International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue: 2 Volume No.9 March - April – 2025
DOI:10.46647/ijetms.2025.v09i02.050 ISSN: 2581-4621

Towards Intelligent Intrusion Detection: Botnet Attack Classification Using ML and DNN

Naga Leela Gangala[1, a)] **Sireesha P, Samyu J N, Vyshnavi G, Shabhareesh A** [2,3,4,5 b)]
[1]*Assistant Professor, Computer Science and Engineering, Srinivasa Ramanujan Institute of Technology, Anantapur, India.*
[2,3,4,5]*Computer Science and Engineering, Srinivasa Ramanujan Institute of Technology*

Abstract

The exponential growth of IoT devices has significantly increased the risk of botnet attacks, which exploit vulnerabilities in networked systems to execute malicious activities. Traditional botnet detection approaches often fail to adapt to the dynamic nature of these attacks. The proposed approach enhances detection capabilities, making it more suitable for large-scale IoT networks. This project proposes a Machine Learning (ML) and Deep Neural Network (DNN)-based approach to detect and classify botnet attacks. Traditional Intrusion Detection Systems (IDS) rely on predefined rules and struggle with modern threats. To address this, the system uses the UNSW-NB15 dataset, which contains real-world network traffic patterns. Various ML models, including Random Forest, Decision Tree, Naive Bayes, SVM, KNN, XGBoost,,Extra Trees Classifier and a DNN, are evaluated based on accuracy, precision, recall, F1-score, and execution time.The results show that XGBoost performs best with 94.02% accuracy, followed by Random Forest (93.29%), Extra Tree Cl
assifier (93.14%), Decision Tree (92.44%), and DNN (90.04%). These models excel in classifying five categories: Normal, Exploits, Fuzzers, Reconnaissance, and Generic attacks. This project highlights the potential of AI-driven solutions to provide scalable, accurate, and automated botnet attack detection, contributing to stronger and smarter network defense systems.

Keywords: Botnet Attacks, IoT, Machine Learning, Deep Neural Networks, XGBoost, Random Forest, Fuzzers

Introduction

In recent years, the internet has transformed into an indispensable infrastructure that connects people, organizations, and systems around the globe. While this interconnectedness has opened up new frontiers of communication, commerce, and automation, it has also introduced unprecedented vulnerabilities in the digital ecosystem. Among the most concerning threats in this space are botnets—networks of infected devices under the control of a malicious actor, commonly referred to as a "botmaster." These botnets are often used for a range of nefarious activities, including distributed denial-of-service (DDoS) attacks, data theft, phishing, and spam campaigns. The scale and stealth of botnet attacks make them particularly dangerous, as they can exploit both known and zero-day vulnerabilities and can often go undetected for long periods.

Traditional cybersecurity systems, such as signature-based intrusion detection systems (IDS) and rule- based firewalls, are rapidly becoming inadequate in the face of such complex and adaptive threats. These conventional methods rely heavily on predefined signatures and manually crafted rules, which may not be capable of detecting new or slightly modified attacks. In contrast, attackers continually evolve their methods, developing polymorphic malware and sophisticated evasion techniques that allow them to bypass traditional defenses. This growing mismatch between the capabilities of attackers and the limitations of legacy defenses has created an urgent need for more intelligent and adaptive security mechanisms.

In this context, machine learning (ML) and deep learning (DL) have emerged as powerful tools for cybersecurity. These approaches allow systems to learn from historical data, identify complex patterns, and generalize to unseen behaviors, making them particularly effective in detecting previously unknown threats. By leveraging ML/DL models, it is possible to analyze massive volumes of network traffic and detect subtle deviations from normal behavior, which could indicate the presence of a botnet or other forms of cyberattacks. Unlike traditional systems, ML-based models do not rely solely on prior knowledge of attack signatures, making them more robust against novel threats.

