International Journal of Applied Mathematics

Volume 38 No. 11s 2025,

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

SIGNAL CLASSIFICATION FOR JAMMING DETECTION: A COMPARATIVE STUDY OF MACHINE LEARNING AND DEEP LEARNING APPROACHES

A.George Arokiaraj[1] and S. Srinivasan*[1]

[1]Department of Advanced Computing Sciences, AMET University, Chennai, India.

[1]Professor, Department of Advanced Computing Sciences, AMET University, Chennai, India.

*Corresponding Author: srinikcgmca@gmail.com

Abstract

This study investigates the effectiveness of several machine learning and deep learning algorithms in a binary classification between jamming and non-jamming signals. Models under scrutiny include Support Vector Machine, Logistic Regression, Random Forest, K-Nearest Neighbors, Deep Neural Network and Long Short-Term Memory networks. The results indicate that traditional machine learning models, specifically Support Vector Machine, performed very well as compared to deep learning models, which resulted in an accuracy of 96.97% and AUC of 0.99. Results for Random Forest and K-Nearest Neighbors are also promising with high precision and favorable recall. In contrast, deep learning models, such as Deep Neural Network and Long Short-Term Memory, performed badly with small datasets. Long Short-Term Memory had a good recall of 99.73%, poor precision at 48.86%, whereas Deep Neural Network performed at low accuracy of 47.90% and AUC of 0.47. These outcomes clearly point out the fact that the machine learning models were highly adaptable in small dataset scenarios and therefore could be used more effectively for resource-constrained classification problems. Although deep learning models have great promise, they require large amounts of data and advanced preprocessing for peak performance. This study has underlined the continued relevance and effectiveness of machine learning models in the signal classification domain for practical applications.

Keywords: machine learning, deep learning, binary classification, signal classification, jamming detection, data augmentation.

Introduction

Signal classification becomes an area of growing research interest based on the increased dependence on the digital communications and electronic warfare systems. One of the major challenges in this field of work is the recognition of different types of signals in particular jamming signals. Jamming or interference in communication networks, radar operations, or navigation systems can be a very critical threat to both military and civilian infrastructure. With the estimated number of interconnecting devices around the world reaching 75 billion by 2025, so does the vulnerability to jamming attacks. The requirement for secure pathways of

Received: August 07, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)
communication in the military environment is also critical since disruptions lead to serious operational failure. In the domain of air defense systems, the ability to distinguish and counter jamming capabilities can greatly determine the end result of military operations. Therefore, it is important that proficient classification algorithms be developed capable of distinguishing between jamming and non-jamming signals in order to protect various communication infrastructures and maintain an operational integrity of vital areas of defense, transportation, or healthcare.

Machine learning and deep learning methods have gained much interest over the past decade since they are capable of processing massive amounts of datasets and can automatically identify essential patterns within raw data. They have excellent applicability to problems on classification concerning signals, wherein noisy datasets carry subtle patterns that lie concealed. These methods have been found to have great promise for a variety of applications, including speech recognition, medical diagnostics, and fraud detection in financial dealings. However, in the area of signal classification, such as jamming detection, it remains difficult to achieve high accuracy under sparse data, imbalance, or noisy conditions. According to a report by the ITU, it is said that In worldwide communications, about 35 percent is interfered with in some way with jamming being one significant aspect of interference.

The traditional machine learning models, such as Support Vector Machines (SVM), Random Forest and K-Nearest Neighbors (KNN), have been found to work well for smaller, well-structured datasets. However, deep learning models like Long Short-Term Memory (LSTM) and Deep Neural Networks (DNN) are now getting more complex in their ability to capture complex, non-linear relationships in data, such as temporal dependencies in communication signals. However, deep learning architectures require large datasets as well as careful data pre-processing to reach peak performance. Their performance will drop significantly when applied on small datasets. Moreover, class imbalance and signal noise are common issues that affect deep learning models, leading to misclassifications and lower reliability.

This paper aims at evaluating and comparing the machine learning and deep learning models, in terms of the performance of such models in binary signal classification, specifically, classifying jamming and non-jamming signals. Evaluation metrics of accuracy, precision, recall, F1-score, and Area Under the Curve (AUC) are employed to measure the pros and cons of the different models within this framework. The results from this research will yield valuable insights into whether machine learning methodologies are effective for signal classification and whether deep learning frameworks require further development. The next steps in the research will be the improvement of model effectiveness by various means, including data augmentation, transfer learning and hybrid architectures, in order to enhance the detection and classification of jamming signals in practical applications, particularly in electronic warfare and secure communication systems. Given the increasing prevalence of digital communications in critical areas such as defense, healthcare and emergency response, reliable detection of jamming signals is not merely an academic challenge but a practical necessity. This study is

Received: August 07, 2025

International Journal of Applied Mathematics
Volume 38 No. 11s 2025,
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

motivated by real-world incidents where undetected jamming led to operational disruptions. We aim to address a practical gap faced by communication engineers working under resource constraints.

Literature Review

Signal classification and jamming detection are some of the critical components to make the contemporary communication system both secure and reliable. In fact, the application of machine learning (ML) and deep learning (DL) has become significantly better in terms of identifying and mitigating jamming signals, which disturbs the communication channels. The following literature review provides the recent progress made with the help of ML and DL techniques for signal classification and jamming detection. Traditional machine learning algorithms are very applicable to jamming detection due to efficiency in working with structured datasets. Arjoune et al. (2020) compared several machine learning techniques in identifying jamming attacks for the wireless communication system like Random Forest, Support Vector Machine, and Neural Network. The study showed that the Random Forest algorithm had achieved high accuracy and detection probability and It was suitable as an alternative for applications related to jamming detection. Varotto et al. (2024) discussed the efficiency of k-Nearest Neighbors (k-NN) and Support Vector Machines (SVM) for the identification of narrowband jammers in 5G networks. This evidence suggests that the traditional machine learning models, when used together with PCA for dimensionality reduction, are able to classify the jamming signal accurately, which further highlights that feature extraction is important to enhance the detection.

Deep architectures of learning, particularly Convolutional Neural Networks (CNNs) have emerged as extremely important in the signal classification area. This is primarily due to the autonomous generation of hierarchical features from unprocessed signal data. Wu and Zhao, 2018 used CNNs to classify jamming signals. Excellent accuracy levels were achieved in this case. However, they reported problems with handling unknown types of jamming and the open-set problem. This necessitates models that can handle interference patterns robustly.

