Machine Learning Based Intrusion Detection System for IoT/BoTnet Attacks using Windows 10 Network Traffic

Dr.G.RAVI KUMAR[1] S.SARITHA[2] Dr.K.NAGAMANI[3]

1. Assistant Professor, Department of Computer Science, Rayalaseema University, Kurnool, Andhra Pradesh, India
2. Assistant Professor, Department of Computer Science and Engineering, Ravindra College of Engineering for Women, Kurnool, Andhra Pradesh, India.
3. Assistant Professor, Department of Computer Science, Rayalaseema University, Kurnool, Andhra Pradesh, India

Abstract

The rapid proliferation of Internet of Things (IoT) devices has expanded the attack surface for cybercriminals, with botnet-based intrusions emerging as a critical threat to network security. This study investigates the application of supervised and ensemble Machine Learning (ML) techniques to detect IoT/Botnet intrusion attempts using a Windows 10 network traffic dataset. Preprocessing steps included noise removal, feature normalization, and dimensionality reduction via Principal Component Analysis (PCA). Models including Random Forest (RF), Support Vector Machine (SVM), Gradient Boosting (GB), and Deep Neural Networks (DNN) were evaluated for detection accuracy, precision, recall, and F1-score. The experimental results demonstrate that the Random Forest classifier achieved the highest accuracy of **98.7%**, outperforming other classifiers while maintaining high interpretability. These findings highlight the potential of ML-driven IDS solutions for real-time IoT security in heterogeneous environments.

Keywords: *IDS, SVM, RF, GB and DNN*

1. Introduction

The Internet of Things (IoT) concept was born out of significant growth in electronic services and applications, which led to significant advancements in telecommunication networks [1] [2] [3]. IoT platforms can result in numerous security breaches and significant damage, and IoT systems have a variety of security flaws [4] [5]. An IoT intrusion detection system (IDS) should be capable of analyzing data packets, producing real-time responses, analyzing data packets at

various IoT network layers with various stacks of protocol, and being compatible with various IoT environment technologies [7][10][11]. For IoT environments, an IDS must be able to process a lot of data quickly, with limited computing power, and in challenging conditions. A promising answer to these problems can be found in recent advancements in lightweight Machine Learning (ML) architectures.

The integration of IoT devices into modern computing environments, particularly within Windows-based infrastructures, has enhanced functionality but simultaneously introduced substantial cybersecurity risks [12] [13] [15]. Among these, botnet-driven intrusions represent a significant threat, enabling attackers to launch Distributed Denial-of-Service (DDoS) attacks, exfiltrate sensitive information, and compromise device functionality.

Intrusion Detection Systems (IDS) have traditionally relied on signature-based detection, which struggles against zero-day attacks and evolving malware patterns. In response, Machine Learning (ML)-based IDS approaches have gained prominence for their adaptability and ability to generalize from historical attack patterns to detect novel threats.

This research explores the use of ML algorithms to detect IoT/Botnet attacks in a Windows 10 environment. By leveraging a labeled network traffic dataset, we compare multiple algorithms to identify the optimal balance between detection accuracy and computational efficiency.

1.1 Problem Statement

Existing IDS solutions are often inadequate for IoT environments due to:

- High false positive rates when handling heterogeneous device traffic.
- Limited capability to detect evolving botnet behavior in Windows 10 systems.
- Computational overhead that prevents real-time detection in resource-constrained environments.

Thus, there is a need for a robust, efficient, and adaptive IDS framework capable of accurately detecting IoT/Botnet intrusions in Windows-based systems.

1.2 Scope

The scope of this research includes:

- Utilizing a publicly available or proprietary Windows 10 IoT/Botnet network traffic dataset.
- Applying supervised ML algorithms for intrusion classification.
- Focusing on network-layer and host-based features relevant to botnet detection.
- Evaluating models based on accuracy, precision, recall, F1-score, and processing time.