This project aims to apply a combination of machine learning and deep learning techniques to the problem of botnet attack detection, using the UNSW-NB15 dataset—a comprehensive and well-structured network intrusion detection dataset developed at the Australian Centre for Cyber Security. The dataset includes a wide range of network traffic data, both normal and malicious, and is labeled according to different types of attacks. For the purpose of this study, the focus has been narrowed to five primary categories: Normal, Exploits, Fuzzers, Reconnaissance, and Generic. These categories were selected to reduce the complexity of the classification task while still retaining a realistic and challenging set of attack vectors that occur commonly in real-world scenarios.

Literature Review

Mudasir Ali et al., The paper proposes a hybrid machine learning model (ACLR) combining ANN, CNN, LSTM, and RNN for efficient botnet attack detection in IoT environments. Using the UNSW-NB15 dataset, the model achieves 96.98% accuracy, outperforming existing methods. Its robust performance enhances real- time botnet detection and cybersecurity[1].Shamshair Ali et al., The paper proposes a hybrid deep learning model combining LSTM Autoencoders and MLP for IoT botnet detection, achieving 99.77% and 99.67% accuracy on N-BAIoT2018 and UNSW-NB15 datasets. It outperforms existing methods, including for zero- day attacks. The study highlights scalability and security challenges, suggesting future improvements with decentralized AI and enhanced encryption[2]. Tamara Zhukabayeva et al., The paper presents a machine learning-based approach for detecting botnet attacks in IoT networks using Random Forest and XGBoost algorithms. The study demonstrates high accuracy (99.21% for Random Forest and 99.18% for XGBoost) in identifying botnet activity using the N-BaIoT dataset. The findings emphasize the effectiveness of machine learning in enhancing IoT security and suggest further testing and development for real-world applications[3]. Mrutyunjaya Panda et al., The paper proposes a machine learning and deep learning-based approach for IoT botnet detection using optimized feature selection. Evaluating multiple classifiers on UNSW-NB15 and an imbalanced IoT-Botnet dataset, the scatter search-based DMLP achieves 100% accuracy with minimal computational cost, demonstrating its effectiveness in securing IoT networks[4]. Velamala Kula Sekhar, et al. The paper explores IoT botnet attacks, emphasizing their risks and the need for effective detection methods. It discusses network traffic analysis, anomaly detection, and network topology analysis to identify malicious activity. Given IoT devices' security vulnerabilities, the study reviews past research and future directions to enhance botnet detection, aiming to develop secure IoT systems[5]. Mousa AL-Akhrasm, et al. The paper examines botnet detection using machine learning-based IDS, addressing dataset noise with RENN, Explore, and DROP5 filters on IoTID20, N-BaIoT, and MedBIoT datasets. While RENN and DROP5 effectively reduced data size, DROP5 struggled with injected noise. N-BaIoT delivered the highest accuracy across models[6]. Rajasree Vennapureddy, et al. The paper systematically reviews IoT botnet detection techniques using ML and DL from 2014 to 2023. It highlights IoT security risks and evaluates various models based on performance metrics. Findings show that while ML and DL perform similarly, traditional ML struggles with real-time monitoring, timely detection, and adapting to new attack methods[7]. Mustafa Alshamkhany, et al. This paper investigates machine learning for detecting Botnet attacks using the Bot-IoT and UNSW datasets. Four classifiers—Naïve Bayes, K-NN, SVM, and Decision Trees—were trained on 82,000 UNSW-NB15 records. The Decision Trees model achieved the best performance with

99.89% accuracy, 100% precision, recall, and F-score, proving highly effective for Botnet detection[8].

Proposed Methodology

The proposed system is designed to detect and classify botnet attacks using the UNSW-NB15 dataset through the application of advanced machine learning and deep learning models. The system utilizes models such as Logistic Regression, Decision Tree (DT), Random Forest (RF), Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Gaussian Naïve Bayes (GNB), XGBoost, Extra Trees Classifier, and Deep Neural Network (DNN).

