Viksit India
2024 | Volume: 02 | Issue: 04 | Article: 39

Antijamming Signal Detection Using Convolutional Neural Networks in Wireless Communications

Anjali Sharma
National Institute of Technology Kurukshetra

Article Info

Received: 5th June 2023
Revised: 30th December 2023
Published: 31th December 2024

*Corresponding author

Email: brahmjit.s@gmail.com

Open Access

DOI:

https://pub.dhe.org.in/
ISSN: 2278-1757

Abstract

This paper presents an in-depth exploration of jamming detection in wireless communication systems using Convolutional Neural Networks (CNNs). Jamming attacks pose a significant threat to wireless networks, particularly in environments where secure communication is critical. Traditional detection mechanisms often fall short due to their limited adaptability and reliance on predefined heuristics. CNNs offer a robust solution through automatic pattern recognition in signal data. This study designs and evaluates a CNN model trained on a simulated dataset of multiple jamming types. Results show that the model achieves high classification accuracy and robustness, supporting its use in real-world applications. This paper details the data generation, model architecture, training methods, experimental setup, evaluation results, and directions for future research.

Keywords: Convolutional Neural Networks.

Introduction

Wireless communication has become the backbone of modern technological infrastructure, enabling applications ranging from personal communication to industrial automation and national security operations. The flexibility and convenience of wireless systems have accelerated their adoption in mobile phones, Wi-Fi networks, satellite communications, and the Internet of Things (IoT). However, this widespread usage has simultaneously exposed wireless networks to various vulnerabilities. Among these, jamming attacks—intentional attempts to disrupt communication by overwhelming a signal with noise or false information—pose one of the most significant threats. Unlike accidental interference, jamming is deliberate and malicious, making it a serious concern for military, commercial, and emergency communication systems.

Jamming disrupts the communication channel by transmitting signals that interfere with legitimate transmissions, leading to degraded performance or complete denial of service (DoS). Common jamming techniques include constant jamming, where a signal is continuously transmitted to block legitimate communication; reactive jamming, which is triggered in response to detected transmissions; and random jamming, which intermittently interferes with signals in an unpredictable manner. These tactics can have far-reaching implications, such as disabling emergency communication networks, compromising military missions, or causing significant financial loss to businesses reliant on wireless systems.

Traditional methods for detecting jamming rely heavily on hardware-based approaches or threshold-based algorithms that monitor signal strength and noise levels. While effective in controlled environments, these approaches often lack adaptability and scalability. They may struggle to detect sophisticated or low-power jamming attempts and are frequently ineffective in dynamic or cluttered communication environments. Additionally, the rigid nature of these solutions makes them unsuitable for real-time detection in mobile or heterogeneous network conditions.

The advent of machine learning (ML) and deep learning (DL) has introduced a paradigm shift in signal processing and network security. These techniques can automatically learn patterns from large datasets, enabling the identification of subtle and complex anomalies that traditional algorithms may miss. Deep learning models, especially Convolutional Neural Networks (CNNs), have shown exceptional performance in fields such as computer vision and natural language processing. CNNs are particularly well-suited for analyzing time-series data due to their ability to capture spatial and temporal correlations, making them ideal candidates for jamming detection tasks.

In this paper, we explore the application of CNNs to detect various types of jamming in wireless communications. We propose a model trained on a dataset of simulated signals representing normal and jammed transmission scenarios. The CNN processes raw signal inputs and classifies them into one of several categories based on the presence and type of jamming. The use of dropout layers, regularization, and validation strategies ensures the model generalizes well to unseen data, addressing the issue of overfitting common in deep learning systems.

Our proposed model is designed with practical implementation in mind. It is computationally efficient enough to be integrated into existing wireless infrastructures and can be deployed on edge devices for real-time detection. Moreover, the model's performance on validation and test datasets shows high precision, recall, and F1-scores, demonstrating its reliability and effectiveness.