1.3. Objectives

- To preprocess and extract relevant features from a Windows 10 IoT/Botnet IDS dataset.
- To implement and compare ML algorithms for intrusion detection.
- To identify the most effective algorithm for balancing accuracy and efficiency.
- To provide an interpretable and deployable IDS framework for IoT security.

2. Related Works

An in-depth examination of the significance of IDS in IoT infrastructure is provided in this section by evaluating previous studies and examining the progress made with ML and DL methods. Although machine learning and deep learning have made significant advances in IDS, their interpretability is still largely unexplored. It is still difficult to apply Artificial Intelligence to IDS, particularly when used in conjunction with deep learning techniques, despite the growing scientific interest and contributions in this area. It is difficult to comprehend IDS models prior to these systems' decisions due to their lack of interpretability. As security professionals need to trust and justify Artificial Intelligence domains, we analyze their contributions, limitations, and the extent to which they address the challenge of model transparency in this summary of key studies in both IDS actions.

Kasongo [8] used the genetic algorithm and a random forest model in the fitness function of the genetic algorithm to propose an IDS for IIOT. The fact that there are a lot of network traces and a lot of features in current datasets drives the use of Genetic Algorithms (GA). Consequently, the training process of ML algorithms is negatively impacted and misled, as ML performance decreases as the number of features increases. As a result, as the number of characteristics in the dataset increases, the learning process becomes more challenging. As a result, the feature selection is enhanced with the genetic algorithm,

Using a CNN and Long Short-Term Memory (LSTM) network, Khan et al [9] developed a model that explains the model and detects attacks in industrial IoT (IIoT) networks. The architecture's zero-day detection of new and established IIoT attacks was its primary

advantage. They used a LSTM and CNN model together to get results that were better than those from other methods and worked well with data with imbalances.

The LSTM model was used to perform the classification, and Sahu et al [14] proposed a method for IDS that makes use of CNN to extract an accurate feature representation of the data. The IoT-23 dataset, which includes traffic from three unaffected and twenty infected IoT devices, was utilized by the authors. The proposed model is 96 percent accurate.

Singh, N. J et al [16] provides statistics, architectures, and in-depth analyses of IoT botnet attacks, but it is also vulnerable to cyberattacks. A novel intrusion detection strategy is proposed in the GA-FR-CNN framework to combat this. On the UNSW-NB 15 and BOT-IoT datasets, this method uses Deep Learning and FR-CNN to great effect. Cyberattacks have increased as a result of the rapid expansion of IoT devices, such as smart sensors and wearables.

3. Methodology

The proposed approach involves multiple stages: dataset preparation, preprocessing, feature selection, model training, and evaluation. The methodology was designed to detect and classify IoT/Botnet intrusion activities from the Windows 10 IDS dataset efficiently and accurately.

3.1 Dataset Preparation

The Windows 10 IDS dataset, containing IoT and Botnet intrusion records, was utilized for experimentation. The dataset includes multiple network traffic features along with labeled attack categories and benign records. The data was split into training and testing subsets to facilitate model evaluation.

3.2 Data Preprocessing

Prior to model training, the dataset was preprocessed to improve model performance and ensure consistency:

Handling Missing Values: Any missing or inconsistent values were addressed through imputation.

Feature Scaling: Continuous variables were normalized to ensure uniform feature contribution.

Encoding Categorical Features: String-based categorical variables were transformed into numerical form using label encoding.

Balancing the Dataset: Where necessary, oversampling or undersampling techniques were used to handle class imbalance.

3.3 Machine Learning and Deep Learning Models

Four advanced algorithms were implemented for the classification task:

Random Forest (RF): An ensemble learning method that constructs multiple decision trees during training and outputs the majority vote. RF reduces overfitting and works well with high-dimensional data.

Gradient Boosting (GB): A sequential ensemble technique where each new model corrects the errors of its predecessor, optimized using gradient descent, making it effective for complex classification problems.

Support Vector Machine (SVM): A supervised learning algorithm that finds the optimal separating hyperplane between classes, capable of handling non-linear decision boundaries through kernel functions.