The primary objectives of the project are:

1. Noise Removal & Data Filtering: Filtering the dataset by removing irrelevant and noisy data to enhance prediction accuracy.
2. Botnet Attack Classification: Classifying network traffic into one of five categories—Normal, Exploits, Fuzzers, Reconnaissance, and Generic—based on the selected top 25 features.

A user-friendly web application was developed to streamline the detection process, allowing users to upload traffic data, view predictions in real time, and visualize model performance.

The process begins with dataset acquisition, followed by pre-processing steps like cleaning, feature selection, and transformation. Models are trained using the filtered and balanced dataset, and their performance is evaluated using accuracy, precision, recall, and F1-score. XGBoost was ultimately selected for deployment due to its high accuracy and efficiency.

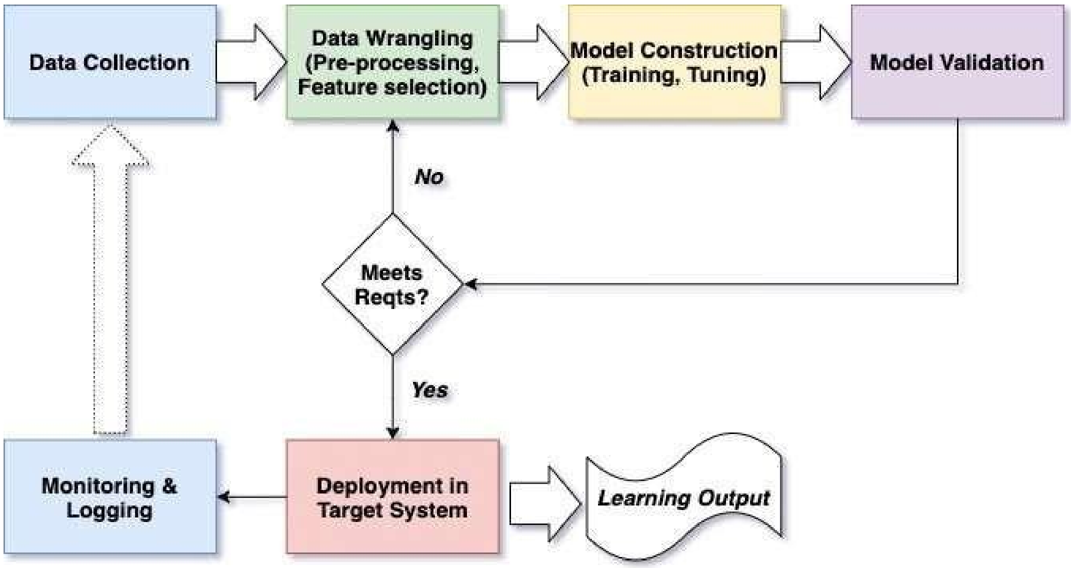
Data Collection
Data Wrangling (Pre-processing, Feature selection)
Model Construction (Training, Tuning)
Model Validation
No
Meets Reqts?
Yes
Monitoring & Logging
Deployment in Target System
Learning Output

Figure 1 Block Diagram

Dataset Specification

We utilize the **UNSW-NB15 dataset**, a widely accepted benchmark dataset for network intrusion detection, containing a mix of normal and attack traffic. The dataset comprises 49 features representing network traffic characteristics such as flow duration, source and destination bytes, packet rate, and protocol type. These attributes provide a comprehensive view of network activity, aiding in detecting botnet behavior. The project focuses on five attack categories based on relevance and frequency:

These categories provide sufficient diversity for robust multi-class classification while maintaining model performance.