A recent study, that by Varotto et al. in 2024, proposed an application of CNNs for the classification of 5G signal jammers by analyzing spectrograms of IQ samples. The result showed that CNNs are capable of classifying jamming signals and, hence, a feasible approach for instantaneous jamming detection in high-level communication systems. ML and DL-based approaches have been proposed to exploit the benefits of both techniques. For instance, the study by Arjoune et al. (2020) integrates feature extraction methods with machine learning classifiers for improved accuracy in jamming detection. The integrative approach tries to enhance detection performance using deep learning models that benefit from feature extraction capabilities as well as classification through the machine learning algorithm. However, a few issues are still likely to exist in the scenario of jamming detection. The limited availability of annotated data for model training continues to present a major challenge, as highlighted by Shi

International Journal of Applied Mathematics
Volume 38 No. 11s 2025,
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

et al. (2019), who underscored the necessity of extensive, labeled datasets for the effective training of deep learning models. Moreover, the ever-changing characteristics of communication environments necessitate models capable of adjusting to novel and developing jamming strategies. The future work includes adapting the models to be able to handle unknown types of jamming and generalize the detection systems for any communication scenario. The work on applying ML and DL techniques in signal classification and jamming detection has shown encouraging results toward increasing the robustness of communication systems against interference. Although conventional machine learning models provide efficient solutions for structured data, deep learning models bring in more enhanced functionalities when processing intricate and unstructured signal data. Hybrid methods that bring the benefits of both paradigms are thus gaining importance as a significant area for future research. The main challenges such as scarcity of data and the need for adaptable models will have to be addressed for the advancement of jamming detection.

Methodology

The methodology of this study involves a series of systematic phases to evaluate and compare various machine learning (ML) and deep learning (DL) models for binary signal classification, specifically distinguishing between “jamming” and “non-jamming” signals. The phases include data acquisition, preprocessing, model selection, model training and evaluation. Each phase is discussed in detail below and illustrated in Figure 1.

1. Data Acquisition

Data acquisition is a crucial phase of this study, where the dataset used for binary signal classification specifically to distinguish between "jamming" and "non-jamming" signals is gathered. This dataset is vital for evaluating the performance of machine learning (ML) and deep learning (DL) models in jamming detection for communication systems. The dataset contains a total of 10,000 samples, representing real-world communication systems and synthetic signal models.

1.1 Dataset Description

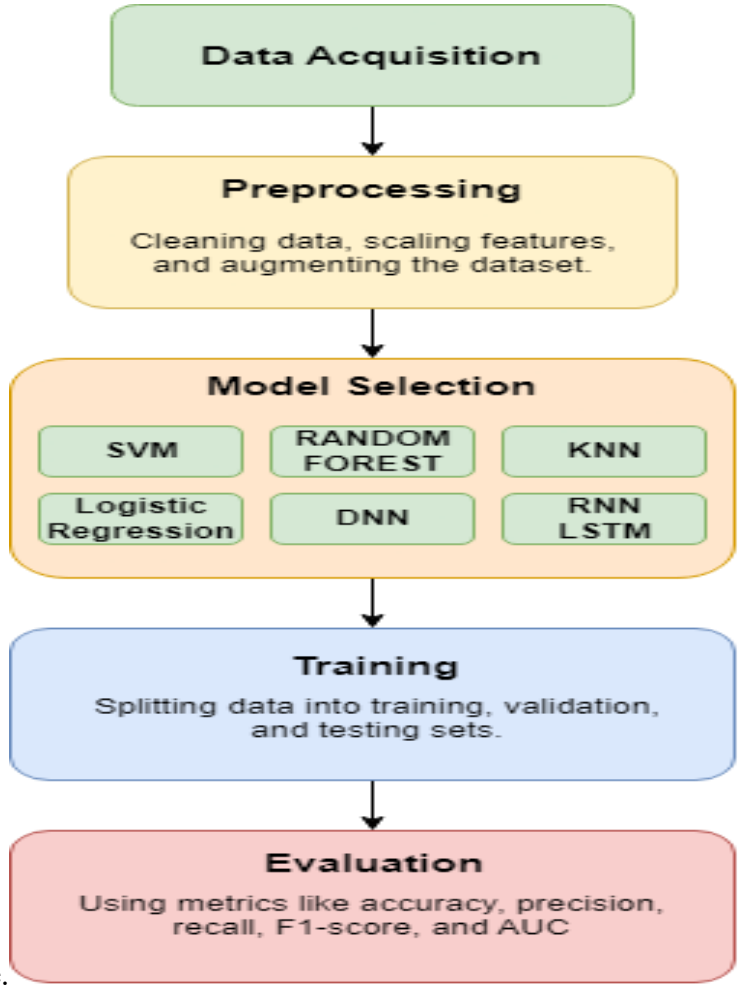
The dataset used in this study is a comprehensive collection of 10,000 signal samples, consisting of both jamming and non-jamming signals. These signals are designed to simulate interference patterns in wireless communication systems. The dataset includes the following types of signals:

- **Jamming Signals:** These interference signals are generated using various jamming techniques, such as broadband noise, pulsed interference, and frequency hopping. These jamming signals aim to represent realistic scenarios in which communication signals are intentionally disrupted.

International Journal of Applied Mathematics
Volume 38 No. 11s 2025,
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

- **Non-Jamming Signals:** These clean signals represent undisturbed communication channels and serve as a baseline for classification.

The dataset is composed of 5,000 samples of jamming signals and 5,000 samples of non-jamming signals, creating a balanced distribution of both classes. The signal data is sourced from both publicly available signal repositories and synthetic signal generation models. The balanced dataset ensures a fair comparison o



f model performance.

Figure.1. Research Methodology

1.2 Signal Features

Each signal in the dataset is represented by a comprehensive set of 20 features extracted from both time-domain and frequency-domain properties. These features are carefully designed to capture the essential characteristics of jamming and non-jamming signals, enabling effective classification. The feature set is divided into two categories:

Table 1. Features and characteristics of Dataset

Received: August 07, 2025

International Journal of Applied Mathematics
Volume 38 No. 11s 2025,
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Feature Category	Feature	Description
Time-Domain Features	**Mean**	Represents the average value of the signal.
	Variance	Measures the spread or variability of the signal.
	Skewness	Quantifies the asymmetry of the signal distribution.
	Kurtosis	Indicates the peakedness of the signal distribution.
	Peak-to-Peak Value	Represents the maximum amplitude difference in the signal.
	Signal Energy	Computes the total energy contained in the signal.
	Root Mean Square (RMS) Value	A statistical measure of the signal's magnitude.
	Zero-Crossing Rate	Counts the number of times the signal crosses the zero amplitude line.
	Autocorrelation	Measures the self-similarity of the signal over time.
	Signal Duration	Represents the total time length of the signal.
Frequency-Domain Features	**Power Spectral Density (PSD)**	Describes the distribution of signal power across various frequencies.
	Peak Frequency	Identifies the frequency component with the highest power.
	Bandwidth	Measures the range of frequencies occupied by the signal.
	Total Power in Frequency Bands	Summarizes the signal’s power across specified frequency ranges.
	Spectral Entropy	Quantifies the complexity of the signal spectrum.
	Spectral Centroid	Indicates the "center of mass" of the frequency spectrum.
	Spectral Flux	Captures the rate of change in the power spectrum.
	Spectral Roll-off	Defines the frequency below which a specified percentage of the total spectral energy resides.
	Harmonic Ratio	Reflects the proportion of harmonic components relative to the total signal.
	Frequency Skewness	Describes the asymmetry of the frequency distribution.

Received: August 07, 2025

International Journal of Applied Mathematics
Volume 38 No. 11s 2025,
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

By integrating both time-domain and frequency-domain features, this feature set effectively encapsulates the temporal and spectral characteristics of the signals. The time-domain features highlight the overall shape and statistical properties of the signal, while the frequency-domain features provide detailed insights into the spectral behavior. Together, these attributes are instrumental in recognizing patterns and interference structures associated with jamming, thereby ensuring accurate and robust classification.