The rest of this paper is structured as follows: Section II reviews existing work on jamming detection and the use of machine learning in signal classification. Section III details the dataset generation, preprocessing, and the CNN model architecture used in our experiments. Section IV outlines the training methodology and performance evaluation metrics. Section V presents the results, including accuracy, confusion matrix analysis, and discussion of model robustness. Finally, Section VI concludes with a summary of findings and directions for future research, including real-world deployment and handling adversarial signal attacks.

Through this research, we aim to contribute a scalable and efficient method for jamming detection, leveraging the power of deep learning to enhance the resilience of wireless communication systems. This is increasingly critical in a world where wireless infrastructure underpins everything from personal convenience to national security.

II. Related Work

Early methods for jamming detection primarily relied on threshold-based techniques, which monitor signal-to-noise ratio (SNR), received signal strength indicator (RSSI), or packet delivery ratio (PDR) to identify abnormal behavior. For instance, a sudden drop in SNR or a rise in bit error rate (BER) would trigger a jamming alert. Although simple to implement, these methods suffer from high false alarm rates and are often ineffective in complex or dynamic environments where fluctuations in signal parameters can occur for benign reasons. Moreover, they lack the flexibility to distinguish between different types of jamming, making them unsuitable for modern wireless networks that require finer granularity in threat classification.

To improve upon these limitations, researchers began incorporating statistical and probabilistic models. Bayesian inference and Hidden Markov Models (HMMs) were used to predict state transitions between jammed and unjammed states based on temporal signal data. These models introduced temporal awareness and probabilistic reasoning into the detection pipeline, improving the ability to detect reactive or random jamming. However, their performance was still constrained by the need for precise modeling and extensive domain knowledge, limiting their generalization to unseen scenarios.

With the proliferation of software-defined radio (SDR) and cognitive radio (CR) technologies, more sophisticated methods for jamming detection emerged. These approaches leveraged real-time spectrum sensing and dynamic frequency allocation to detect and avoid jamming. While effective, they added complexity to the network and often required additional hardware or tight synchronization among network nodes.

The advent of machine learning marked a turning point in the field. Supervised learning algorithms such as Support Vector Machines (SVMs), Decision Trees, and k-Nearest Neighbors (k-NN) began to be used for classifying signal patterns. These models could be trained on labeled datasets to recognize patterns associated with normal and jammed signals. For example, Wang et al. (2015) applied SVMs to classify jamming types based on features like packet delivery ratio and RSSI variance. Their work demonstrated improved accuracy over traditional threshold-based techniques, especially in differentiating between jamming and interference from environmental factors.

Despite these advances, traditional ML algorithms have inherent limitations when dealing with high-dimensional, raw signal data. They typically require handcrafted feature extraction, which is both time-consuming and prone to bias. Moreover, their capacity to learn complex, hierarchical patterns is limited compared to deep learning models.

Several studies have demonstrated the efficacy of CNNs in jamming detection. Feng et al. (2020) implemented a CNN model trained on spectrogram images generated from time-frequency representations of wireless signals. The model was able to classify different jamming types, such as barrage, sweep, and deceptive jamming, with high accuracy. Similarly, Kang and Lee (2021) proposed a hybrid model combining CNNs with recurrent neural networks (RNNs) to capture both spatial and temporal features, achieving improved detection rates in mobile ad-hoc networks.

Another significant contribution comes from the work of Testi et al. (2022), who compared the performance of several deep learning models—including CNNs, long short-term memory (LSTM) networks, and multilayer perceptrons (MLPs)—on a synthetic jamming dataset. Their results showed that CNNs provided the best trade-off between accuracy and computational efficiency, making them suitable for deployment on edge devices.

In conclusion, the body of research on jamming detection has evolved from simple threshold-based methods to sophisticated deep learning approaches. CNNs have emerged as a particularly promising solution due to their robustness, scalability, and ability to process raw signal data without extensive feature engineering. While challenges such as data availability and model interpretability remain, ongoing advancements in deep learning, hardware acceleration, and wireless technologies are paving the way for practical, real-time jamming detection systems based on CNNs.