1. Normal
2. Exploits
3. Fuzzers
4. Reconnaissance
5. Generic

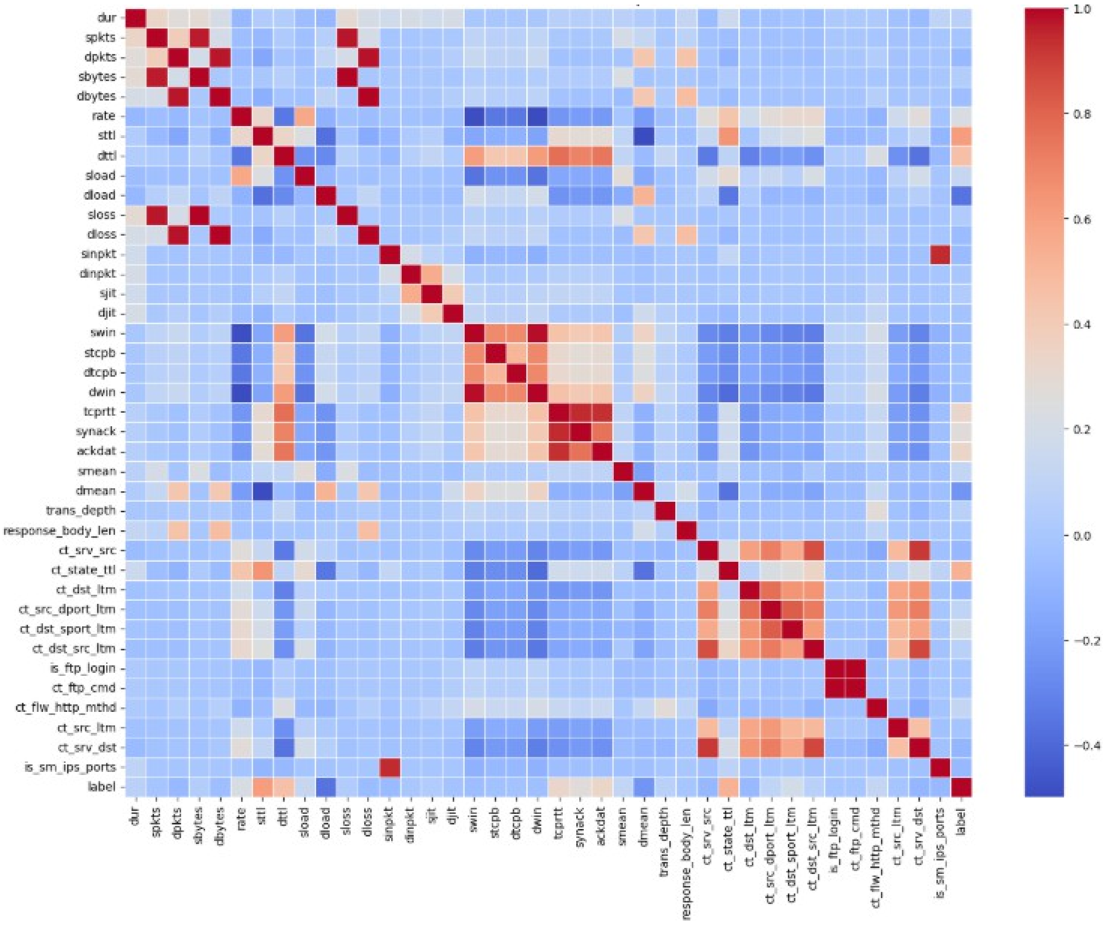

Figure 2: Distribution of Dataset Labels

Preprocessing

The gathered data should be cleaned and preprocessed to get rid of noise, artefacts, and outliers. Standardise or normalise the data to guarantee that the scale of the features is constant.This stage ensures that the data is clean, consistent, and structured for effective feature extraction and classification. The key steps involved in preprocessing sensor data for mental stress detection are Noise Removal, Missing Data Handling, Outlier Detection.

Feature Selection

Feature selection plays a crucial role in optimizing the performance of the botnet detection system by reducing dimensionality and computational overhead. With over 40 features in the dataset, not all contribute significantly to the model’s accuracy. To address this, the SelectKBest method with the f_classif scoring function was employed to identify and retain the top 25 most relevant features based on their statistical relationship with the target variable, *attack_cat*. This technique improves model generalization, eliminates noise and redundancy, speeds up training, and ensures efficient use of storage and processing resources.