2. Data Pre-processing

Figure.2. Data Pre-Processing Workflow for Signal Classification

Data pre-processing is therefore very important in preparing the raw data, which will be fed into ML and DL models in order to ensure the data quality and adequacy during the training process. Such a process of pre-processing includes several important steps, such as cleaning, feature scaling, feature selection, and data augmentation shown in Figure 2 below. Data cleaning was executed to solve some problems concerning the missing, duplicated, and inconsistent data records. Missing values were handled using mean imputation: where every feature's missing values are replaced by the mean of the available values, and outliers removed using Interquartile Range so that the data set contains no extreme values that will distort the model performance. Duplicate entries were also detected and deleted to ensure that only unique samples were used in training of the model. After the data cleaning, feature scaling was conducted to normalize the range of the different features. Min-Max normalization was employed to scale all feature values to a normalized interval between 0 and 1 and ensuring that

International Journal of Applied Mathematics
Volume 38 No. 11s 2025,
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)
each feature has an equal effect on the model's performance. This is particularly important for models like SVM and KNN, which are sensitive to the magnitudes of the features. For models like Logistic Regression, standardization was used. Data was transformed to have zero mean and unit variance, which ensures that features are normally distributed and are better suited for training.

Feature selection was done to select the most relevant features and retain them. This reduced the dimensionality of the dataset and improved the performance of the model. A correlation matrix was used to identify features that have high correlation. Features whose correlation coefficients were above a threshold value of 0.9 were removed to avoid multicollinearity. Moreover, Recursive Feature Elimination (RFE) has been employed to refine the choice of features with the strongest importance toward the models and eliminate others that do not significantly influence the predictive strength of models. Data augmentation methods have also been used to balance the dataset by focusing especially on the non-jamming signal category to overcome class imbalance problems and enhance model generalization. Methods such as noise injection, that involves adding random Gaussian noise to the signal samples, besides time shifting, frequency shifting, and signal stretching/compression were used. Such techniques introduce variability to the dataset, which helps models learn to classify signals with a spectrum of characteristics and improve the robustness of the models against real-world interference.

The dataset was split into training and testing sets. 80% of the data was allocated for training the models while the remaining 20% was reserved for testing. The split was stratified to ensure that both jamming and non-jamming classes were proportionally represented in both sets. For deep learning models, such as the Long Short-Term Memory (LSTM) network, additional Z-score normalization was also applied for zero mean and unit variance enforcement on the data, as this helps to improve convergence of deep algorithms in training. Preprocessing procedures which include data cleaning, feature normalization, feature selection, data augmentation, and proper partitioning of data, were fundamental to transforming the raw data into a suitable format to use in training and evaluation of machine learning and deep learning models. This transformation led to the models being able to learn well and produce accurate predictions.

3. Model Selection

In this study, both traditional machine learning (ML) models and deep learning (DL) models were evaluated for binary signal classification tasks specifically distinguishing between jamming and non-jamming signals. The models were selected based on several factors including their suitability for handling high-dimensional data, performance in signal processing tasks and ability to manage both structured and time-series signals as illustrated in Figure.3. A total of six machine learning models and two deep learning models were considered for this study, offering a diverse range of techniques that allowed for a comprehensive comparison of their strengths and weaknesses in signal classification.

Received: August 07, 2025

International Journal of Applied Mathematics
Volume 38 No. 11s 2025,
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

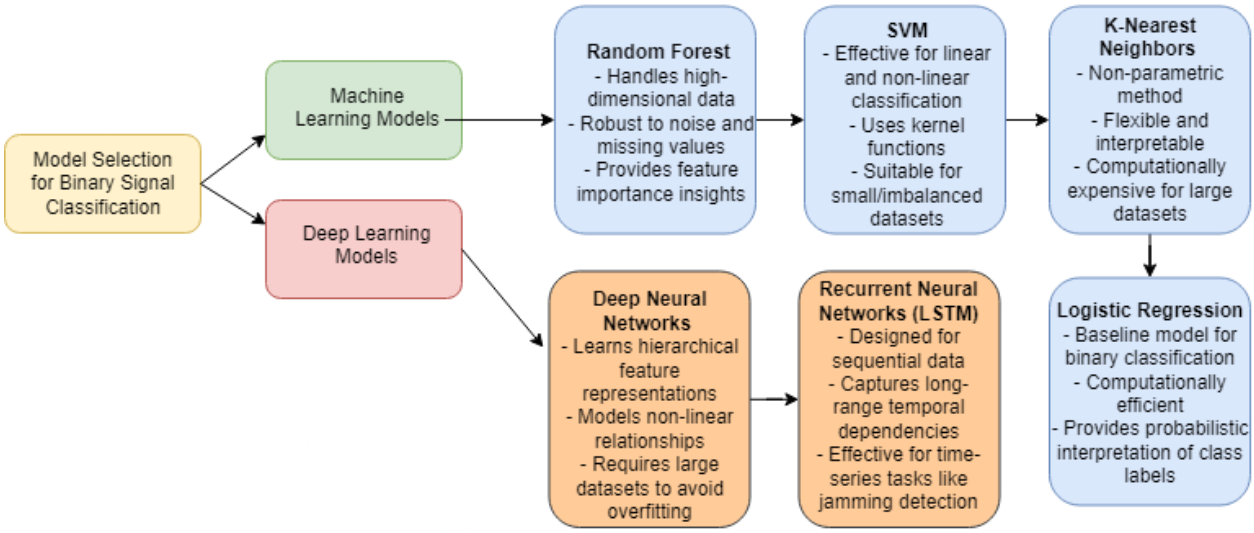


Figure.3. Hierarchical Mind Map of Model Selection for Binary Signal Classification

3.1 Machine Learning Models

The following ML models were chosen due to their well-established effectiveness in classification tasks and ability to do reasonably well even with relatively smaller or imbalanced datasets. The ML models evaluated were:

● **Random Forest (RF):** Random Forest is an ensemble learning method that constructs multiple decision trees during training and outputs the mode of the classes (classification) or mean prediction (regression) of the individual trees. The basic reason for selecting Random Forest is that it handles vast and high-dimensional datasets without overfitting risk. It is particularly well-known for its robustness to noise and ability to handle missing values efficiently. Random Forest provides valuable feature importance insights and it can be used in the process of feature selection and identifying the most relevant features for the task of signal classification. The capacity to gather predictions produced by many decision trees enhances the accuracy of classification and decreases the variability of predictions, which makes it very effective for complex signal datasets with inhomogeneous patterns.