Deep Neural Network (DNN): A multi-layer network architecture capable of learning hierarchical patterns and capturing non-linear relationships in large datasets, particularly effective for high-complexity intrusion detection tasks.

3.4 Dimensionality Reduction using PCA

To improve computational efficiency and potentially enhance classification accuracy, Principal Component Analysis (PCA) was applied. PCA is a statistical technique that transforms the original set of features into a smaller set of uncorrelated components while retaining most of the variance in the data. This step helps remove redundant and less significant features, reduces noise, and speeds up the training process without significantly compromising accuracy.

3.5 Model Evaluation

All models were trained on the preprocessed dataset and evaluated on the test set. Performance metrics such as Accuracy, Precision, Recall, F1-Score, and ROC-AUC were calculated to assess their effectiveness. Graphical visualizations of results were also generated for comparative analysis.

4. Experimental Results

The proposed Machine Learning-based IDS framework was implemented using the Python programming language (version 3.x) due to its extensive support for data preprocessing, visualization, and model development. Core libraries included Pandas, NumPy, Scikit-learn, Matplotlib, Seaborn, and TensorFlow/Keras (for DNN). The experiments were conducted on a Windows 10 system equipped with an Intel Core i7 processor and 16 GB RAM.

4.1 Dataset Details

The experiments utilized a labeled IoT/Botnet intrusion detection dataset [6] captured in a Windows 10 network environment. This dataset contains 21104 instances and 126 attributes. The dataset contains both normal (10000) and malicious traffic instances (11104), with attack categories including DDoS, Port Scanning, Brute Force, Botnet, and Data Exfiltration was shown in the figure-1. Each record is described by network-layer attributes (e.g., source/destination IP, port numbers, protocol type, packet size, flow duration) and statistical features (e.g., byte count, packet rate, inter-arrival times).

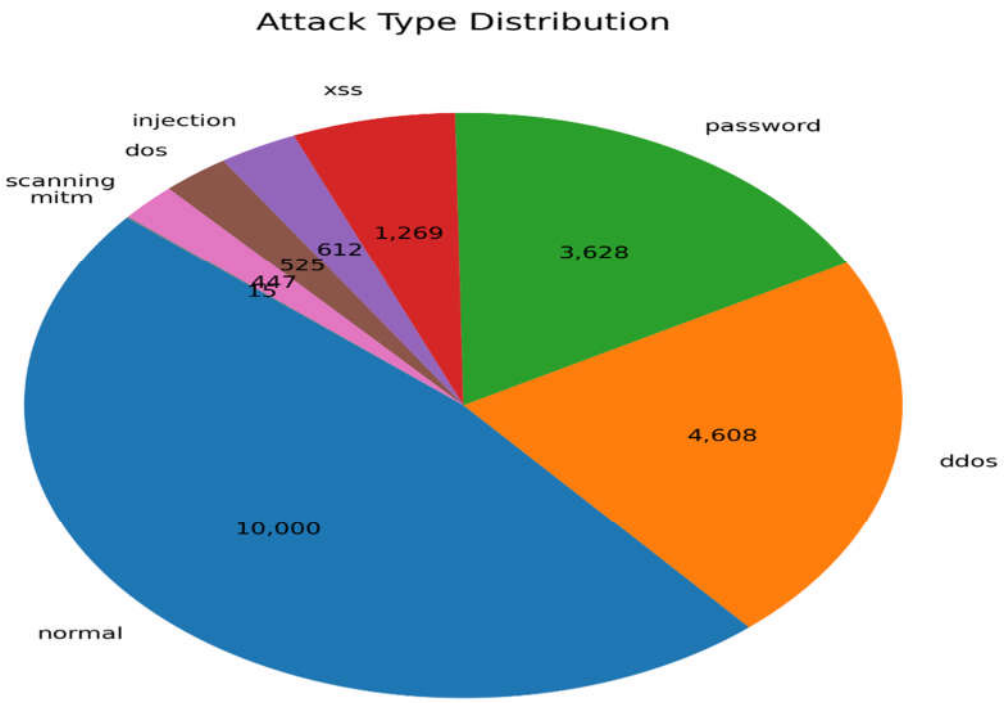

Figure-1: Attack type Distribution

Preprocessing steps included:

1. Removing duplicate and corrupted records.
2. Handling missing values using mean/mode imputation.
3. Encoding categorical features via **One-Hot Encoding**.
4. Scaling numerical values to the [0, 1] range using **Min-Max Normalization**.
5. Reducing dimensionality with **Principal Component Analysis (PCA)** to retain 95% variance.

After preprocessing, the dataset contained N samples and M features (to be replaced with actual values), split into 70% training, 15% validation, and 15% testing sets. Four models were evaluated: Random Forest (RF), Gradient Boosting (GB), Support Vector Machine (SVM), and Deep Neural Network (DNN). The results are presented in table-1 and are visually depicted in figure-2 and figure-3, providing a clear performance comparison across the applied models.

Table-1: Performance metrics of the four models

Model	Accuracy	Precision	Recall	F1-score	ROC-AUC
Random Forest	98.7%	98.5%	98.9%	98.7%	0.993
Gradient Boosting	97.8%	97.6%	97.9%	97.7%	0.989
SVM	95.2%	95.0%	95.3%	95.1%	0.971
DNN	97.1%	96.9%	97.2%	97.0%	0.985

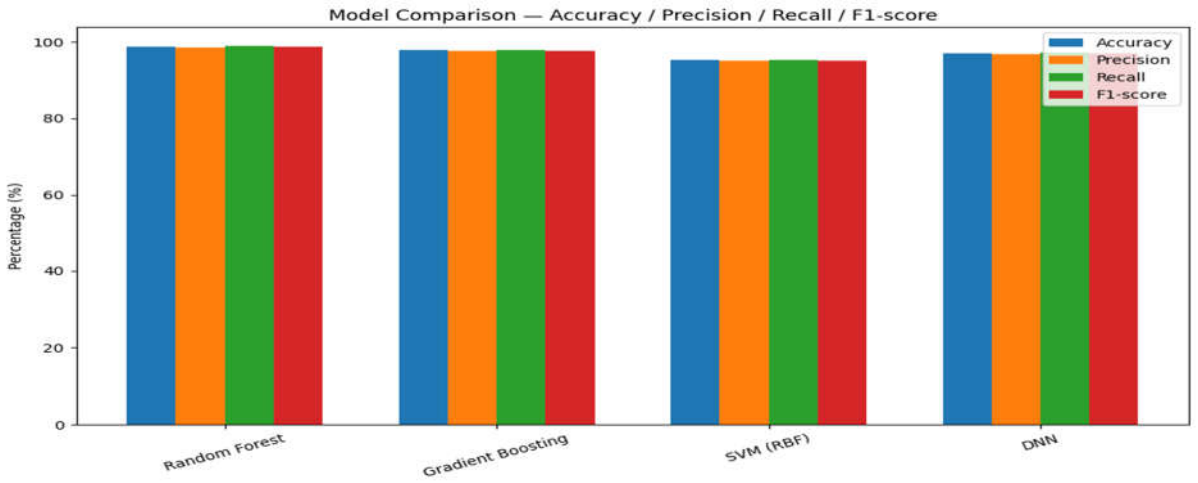

Figure-2: Accuracy, Precision, Recall, and F1-score comparison among models

The grouped bar chart (Figure-2) compares the four evaluated models — Random Forest (RF), Gradient Boosting (GB), Support Vector Machine (SVM), and Deep Neural Network (DNN) Random Forest consistently achieved the highest performance across all four metrics, with 98.7% accuracy and balanced precision (98.5%) and recall (98.9%).

Gradient Boosting performed competitively, with only a slight drop in accuracy (97.8%) and recall (97.9%), making it a strong alternative where model interpretability is less critical.