Model Training and Evaluation

To evaluate the effectiveness of botnet detection, a diverse set of machine learning and deep learning algorithms were applied to the preprocessed dataset:

1. Decision Tree

Captures simple decision rules and performs well on datasets with clear logical splits.

2. Random Forest

An ensemble method that combines multiple decision trees, offering robustness to overfitting and noise. Achieved 93.29% accuracy.

3. Support Vector Machine (SVM)

Effective in high-dimensional spaces and well-suited for binary or multi-class classification using kernel tricks.

4. K-Nearest Neighbors (KNN)

A distance-based method used for simple classification, though it is computationally expensive for large datasets.

5. Gaussian Naïve Bayes (GNB)

A probabilistic classifier based on Bayes' theorem, which assumes feature independence and performed less effectively due to the complex feature interdependencies.

6. XGBoost Classifier

A high-performance boosting algorithm that sequentially builds trees to minimize error, achieving the highest accuracy of 94.02% in this project.

7. Extra Tree Classifier

Builds randomized decision trees to increase variance and diversity among ensemble members. Reached 93.14% accuracy.

8. Deep Neural Network (DNN)

A multi-layered architecture capable of modeling complex non-linear relationships. Demonstrated 90.45% accuracy and handled class imbalance well.

1. Evaluation Metrics:

1. **Accuracy**: Accuracy is a commonly used metric that measures the proportion of correct predictions made by the model, encompassing both true positives (TP) and true negatives (TN). It is defined as the ratio of the total correct predictions to the total number of predictions made.

Formula:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

2. **Precision**: Precision, also referred to as Positive Predictive Value, quantifies the accuracy of the model's positive predictions. It is defined as the proportion of true positive predictions among all instances that the model has predicted as positive.

Formula:

$$\text{Precision} = \frac{TP}{TP + FP}$$

3. **Recall:** Recall is a metric that measures the model's ability to identify all relevant instances in the dataset. It is the proportion of true positives among all actual positive instances.

Formula:

$$\text{Recall} = \frac{TP}{TP + FN}$$

4. **ROC-AUC**: Assesses classification performance by analyzing the trade-off between true positive and false positive rates.
5. **F1-Score**: The F1-score is the harmonic mean of precision and recall, providing a balance between these two metrics. It is particularly useful when the dataset is imbalanced and both false positives and false negatives need to be minimized.
Formula:

$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

6. **Computational Efficiency**: Evaluates the model’s training time and resource utilization, crucial for real- time deployment in IoT environments.
7. **Confusion Matrix Analysis**: Provides insights into the classification accuracy of each attack type, helping to fine-tune the model further.

Results

The proposed model is compared against standalone ANN, CNN, LSTM, and RNN models. Among the classical machine learning models, XGBoost achieved the highest accuracy of 94.02%, making it the best- performing model. It demonstrated excellent generalization and computational efficiency, making it suitable for real-time applications. Random Forest and Extra Trees Classifier also performed well, with accuracies of 93.29% and 93.14%, respectively. Simpler models like Logistic Regression and Naïve Bayes lagged behind due to limited capacity in modeling non-linear attack patterns. These results validate the effectiveness of machine learning and deep learning in botnet detection, offering a scalable and intelligent solution for enhancing cybersecurity systems with automated, real-time threat classification capabilities.

Conclusion

This research introduces a machine learning-based botnet attack detection system that leverages the UNSW-NB15 dataset to classify network traffic into five categories: Normal, Exploits, Fuzzers, Reconnaissance, and Generic. The dataset was preprocessed by cleaning, standardizing, and selecting the top 25 features using the SelectKBest algorithm to ensure optimal input quality. A comparative evaluation of multiple models revealed that XGBoost achieved the highest accuracy of 94.02%, outperforming other classifiers like Random Forest (93.29%) and Extra Trees Classifier (93.14%). The Deep Neural Network (DNN) also demonstrated strong performance with 90.45% accuracy, excelling in capturing complex patterns and handling class imbalance. Simpler models like Logistic Regression and Naive Bayes showed lower accuracy due to their inability to model non-linear relationships effectively.These results validate the effectiveness of machine learning and deep learning in botnet detection, offering a scalable and intelligent solution for enhancing cybersecurity systems with automated, real-time threat classification capabilities.