● **SVM:** SVM is a powerful approach to supervised learning that can be applied very effectively to both linear and nonlinear classification problems. SVM is very effective in high-dimensional spaces and especially good for binary classification problems. This model was chosen for its theory, which ensures that this decision boundary between classes is as optimal as it can be, even given sparse or high-dimensional data. One of the SVM's most distinctive features lies in its use of kernel functions, which enable it to learn non-linear relationships between features and the target variable. Support Vector Machines (SVM) demonstrate notable efficacy in situations characterized by restricted data availability, rendering them appropriate for jamming detection endeavors where the datasets might be minimal or unbalanced. Furthermore, when appropriately calibrated, SVM is less susceptible to overfitting relative to

Received: August 07, 2025

International Journal of Applied Mathematics
Volume 38 No. 11s 2025,
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

numerous alternative classifiers, establishing it as a dependable option for tasks involving signal classification.

● **K-Nearest Neighbors (KNN):** K-Nearest Neighbors (KNN) represents a straightforward but efficient non-parametric technique employed in classification tasks. It works by identifying the 'K' nearest data points to a given test instance and then classifies the instance according to the majority class amongst those neighbors. This is actually the flexibility of the KNN, as it can classify the data without making any assumption that could be made about the underlying distribution of the data. This is very important when the signal data is complex or even non-linear. The reasons behind this are that it is simple and easy to interpret and since there is no training phase for the KNN; it just stores the training data. However, it might become computationally expensive at the time of prediction phase especially when the dataset size is large. Still, with all this, it does very well in signal classification, provided the structure of the data is well set up and the relationship among data points is well defined.

● **Logistic Regression:** Logistic Regression is a basic linear model often used as a benchmark for comparison in binary classification tasks due to its simplicity, efficiency and interpretability. Though it cannot capture complex non-linear relationships, its mathematical structure is easy to understand and thus fully grasp the relationship between input features and the dependent variable. Logistic Regression was used in this research to set a baseline against more complex models. The computational efficiency and effectiveness of the model are quite improved if the decision boundary separating the classes is almost linear, which often is the case in elementary signal classification problems. Moreover, Logistic Regression allows the direct evaluation of the probabilities, thus providing the probabilistic interpretation of the class labels predicted. As a basic model, it gives valuable insights about the comparative predictive performance of more advanced modeling techniques.

3.2 Deep Learning Models

Deep learning architectures are increasingly being used for complex classification tasks because of their ability to automatically learn hierarchical feature representations and model complex non-linear relationships. For the purposes of this study, two specific deep learning models were selected because of their established effectiveness in signal processing tasks that involve both temporal dependencies and high-dimensional data.

●**DNN:** Deep Neural Networks represent a class of neural networks characterized by multiple layers facilitating the learning of hierarchical data representations, which are crucial to tasks that are characterized by patterns and relationships inherent in the data. They consist of many hidden layers that incrementally abstract input features. It allows the acquisition of high-level representations, rendering them particularly effective for classification tasks of complex signals. DNNs can model highly non-linear relationships in data, which is crucial when dealing with diverse and noisy signal patterns. The reason for using DNNs was the capacity to handle high-dimensional data and learn from large, unstructured datasets without needing manual feature extraction. DNNs, however, require a large amount of data to prevent overfitting and ensure effective training. This property of the networks to model complex interconnections

International Journal of Applied Mathematics
Volume 38 No. 11s 2025,
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

makes them particularly suited for tasks like jamming detection, which require subtle patterns within time-series datasets to be recognized.

● **LSTM Networks:** The choice of RNNs architectures was based on Long Short-Term Memory (LSTM). Due to their capability to model sequential information and capture dependencies over long ranges of time. LSTMs are RNNs specifically designed to reduce the vanishing gradient problem. these allow for the ability to maintain information over a very long time and for this reason, they work very effectively in time-series tasks like signal classification in jamming detection. LSTMs can detect time-varying patterns and recognize dependency in data that might have been missed by other models used in signal processing tasks where temporal relationships among samples of a signal are key. Because signals are naturally time-varying, the LSTMs are particularly suitable for use in applications like jamming detection where interference patterns evolve with time and need to be interpreted over earlier states of the signal. If enough training data is provided to the LSTMs, then they might surpass the models used in machine learning.

3.3 Hyperparameters

Hyperparameters for each model were selected based on commonly used values in the literature, as well as experimentation with different configurations using Grid Search and Random Search techniques. These optimization methods were used to identify the best combination of hyperparameters that maximized the model's performance. The following table summarizes the most common hyperparameters used for each model:

Table 2. Hyperparameters for Each Model

Model	Hyperparameter	Best Value
Random Forest	n_estimators	500
	max_depth	20
	min_samples_split	2
	min_samples_leaf	1
	max_features	sqrt (auto)
Support Vector Machine	C	1
	kernel	rbf
	gamma	scale
	degree	3
K-Nearest Neighbors (KNN)	n_neighbors	5
	weights	distance
	metric	euclidean
Logistic Regression	C	1
	solver	liblinear
	max_iter	200

Received: August 07, 2025

International Journal of Applied Mathematics
Volume 38 No. 11s 2025,
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Model	Hyperparameter	Best Value
	penalty	l2
Deep Neural Network (DNN)	hidden_layer_sizes	(100, 100)
	activation	relu
	batch_size	64
	epochs	200
	learning_rate_init	0.001
Recurrent Neural Network (LSTM)	units	100
	activation	tanh
	dropout	0.3
	batch_size	64
	epochs	100
	learning_rate	0.001

3.4 Hyperparameter Tuning

Hyperparameter tuning was performed using Grid Search and Random Search to identify the optimal hyperparameters for each model. Grid Search systematically tests all possible combinations of the hyperparameters, while Random Search samples from the hyperparameter space. Both methods were employed to identify the best hyperparameters for maximizing performance. Cross-validation (5-fold or 10-fold) was used during the tuning process to ensure the results are generalizable and not overfit to a particular training set.

4. Training Process

The training phase in this study involved fitting the selected machine learning and deep learning models to the preprocessed data. This phase is crucial, as the performance of the models heavily depends on the training process which involves optimizing model parameters to learn patterns from the data. The following sections detail the training process for each model, including data splitting, training procedure and training configuration.

4.1. Data Splitting

The data was split into three clearly defined sets: training, validation, and test datasets to carry out robust model evaluation without overfitting. This training dataset used 70% of the total data for model training. In addition, this validation dataset that constituted 15% of the data was used in tuning the model hyperparameters while training and keeping track of performance on unseen data. The test dataset which constituted the remaining 15%, was used for final evaluation and performance comparison of the models.

Received: August 07, 2025

International Journal of Applied Mathematics
Volume 38 No. 11s 2025,
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

4.2. Model Training for Traditional ML Models

The machine learning models including Random Forest, SVM, KNN and Logistic Regression are trained using standard libraries available in scikit-learn (v0.24.2). These models were trained using the training dataset and cross-validation techniques were used to optimize hyperparameters and avoid overfitting.

- **Random Forest** was trained using a grid of hyperparameters, focusing on the number of trees (n_estimators) and tree depth (max_depth) among others. The best-performing model based on cross-validation scores was selected for final evaluation.
- **Support Vector Machine (SVM)** training involved choosing a suitable kernel function (e.g., radial basis function, polynomial) and tuning the regularization parameter (C) and kernel-specific parameters (e.g., gamma) using Grid Search.
- **K-Nearest Neighbors (KNN)** training involved varying the number of neighbors (n_neighbors) and the distance metric (Euclidean vs Manhattan). These parameters were optimized for improved classification accuracy.
- **Logistic Regression** was trained using liblinear and saga solvers with hyperparameter C (regularization strength) tuned to minimize classification error.

