. Eksioglu, H. Medal and A. Nandi, "Jamming attacks on wireless networks: A taxonomic survey," in International Journal of Production Economics, vol. 172, pp. 76-94, 2016.

[3] W. Xu, W. Trappe, Y. Zhang, and T. Wood, "The Feasibility of Launching and Detecting Jamming Attacks in Wireless Networks," in Proceedings of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing, 2005, pp. 46-57.

[4] F. Jameel, S. Wyne, G. Kaddoum and T. Q. Duong, "A Comprehensive Survey on Cooperative Relaying and Jamming Strategies for Physical Layer Security," in IEEE Communications Surveys & Tutorials, vol. 21, no. 3, pp. 2734-2771, third quarter 2019, doi 10.1109/COMST.2018.2865607.

[5] R. Nicole, "Title of paper with only first word capitalized," J. Name Stand. Abbrev., in press.

[6] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," IEEE Trans. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

[7] A. Mpitziopoulos, D. Gavalas, C. Konstantopoulos, and G. Pantziou, "A survey on jamming attacks and countermeasures in WSNs," IEEE Commun. Surv. Tutorials, vol. 11, no. 4, pp. 42-56, 4th Quart., 2009.

[8] Y. Zou, J. Zhu, X. Wang, and L. Hanzo, "A survey on wireless security: Technical challenges, recent advances, and future trends," Proc. IEEE, vol. 104, no. 9, pp. 1727-1765, Sep. 2016.

[9] F. T. Zahra, Y. S. Bostanci and M. Soyturk, "Real-Time Jamming Detection in Wireless IoT Networks," in IEEE Access, vol. 11, pp. 70425-70442, 2023, doi: 10.1109/ACCESS.2023.3293404.

[10] Y. Liu, P. Ning, and M. K. Reiter, "A. El-Keyi, O. Üreten, H. Yanikomeroglu, and T. Yensen, "LTE for public safety networks: Synchronization in the presence of jamming," IEEE Access, vol. 5, pp. 20 800–20 813, 2017," in ACM Transactions on Information and System Security (TISSEC), vol. 14, no. 1, pp. 1-33, 2011.

[11] A. Proaño and L. Lazos, "Selective Jamming Attacks in Wireless Networks," 2010 IEEE International Conference on Communications, Cape Town, South Africa, 2010, pp. 1-6, doi: 10.1109/ICC.2010.5502322.

[12] F. T. Zahra, Y. S. Bostanci, and M. Soyturk, "The Consequences of Jamming Attacks on IoT Networks: Evaluating the Performance Metrics in Noiseless and Noisy Environments," in 31st IEEE Conference on Signal Processing and Communications Applications, Istanbul, Turkey, 2023, pp. 1-4.

[13] E. Jayabalan and R. Pugazendi, "Deep learning model-based detection of jamming attacks in low-power and lossy wireless networks," in Soft Computing, vol. 26, pp. 12893-12914, 2021, doi: 10.1007/s00500-021-06111-7.

[14] B. Susilo and R. F. Sari, "Intrusion Detection in IoT Networks Using Deep Learning Algorithm," Information, vol. 11, no. 5, p. 279, May 2020, doi: 10.3390/info11050279.

[15] M. S. Abdalzaher, M. Elwekeil, T. Wang and S. Zhang, "A Deep Autoencoder Trust Model for Mitigating Jamming Attack in IoT Assisted by Cognitive Radio," in IEEE Systems Journal, vol. 16, no. 3, pp. 3635-3645, Sept. 2022, doi: 10.1109/JSYST.2021.3099072.

[16] X. Liu, Z. Liu, Y. Zhang, W. Zhang, D. Lv, and Q. Zhou, "TCN enhanced novel malicious traffic detection for IoT devices," Connection Science, vol. 34, no. 1, pp. 1322-1341, 2022.

[17] Y. Zhang, B. Jiu, P. Wang, H. Liu and S. Liang, "An End-to-End Anti-Jamming Target Detection Method Based on CNN," in IEEE Sensors Journal, vol. 21, no. 19, pp. 21817-21828, 1 Oct.1, 2021, doi: 10.1109/JSEN.2021.3103042.

[18] S. Gecgel, C. Goztepe, and G. Karabulut Kurt, "Jammer Detection based on Artificial Neural Networks: A Measurement Study," 2019 IEEE VTS Asia Pacific Wireless Communications Symposium (APWCS), Singapore, Singapore, 2019, pp. 43-48, doi: 10.1145/3324921.3328788.

[19] R. Cahuantzi, X. Chen and S. Güttel, "A comparison of LSTM and GRU networks for learning symbolic sequences," arXiv preprint arXiv:2107.02248, 2021.

[20] A. Vaswani et al., "Attention is all you need," in Advances in Neural Information Processing Systems, 2017, pp. 5998-6008. [Online]. Available: https://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf

[21] Y. Arjoune, F. Salahdine, M. S. Islam, E. Ghribi and N. Kaabouch, "A Novel Jamming Attacks Detection Approach Based on Machine Learning for Wireless Communication," 2020 International Conference on Information Networking (ICOIN), Barcelona, Spain, 2020, pp. 459-464, doi: 10.1109/ICOIN48656.2020.9016462.