References

[1]. Mudasir Ali, Mobeen Shahroz, Muhammad Faheem Mushtaq, Sultan Alfarhood, Mejdl S. Safran, Imran Ashraf, "Hybrid Machine Learning Model for Efficient Botnet Attack Detection in IoT Environment", IEEE Access, vol. 12, pp. 40682-40699, 2024.
[2]. Shamshair Ali, Rubina Ghazal, Nauman Qadeer, Oumaima Saidani, Fatimah Alhayan, Anum Masood, Rabia Saleem, Muhammad Attique Khan, Deepak Gupta. "A novel approach of botnet detection using hybrid deep learning for enhancing security in IoT networks." Alexandria Engineering Journal, Vol. 103,
pp. 88–97, June 2024.
[3]. Tamara Zhukabayeva, Yerik Mardenov, Lazzat Zholshiyeva, Khu Ven-Tsen, Aigul Adamova, Nurdaulet Karabayev. "Enhancing IoT Security: Effective Botnet Attack Detection Through

Machine Learning." Procedia Computer Science, Vol. 241, pp. 421– 426, August 2024.

[4]. M. Alshamkhany, W. Alshamkhany, M. Mansour, M. Khan, S. Dhou, F. Aloul, "Botnet Attack Detection Using Machine Learning" ,IEEE International Conference on Information and Communication Technology (IIT), pp. 1-6, Nov. 2020.

[5]. I. Ali, A. I. A. Ahmed, A. Almogren, M. A. Raza, S. A. Shah, A. Khan, and A. Gani, ‘‘Systematic literature review on IoT-based botnet attack,’’ IEEE Access, vol. 8, pp. 212220–212232, 2020.

[6]. A. O. Prokofiev, Y. S. Smirnova, and V. A. Surov, ‘‘A method to detect Internet of Things botnets,’’ in Proc. IEEE Conf. Russian Young Res. Electr. Electron. Eng. (EIConRus), Jan. 2018, pp. 105–108.

[7]. H.-T. Nguyen, Q.-D. Ngo, and V.-H. Le, ‘‘A novel graph-based approach for IoT botnet detection,’’
Int. J. Inf. Secur., vol. 19, no. 5, pp. 567–577, Oct. 2020

[8]. Y. Meidan, M. Bohadana, Y. Mathov, Y. Mirsky, A. Shabtai, D. Breitenbacher, and Y. Elovici, ‘‘N- BaIoT—Network-based detection of IoT botnet attacks using deep autoencoders,’’ IEEE Pervasive Comput., vol. 17, no. 3, pp. 12–22, Jul./Sep. 2018.

[9]. W. Yassin, R. Abdullah, M. F. Abdollah, Z. Mas’ud, and F. A. Bakhari, ‘‘An IoT botnet prediction model using frequency based dependency graph: Proof-of-concept,’’ in Proc. 7th Int. Conf. Inf. Technol., IoT Smart City, Dec. 2019, pp. 344–352.

[10]. S. Almutairi, S. Mahfoudh, S. Almutairi, and J. S. Alowibdi, ‘‘Hybrid botnet detection based on host and network analysis,’’ J. Comput. Netw. Commun., vol. 2020, pp. 1–16, Jan. 2020.

[11]. A. Blaise, M. Bouet, V. Conan, and S. Secci, ‘‘Botnet fingerprinting: A frequency distributions scheme for lightweight bot detection,’’ IEEE Trans. Netw. Service Manage., vol. 17, no. 3, pp. 1701–1714, Sep. 2020.

[12]. M. Panda, A. A. Mousa, and A. E. Hassanien, "Developing an Efficient Feature Engineering and Machine Learning Model for Detecting IoT-Botnet Cyber Attacks" ,IEEE Access, vol. 9, pp. 102790- 102802, June 2021.