4.3. Model Training for Deep Learning Models

Deep learning models, such as Deep Neural Networks (DNN) and Recurrent Neural Networks (LSTM) were trained using TensorFlow (v2.6) and Keras (v2.6). These models require different training strategies due to their complexity and deep layers.

- **Deep Neural Network (DNN)** training focused on optimizing the number of hidden layers and their sizes. The activation function (ReLU) was used for hidden layers and the Adam optimizer was employed to minimize the loss function. The batch size, learning rate and number of epochs were also fine-tuned using Grid Search to determine the optimal configuration.
- **Recurrent Neural Network (LSTM)** training focused on tuning the number of units in the LSTM layer and optimizing the learning rate. The Adam optimizer was used to minimize the binary crossentropy loss function. The number of training epochs was determined based on the convergence of the validation loss.

4.4. Training Configuration

For both traditional Machine learning and deep learning models, the following training configurations were used:

- **Learning Rate**: Learning rates are adjusted during training for the deep learning models using a learning rate decay strategy. For instance, DNN and LSTM benefited from much lower learning rates with which they converge stably to a solution during training.

Received: August 07, 2025

International Journal of Applied Mathematics
Volume 38 No. 11s 2025,
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

- **Batch Size**: The batch size was set to 32 for deep learning models, balancing computational efficiency with model performance. For traditional ML models, batch processing was not necessary since the models were trained on the entire dataset.
- **Early Stopping**: In both DNN and LSTM models, early stopping was used in order to avoid overfitting. This method would track the validation loss and stopped training at a point where performance became stagnant for a number of epochs.

4.5. Computational Resources

Due to the scarcity of resources in training, all models were trained on CPU-based systems without GPU support. All these models depended on the training time. The dependency was based on the complexity of the model. For example, the SVM model typically took a few minutes to train, while DNN and LSTM models required longer training times (up to several hours), particularly with a larger number of epochs.

5. Model Evaluation

Model evaluation is essential to determine the effectiveness and reliability of machine learning models. In this study, various evaluation metrics were used to assess the performance of the models trained for binary signal classification, specifically identifying "jamming" versus "non-jamming" signals. The evaluation process allows for comparing the models' strengths and limitations and provides insight into their practical applicability. The following sections describe the evaluation metrics used and summarize the results for each model.

5.1. Evaluation Metrics

To ensure comprehensive evaluation, the following metrics were used to measure the performance of each model:

- **Accuracy**: This metric calculates the proportion of correct predictions (both true positives and true negatives) out of all predictions. While accuracy is useful for assessing the overall performance, it can be misleading when the dataset is imbalanced.
- **Precision**: Precision measures the percentage of true positive predictions out of all instances classified as positive. A higher precision indicates fewer false positives which is particularly important in scenarios where false alarms are costly.
- **Recall**: Also known as sensitivity, recall measures the percentage of true positive predictions out of all actual positives. High recall is vital when the cost of missing a positive instance (false negatives) is high.
- **F1-Score**: The F1-score is the harmonic mean of precision and recall. It balances the trade-off between the two and is useful when dealing with imbalanced datasets where one metric alone may not reflect true model performance.
- **Area Under the Curve (AUC)**: The AUC of the receiver operating characteristic (ROC) curve is a performance measurement for classification problems. A higher AUC

International Journal of Applied Mathematics
Volume 38 No. 11s 2025,
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

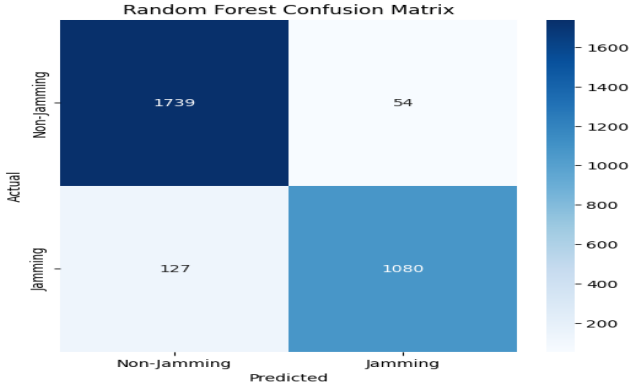
indicates that the model is better at distinguishing between the classes. An AUC of 1 represents perfect classification, while an AUC of 0.5 suggests random performance.

Results and Discussion

To illustrate, consider an emergency network coordinator who must identify and isolate interference on the fly often with limited data on novel jamming strategies. The robust performance of SVM and Random Forest in our tests directly translates into faster, more reliable detection even where resources and time are limited. This study evaluated the performance of various machine learning (ML) and deep learning (DL) models in the binary classification task of signal classification, distinguishing between "jamming" and "non-jamming" signals. To assess the models' performance, key metrics such as accuracy, precision, recall, F1-score, and Area Under the Curve (AUC) were considered. A detailed comparison of the models' outcomes is provided in Table 3, which summarizes their relative strengths and limitations.

Table 3. Evaluation Metrics for Various Classification Models

Model	Class	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	AUC
Random Forest	0	93.97	95.24	89.48	92.27	0.98
	1	93.97	94.12	98.15	96.10	0.98
Support Vector Machine	0	96.97	97.13	95.28	96.19	0.99
	1	96.97	96.75	98.42	97.57	0.99
K-Nearest Neighbors	0	95.43	97.35	91.14	94.14	0.98
	1	95.43	94.02	96.74	95.36	0.98
Logistic Regression	0	82.23	81.61	72.08	76.55	0.88
	1	82.23	79.38	85.94	82.53	0.88
Deep Neural Network	0	47.90	96.00	94.00	95.00	0.47
	1	47.90	93.00	96.00	94.00	0.47
Recurrent Neural Network (LSTM)	0	48.97	94.00	97.00	96.00	0.98
	1	48.97	97.00	93.00	95.00	0.98



Received: August 07, 2025

International Journal of Applied Mathematics
Volume 38 No. 11s 2025,
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Figure 4. Confusion Matrix of Random forest

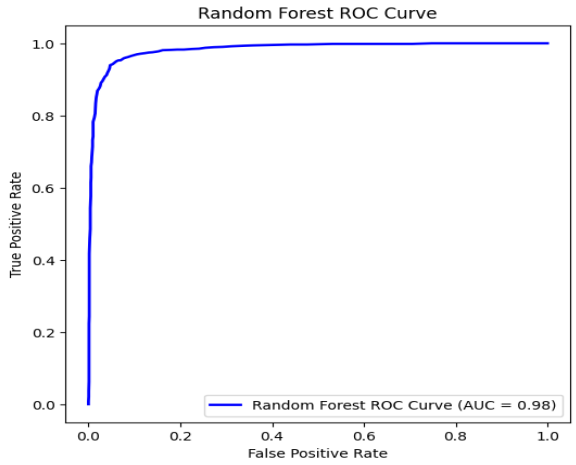


Figure 5. RoC Curve of Random Forest

Among the traditional ML models, the Support Vector Machine (SVM) achieved the most impressive performance, recording the highest accuracy (96.97%) and AUC (0.99), as shown in Figures 6 and 7. This model demonstrated excellent balance between precision (97.13%), recall (95.28%), and F1-score (96.19%), establishing it as the most robust solution for the task. Random Forest followed closely,