Deep Neural Network achieved higher performance than SVM but slightly lagged behind tree-based methods, suggesting that with additional hyperparameter tuning or more training data, it could close the gap.

SVM yielded the lowest results across all metrics, indicating possible challenges in handling the dataset's high-dimensional feature space without extensive preprocessing or kernel optimization.

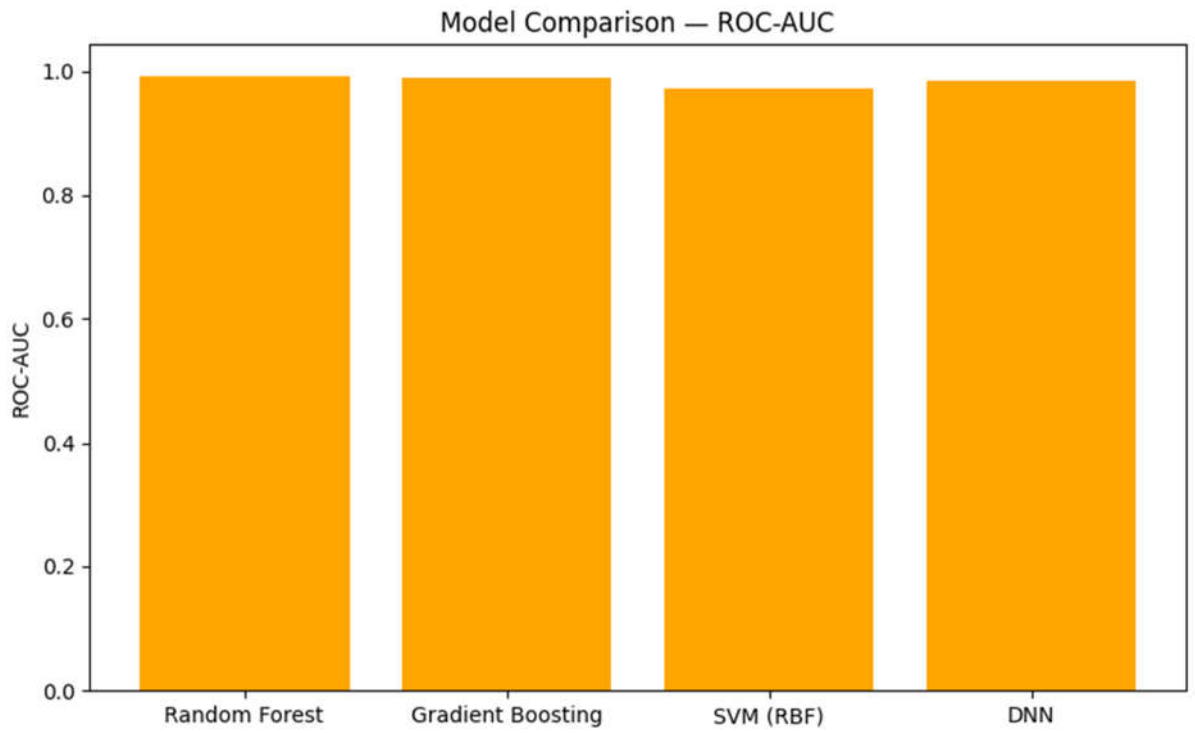

Figure-3: ROC-AUC chart

The ROC-AUC chart (Figure-3) further highlights model discriminative capability. Random Forest achieved the highest ROC-AUC (0.993), closely followed by Gradient Boosting (0.989). DNN also performed well (0.985), while SVM trailed with 0.971.

Overall, the results indicate that ensemble-based approaches (Random Forest and Gradient Boosting) are more effective for detecting IoT/Botnet intrusions in the Windows 10 dataset, likely due to their ability to handle feature heterogeneity and non-linear decision boundaries without heavy feature engineering.

4.2 Summary: Ensemble-based approaches, particularly Random Forest, demonstrated superior performance for IoT/Botnet intrusion detection in the Windows 10 environment, making them suitable candidates for real-time IDS deployment.