III. Methodology

The methodology section outlines the structured approach adopted to develop and evaluate a Convolutional Neural Network (CNN) for detecting jamming signals in wireless communication. The development process encompasses dataset generation, signal preprocessing, model architecture design, training strategy, and evaluation metrics. This systematic approach ensures that each phase contributes effectively to the goal of reliable and accurate jamming signal classification.

A. Dataset Generation

A critical step in developing a supervised machine learning model is the construction of a representative dataset. Since real-world jamming datasets are often unavailable due to security concerns and operational constraints, we opted to generate synthetic data using signal simulation tools. The dataset includes four distinct signal classes: normal, constant jamming, reactive jamming, and random jamming. Each signal was constructed using standard modulation schemes (e.g., BPSK, QPSK) and subjected to varying channel noise levels to simulate realistic wireless environments. Each class contained approximately 1000 samples, each represented as a one-dimensional time series vector of amplitude values.

B. Signal Preprocessing

Raw wireless signals are often noisy and vary significantly in amplitude and frequency characteristics. To ensure consistency, each signal sample was normalized to have zero mean and unit variance. Additionally, one-hot encoding was applied to the class labels to enable multi-class classification using categorical cross-entropy loss. Data was then split into training (70%), validation (15%), and test (15%) sets using stratified sampling to preserve class distribution across subsets.

C. CNN Model Architecture

The core of the detection system is a 1-dimensional Convolutional Neural Network designed to extract spatial patterns within signal sequences. The model architecture consists of the following layers:
- Input Layer: Accepts a 1D signal vector of fixed length (e.g., 256 time steps).
- Convolutional Layer 1: Applies 64 filters with a kernel size of 3 and ReLU activation.
- Max Pooling Layer: Reduces dimensionality by taking the maximum over non-overlapping windows of size 2.
- Dropout Layer: Applies a dropout rate of 0.3 to prevent overfitting.
- Flatten Layer: Converts the 2D output into a 1D vector for the dense layer.
- Dense Layer: Fully connected layer with 100 neurons and ReLU activation.
- Output Layer: Softmax layer with 4 output neurons corresponding to each signal class.

This architecture was chosen for its balance between computational efficiency and classification performance. The convolutional layers enable hierarchical feature extraction, while dropout regularization promotes model generalization.

D. Training Strategy

Training was conducted using the Adam optimizer with an initial learning rate of 0.001. Categorical cross-entropy was used as the loss function due to the multi-class nature of the problem. Training was executed over 100 epochs with early stopping based on validation loss. A batch size of 32 was selected for efficient memory usage.

Model performance was monitored using accuracy and loss metrics on both training and validation sets. Early stopping was configured with a patience of 10 epochs, and the best model weights were restored for final evaluation. The entire training process was implemented using TensorFlow and Keras libraries in Python.

E. Evaluation Metrics

To thoroughly evaluate the model, multiple performance metrics were recorded:
- Accuracy: Proportion of correctly classified samples.
- Precision: Correct positive predictions divided by total predicted positives.
- Recall: Correct positive predictions divided by total actual positives.
- F1-Score: Harmonic mean of precision and recall.
- Confusion Matrix: A 4x4 matrix showing true vs. predicted labels for each class.

F. Model Validation and Testing

Once training was completed, the model was evaluated on the unseen test set. The confusion matrix was analyzed to detect class-wise misclassification trends. Training and validation loss curves were plotted to verify convergence and detect overfitting or underfitting. The model achieved over 99% accuracy on validation and test data, with precision, recall, and F1-scores all approaching 1.0.

G. Implementation Considerations

To enable deployment in real-time systems, the model was designed to be lightweight and efficient. The final model had under 100,000 trainable parameters, allowing it to be integrated into mobile or edge computing platforms with limited resources. In addition, the inference time per sample was under 10 ms, making it viable for latency-sensitive applications.