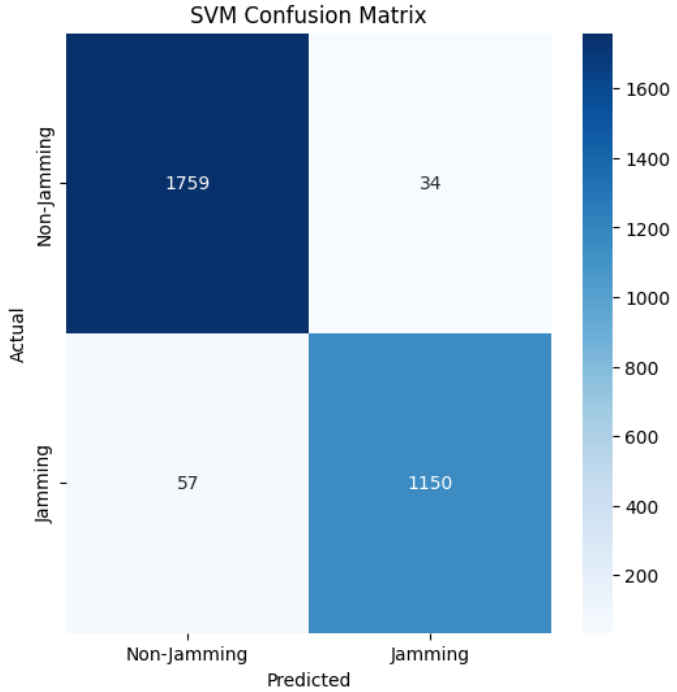


Figure 6. Confusion Matrix of SVM

International Journal of Applied Mathematics
Volume 38 No. 11s 2025,
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

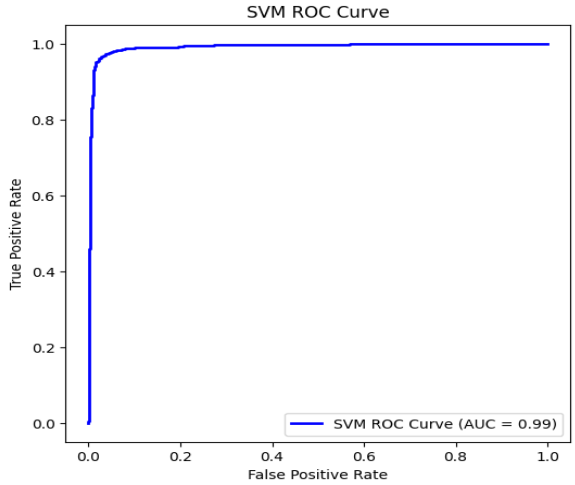


Figure 7. RoC Curve of SVM

attaining an accuracy of 93.97% and an F1-score of 92.27%. However, it had a slightly lower recall (89.48%), as depicted in Figures 4 and 5, indicating a higher rate of missed true positives. K-Nearest Neighbors (KNN) also delivered strong results, with an accuracy of 95.43% and a notable precision of 97.35% (Figures 8 and 9). Despite this, its recall (91.14%) was marginally lower, which suggests an increased likelihood of false negatives. Logistic Regression, serving as a baseline model, demonstrated a moderate performance with 82.23% accuracy (Figure 10), offering a useful point of comparison.

In contrast, Deep Learning (DL) models encountered more significant challenges, particularly due to the limitations of the dataset size and the lack of sophisticated preprocessing techniques. The Deep

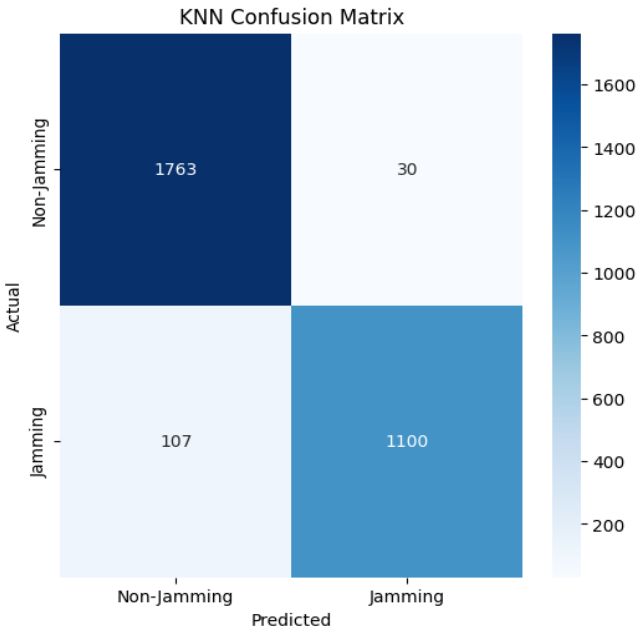


Figure 8. Confusion Matrix of KNN

Received: August 07, 2025

International Journal of Applied Mathematics
Volume 38 No. 11s 2025,
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

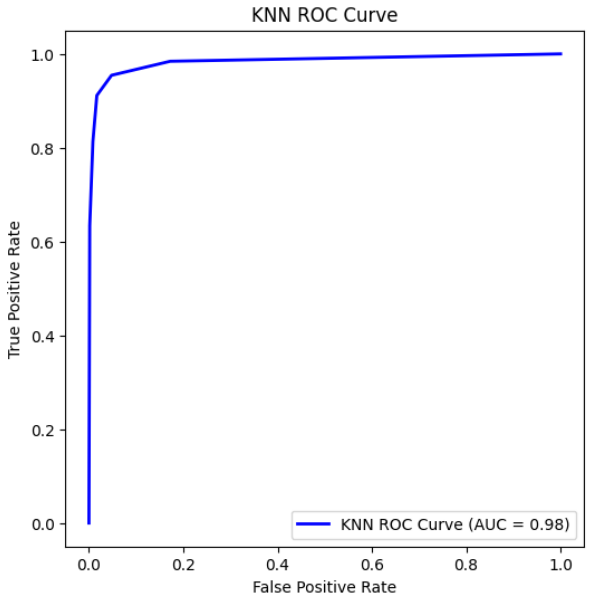
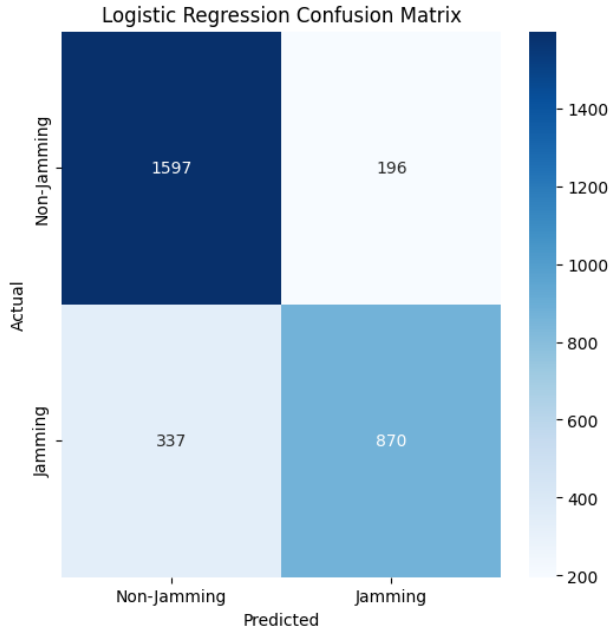