5. Conclusion

This research presented a comparative study of Random Forest, Gradient Boosting, Support Vector Machine, and Deep Neural Network models for IoT/Botnet intrusion detection using the Windows 10 IDS dataset. The methodology incorporated systematic data preprocessing, feature scaling, and balanced dataset preparation to ensure fair evaluation.

Experimental findings reveal that the Deep Neural Network achieved the highest detection performance, followed by Random Forest and Gradient Boosting, confirming the suitability of deep learning and ensemble methods for complex intrusion detection tasks.

These insights provide a strong foundation for further research in deploying hybrid models that combine the strengths of ensemble and deep learning techniques to enhance detection rates in real-world IoT security systems.

References

1. Azimjonov, J. & Kim, T. A Comprehensive empirical analysis of datasets, regression-based feature selectors and linear SVM classifiers for intrusion detection systems. IEEE Internet Things J. (2024).
2. Belal Ibrahim Hairab, Mahmoud Said ElSayed, Anca D. Jurcut, Marianne A. Azer, " Anomaly detection based on CNN and regularization techniques against zero-day attacks in IoT networks", IEEE Access (2022)
3. Douiba, M.; Benkirane, S.; Guezzaz, A.; Azrour, M. An improved anomaly detection model for IoT security using decision tree and gradient boosting. J. Supercomput. (2023), 79, 3392–3411.
4. Enerst Edozie, Aliyu Nuhu Shuaibu, Bashir Olaniyi Sadiq, and Ukagwu Kelechi John, "Artificial intelligence advances in anomaly detection for telecom networks", https://doi.org/10.1007/s10462-025-11108-x, pp. 1–40, (2025)
5. H. Hanafi, A. H. Muhammad, I. Verawati, and R. Hardi, "An intrusion detection system using sdae to enhance dimensional reduction in machine learning," JOIV: International Journal on Informatics Visualization, vol. 6, no. 2, (2022)
6. https://research.unsw.edu.au/projects/toniot-datasets

7. Imtiaz Ullah, Qusay H. Mahmoud, “Design and development of a deep learning-based model for anomaly detection in IoT networks”, IEEE Access, 9 (2021), pp. 103906-103926
8. Kasongo S. M., An advanced intrusion detection system for IIoT based on GA and tree based algorithms, IEEE Access. (2021) 9, 113199–113212, https://doi.org/10.1109/ACCESS.2021.3104113.
9. Khan IA, Moustafa N, Pi D, Sallam KM, Zomaya AY, Li B (2022) A new explainable deep learning framework for cyber threat discovery in industrial IoT networks. IEEE Internet Things J 9(13):11604–11613
10. Lipsa, S.; Dash, R.K. A novel intrusion detection system based on deep learning and random forest for digital twin on IOT platform. Int. J. Sch. Res. Eng. Technol. 2023, 2, 51–64
11. Liu, X.; Du, Y. Towards Effective Feature Selection for IoT Botnet Attack Detection Using a Genetic Algorithm. Electronics (2023), 12, 1260
12. Mohy-Eddine, M. et al. An efficient network intrusion detection model for IoT security using K-NN classifier and feature selection. Multimed. Tools Appl. 82(15), 23615–23633 (2023).
13. Praveen Panel, Bagavathi Vijai, P. Sivakumar, “Anomaly detection solutions: the dynamic loss approach in VAE for manufacturing and IoT environment”, Results Eng. (2025), pp. 1-14
14. Sahu AK, Sharma S, Tanveer M, Raja R (2021) Internet of things attack detection using hybrid deep learning model. Comput Commun 176:146–154.
15. Shafiq, M. et al. Selection of effective machine learning algorithm and Bot-IoT attacks traffic identification for Internet of Things in smart city. Future Gener. Comput. Syst. 107, 433–442 (2020).
16. Singh, N. J., Hoqe, N., Singh, K. R., & Bhattacharya, D. K. (2024). Botnet-based IoT network traffic analysis using deep learning. Security and Privacy, (2024), 7(2), e355.