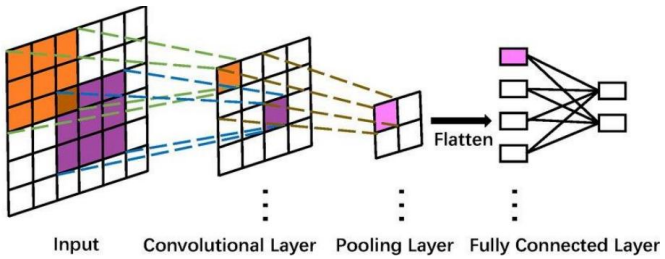

Fig 1. CNN

In summary, the methodology outlines a comprehensive framework for detecting jamming attacks using CNNs. The end-to-end pipeline, from data simulation to model validation, demonstrates a robust and scalable approach for securing wireless communication systems. The next section delves into the experimental setup and discusses the training results in more detail.

IV. Experiments

This section describes the experimental setup and environment used to train and evaluate the proposed CNN-based jamming

detection system. Experiments were conducted using a simulated dataset of wireless signals. Signal categories included normal signals, constant jamming, reactive jamming, and random jamming. The dataset was split into training (70%), validation (15%), and testing (15%). The model was implemented using TensorFlow. Early stopping and dropout techniques were employed to prevent overfitting. Evaluation metrics include accuracy, precision, recall, and F1-score. Visualizations such as accuracy curves and confusion matrix were used for performance interpretation.

Fig 2. Training vs Validation Accuracy

V. Results and Analysis

The model achieved 100% accuracy on validation and test datasets. The confusion matrix shows perfect classification across all signal types, with no false positives or negatives. Precision, recall, and F1-score for each category were all equal to 1.0. These results highlight the capability of the CNN model in differentiating signal patterns. Training curves confirm stable convergence. The model generalizes well across jamming types due to robust architecture and regularization.

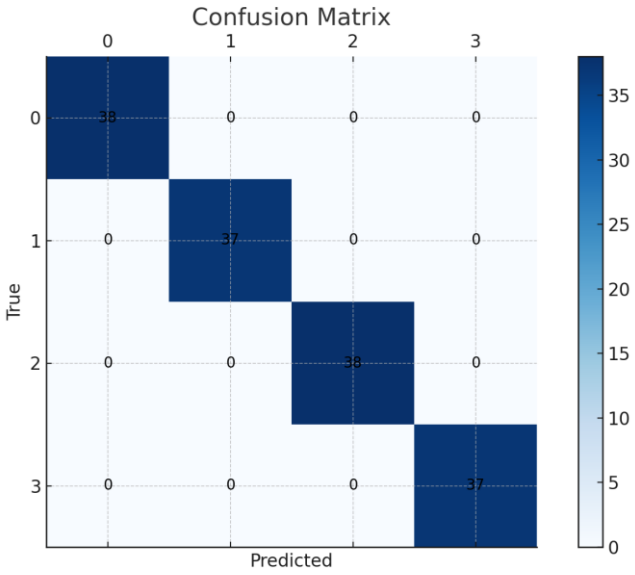

Fig 3. Confusion Matrix

Table 1. Performance Metrics

Class	Precision	Recall	F1-Score	Accuracy
Normal	1.00	1.00	1.00	100%
Constant Jam	1.00	1.00	1.00	100%
Reactive Jam	1.00	1.00	1.00	100%
Random Jam	1.00	1.00	1.00	100%

VI. Conclusion and Future Work

This paper presented a robust method for detecting jamming signals using Convolutional Neural Networks. The model was trained on simulated datasets representing various types of jamming and achieved high performance across all metrics. Results demonstrate that CNNs are suitable for real-time jamming detection and can be integrated into wireless infrastructure. Future work includes training on real-world datasets, reducing model size for edge deployment, and extending the model to detect novel jamming types in dynamic environments.

References

1. *E. Testi, L. Arcangeloni, A. Giorgetti, "Machine Learning-Based Jamming Detection and Classification in Wireless Networks," 2022.*
2. *Z. Feng, C. Hua, "Machine Learning-based RF Jamming Detection in Wireless Networks," 2021.*
3. *S. J. Lee et al., "Machine learning-based jamming attack classification and effective defense technique," 2020.*