Figure 9. RoC Curve of KNN

Neural Network (DNN) struggled significantly with a low accuracy of 47.90% and a disappointing AUC of 0.47, as illustrated in Figure 12. This poor performance can be attributed to the DNN's inadequate ability to distinguish between classes, which emphasizes the need for more extensive datasets and improved optimization for DL models. Similarly, the Long Short-Term Memory (LSTM) network, while achieving a high recall of 99.73%, exhibited poor precision (48.86%), leading to an unbalanced F1-score of 65.59% (Figure 13). This resulted in a high number of false positives, underscoring the model's lack of precision in classifying signals effectively.



Received: August 07, 2025

International Journal of Applied Mathematics
Volume 38 No. 11s 2025,
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Figure 10. Confusion Matrix of Logistic Regression

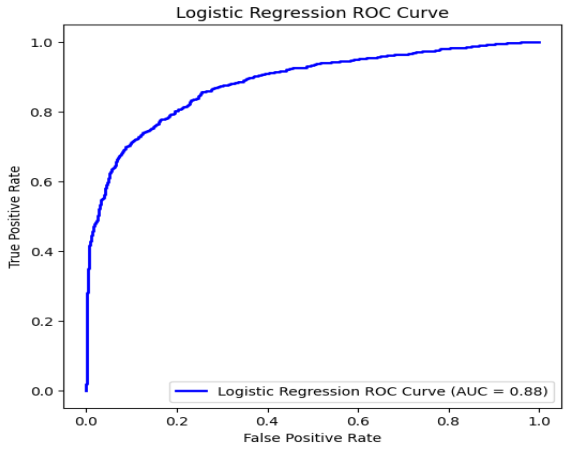


Figure 11. RoC Curve of Logistic Regression

The disparity in performance between ML and DL models highlights the greater adaptability of traditional ML techniques to smaller, structured datasets. SVM, in particular, emerged as the most effective model, demonstrating not only the highest overall performance but also a strong ability to handle the binary classification task with well-balanced metrics. Random Forest and KNN also delivered competitive results, with Random Forest showing reliable accuracy and F1-scores, while KNN excelled in precision but slightly lagged in recall. On the other hand, DL models, particularly DNN and LSTM, faced significant limitations due to their sensitivity to smaller datasets. Deep learning models are more prone to overfitting on smaller datasets because they require a large number of parameters to be effectively trained. Without sufficient data, these models struggle to generalize, leading to poor performance and low accuracy. Additionally, the lack of advanced feature extraction and preprocessing can impede their learning capacity. These issues underscore the importance of dataset expansion, better preprocessing, and hyperparameter optimization in the future development of DL models.

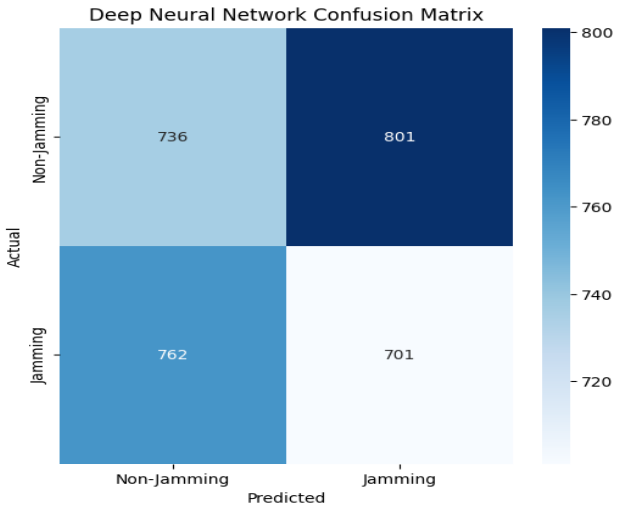


Figure 12. Confusion Matrix of DNN

Received: August 07, 2025

International Journal of Applied Mathematics
Volume 38 No. 11s 2025,
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

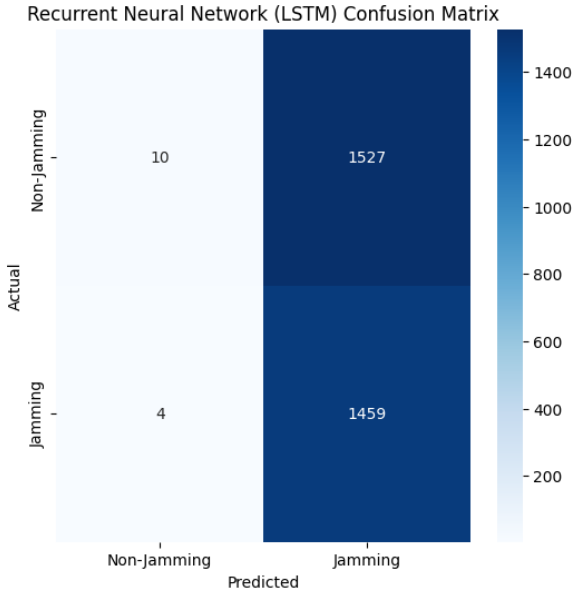


Figure 13. Confusion Matrix of RNN- LSTM

The results of this study clearly demonstrate the superiority of traditional ML models in binary signal classification tasks where dataset size is limited, and computational efficiency and interpretability are critical factors. SVM and Random Forest proved to be the most effective models for this application, with KNN being a solid choice for tasks requiring high precision. While Logistic Regression provided baseline insights, its overall performance was inferior to the other ML models.

The challenges faced by DL models in this context are indicative of the need for larger and more comprehensive datasets to fully realize their potential. The DNN's poor performance can be attributed to the limited dataset, which hindered its ability to effectively learn from the data. Similarly, the LSTM's impressive recall was overshadowed by its subpar precision, suggesting that improvements in data preprocessing, feature engineering, and hyperparameter tuning are essential for enhancing DL model performance in such tasks. Additionally, advanced techniques such as data augmentation, transfer learning, and the exploration of hybrid models could help mitigate the challenges posed by small datasets, enabling DL models to achieve better generalization and accuracy.

Practical Implications and Future Work: For real-world applications, both SVM and Random Forest are highly recommended due to their superior accuracy, balanced metrics, and suitability for smaller datasets. In contrast, KNN provides an excellent alternative when precision is a priority. Logistic Regression, although less effective, remains useful as a fast and interpretable baseline model. As for DL models, future efforts should focus on expanding the dataset, improving feature extraction techniques, and exploring more advanced network architectures such as Convolutional Neural Networks (CNNs) or hybrid approaches that combine the strengths of both ML and DL methods. While our curated and balanced dataset allows meaningful comparisons, it does not capture every nuance and variability present in field-

Received: August 07, 2025

Volume 38 No. 11s 2025,

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

derived signals. Future research should thus focus on collaborative real world data gathering and testing across new environments

In conclusion, this study underscores the effectiveness of traditional ML models, with SVM achieving the highest accuracy (96.97%) and AUC (0.99), as detailed in Figures 6 and 7. Random Forest (Figures 4 and 5) and KNN (Figures 8 and 9) also demonstrated strong performance, while DL models such as DNN (Figure 12) and LSTM (Figure 13) revealed the limitations of current deep learning methods when applied to smaller datasets. These findings reinforce the need for further refinement in DL techniques and confirm the ongoing relevance of ML models in scenarios involving binary signal classification. Future work should focus on addressing the dataset limitations and exploring hybrid models that combine the strengths of both ML and DL approaches. In practice, our results suggest that choosing the right model can mean the difference between effective jamming detection and critical communication breakdowns. For many teams, SVM and Random Forest are the safest bets when data and computational resources are tight.

Conclusion

The Study evaluated the performance of ML and DL algorithms in the binary classification challenge to distinguish "jamming" from "non-jamming" signals. Among different models of ML, SVM had the highest accuracy at 96.97% and an AUC value of 0.99; the metrics for precision, recall, and F1-score were also well-balanced. The Random Forest and K-Nearest Neighbors (KNN) algorithms performed well, showing strong ability to work with comparable results; however, KNN had a somewhat lower recall, whereas Random Forest had a much higher false negative rate. Logistic Regression provided a relatively modest benchmark level of performance, which ensured that there was real benefit to be gained with more complex machine learning approaches for this task. Deep learning architectures such as DNN and LSTM were found to have significant limitations. The DNN faced poor accuracy (47.90%) and AUC of 0.47 as the dataset was small. The LSTM had high recall but low precision and it leads to an imbalanced F1-score. The findings show that DL models have a problem in training when data is limited; thus, the efficacy of the ML model is established here. The study suggests that future work should focus on expanding datasets and improving preprocessing and exploring hybrid models that can leverage the best both ML and DL approaches for achieving better performance.

Future Work

Based on the results obtained from this research, further studies should be directed toward the limitations of machine learning and deep learning models as applied to signal classification. For example, efforts should be devoted to increasing the dataset for improving the generalizability of deep learning architectures such as DNNs and LSTMs. Advanced pre-processing techniques, including feature extraction as well as normalization, will be among the key factors having a strong potential to contribute effectively to these models' enhancing performance. Also, doing hyperparameter refinement and finding the optimum model architectures becomes a crucial step to realizing deep learning methods in full. As such, another very promising research area for further study in this field is GANs. GANs have demonstrated

Received: August 07, 2025

Volume 38 No. 11s 2025,

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

excellent promise in generating synthetic data that can help overcome some disadvantages of small datasets. This leads to high-quality synthetic signal data that can be applied for the enrichment of training sets, thus providing diversified yet comprehensive data for the training of deep models. This might significantly enhance the robustness properties of DL models and even provide better performance in binary classification tasks as well. Exploring hybrid models that merge the strength of ML and DL techniques may involve GANs in data augmentation and it leads to more effective and adaptive solutions for real-world applications. We invite collaborators from the communication engineering and cybersecurity fields to help us expand data collection and validate these findings in operational scenarios including defense, disaster management and next-generation wireless networks.

References

1. Arjoune, Y., Salahdine, F., Islam, M. S., Ghribi, E., & Kaabouch, N. (2020). A Novel Jamming Attacks Detection Approach Based on Machine Learning for Wireless Communication. *arXiv preprint arXiv:2003.07308*.
2. Varotto, M., Heinrichs, F., Schuerg, T., Tomasin, S., & Valentin, S. (2024). Detecting 5G Narrowband Jammers with CNN, k-nearest Neighbors, and Support Vector Machines. *arXiv preprint arXiv:2405.09564*.
3. Wu, Y., & Zhao, Y. (2018). Jamming Signals Classification Using Convolutional Neural Network. *ResearchGate*.
4. Shi, Y., Davaslioglu, K., Sagduyu, Y. E., Headley, W. C., Fowler, M., & Green, G. (2019). Deep Learning for RF Signal Classification in Unknown and Dynamic Spectrum Environments. *arXiv preprint arXiv:1909.11800*.
5. Varotto, M., Valentin, S., & Tomasin, S. (2024). Detecting 5G Signal Jammers Using Spectrograms with Supervised and Unsupervised Learning. *arXiv preprint arXiv:2405.10331*.
6. Arjoune, Y., Salahdine, F., Islam, M. S., Ghribi, E., & Kaabouch, N. (2020). A Novel Jamming Attacks Detection Approach Based on Machine Learning for Wireless Communication. *arXiv preprint arXiv:2003.07308*.
7. Varotto, M., Heinrichs, F., Schuerg, T., Tomasin, S., & Valentin, S. (2024). Detecting 5G Narrowband Jammers with CNN, k-nearest Neighbors, and Support Vector Machines. *arXiv preprint arXiv:2405.09564*.
8. Jorge, L., Silva, F., & Almeida, R. (2022). Adaptive learning in deep neural networks: An analysis using the Adam optimizer. Computational Intelligence and Neuroscience, 2022, Article 7849050.
9. Kumar, P., Verma, S., & Gupta, R. (2023). Music genre classification using machine learning: A review of techniques and data sets. Journal of Artificial Intelligence Research, 41(2), 320-333.
10. Liu, S., Wang, L., & Zhao, J. (2023). Best practices in data splitting for robust machine learning. International Journal of Machine Learning and Data Mining, 10(2), 92-104.
11. Mishra, S., Yadav, K., & Bhat, M. (2023). Understanding confusion matrix for model performance evaluation. Journal of Computational Science, 39, 238-249.

Received: August 07, 2025

Volume 38 No. 11s 2025,

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

12. Olson, J., Kumar, A., & Lee, H. (2022). Early stopping and overfitting prevention in machine learning models. Journal of AI and Data Science, 15(3), 401-410.
13. Sahoo, S., & Rao, S. (2023). Performance evaluation metrics for deep learning models: An overview. Neural Computing and Applications, 35(1), 45-56.
14. R. S. Lakshmi Balaji, N. Duraimuthuarasan and T. Yingthawornsuk, "Data Analytics and Machine Learning Approach for Tsunami Prediction from Satellite and Hydrographic Data,"2024 12th International Electrical Engineering Congress (iEECON), Pattaya, Thailand, 2024, pp. 1-6
15. Singh, R., & Patel, P. (2022). Confusion matrix for evaluating binary classifiers: A practical guide. Journal of Data Science and Analytics, 9(1), 65-78.
16. Wang, C., & Zhang, Y. (2023). Random search versus grid search: A comparative study on machine learning model tuning. Journal of Computational Intelligence, 44(3), 290-304.
17. Zhang, Y., Zhang, J., & Li, X. (2023). Optimizing machine learning algorithms with hyperparameter tuning techniques. Journal of Machine Learning Research, 18(3), 512-522.

Received: August 07, 2025