International Journal on Science and Technology (IJSAT)
E-ISSN: 2229-7677 ● Website: www.ijsat.org ● Email: editor@ijsat.org

Efficient Botnet Attack Detection using Machine Learning Models

Mr.Harish Reddy Gantla[1], Barapati Hasini[2], Perumbuduru Meghana[3], Gowrla Meghasree[4], Elethi Tharun[5]

[1]Assoc. Prof. Dept. of CSE, [2,3,4,5]UG Scholar,. Dept. Of CSE

[1,2,3,4,5]Vignan Inst. of Tech. & Sci.
Telangana, India
[1]harsha.rex@gmail.com,[2]hasini.barapati2003@gmail.com,[3]perumbudurumeghna@gmail.com,
[4]meghasreegowrla@gmail.com,[5]tharunkumar84@gmail.com

Abstract
As the number of cyber threats continues to rise, botnet attacks have become a significant concern in the field of network security. This study aims to create an effective botnet attack detection system using machine learning models, with a particular emphasis on boosting techniques like XGboost, ADAboost, CATboost, Gradient boosting, and LightGBM. These models are assessed using accuracy, precision, recall, and f1-score to identify the most effective approach. The system goes through a thorough process of feature engineering and data preprocessing to improve its ability to detect objects. By utilizing ensemble learning, the proposed framework enhances the accuracy of detecting botnet attacks while minimizing false positives and achieving high detection rates. The algorithm is a valuable tool for cybersecurity applications, as it can detect and prevent malicious activities.

Keywords: Botnet, Boosting, XG Boost, CAT Boost, Gradient Boosting, Light GBM.

1. INTRODUCTION

In the modern digital landscape, cybersecurity has emerged as a critical domain due to the growing number and intricacy of cyber threats. Botnets, a collection of vulnerable devices controlled by central attackers, can be employed to carry out various malicious activities, such as distributed denial of service (DDOS) attacks, data theft, and spam distribution.

The intricate and dynamic nature of the botnet's secrets and power generation capabilities make the discovery a challenging and intricate intellectual and adaptive solution. Specifically, the reliability and high prediction method, also known as ensemble training, demonstrated positive outcomes in abnormal systems and aggression.

Boost techniques like XGboost, ADAboost, CATboost, Gradient boosting, and LightGBM are frequently employed in diverse classification tasks and consistently demonstrate outstanding performance when dealing with imbalanced and intricate attack patterns. By combining predictions from multiple weak

learners, these methods create powerful classifiers that enhance both accuracy and the ability to generalize to new data. The objective of this research is to analyze and contrast multiple boost-based algorithms for identifying bot network attacks in machine learning. This research encompasses thorough pre-processing and technical procedures to enhance the quality of data records before training the model.

The main objective is to establish a dependable system for identification. Accurate identification structures can differentiate between normal and malicious movements with high precision and at least five detections. This provides a wide range of classification methods. This system can greatly minimize the risks associated with the attack of the botnet. The suggested approach is a flexible and innovative solution that supports the rapid growth of the cyber security industry.

2. LITERATURE REVIEW

1. Hybrid Machine Learning Model for Efficient Botnet Attack Detection in IoT Environment: **M. Ali, M. Shahroz, M. F. Mushtaq, S. Alfarhood, M. Safran, and I. Ashraf** (2024) The authors concluded that the new vaccine was safe and effective. Ashraf (2024) suggested a botnet detection system called ACLR, which is a combination of ANN, CNN, LSTM, and RNN models trained on the unsw-nb15 dataset. Their approach combines the best features of various deep learning models to improve detection accuracy, which is validated through cross-validation techniques to ensure dependability across different datasets. Despite its potential, the model's computational requirements and extensive training time may hinder its real-time deployment and ability to adapt to novel attack types that were not part of the training dataset. In addition, imbalanced network traffic processing is a problem in determining less common categories.

2. Efficient Hybrid Model for Botnet Detection Using Machine Learning: **P. Saxena and R. B. Patel** (2024) created a hybrid machine learning model that combines KNN, Random Forest, and Logistic Regression to identify botnet fraud, utilizing the advantages of pattern recognition, ensemble decision-making, and binary classification. Their model, applied to a botnet dataset, demonstrates a robust detection capability but faces limitations such as dependency on dataset quality, challenges with unseen botnet behaviours, increased computational complexity, and difficulty dealing with imbalanced attack data, which can impact the detection of rare threats.

3. Botnet detection: a review of machine learning and artificial intelligence strategies: **N. Mohamed (2024)** carried out an extensive analysis of machine learning and artificial intelligence techniques used for botnet detection, classifying them into supervised, unsupervised, deep learning, and reinforcement learning approaches. The paper discusses the advantages and evolution of each approach in handling complex botnet behaviours but highlights persistent challenges, including the need for extensive and diverse datasets, the dynamic evolution of botnets, high computational costs, and issues with adaptability and false positive rates, particularly when dealing with encrypted or obfuscated traffic.

4. Botnet attack detection using machine learning: Mustafa Alshamkhany, Wisam Alshamkhany, Mohamed Mansour, and Salam Dhou (2020) explored botnet detection using naïve bayes, KNN, SVM, and Decision Trees trained on the bot-IOT

and UNSW-NB15 datasets. Their investigation revealed that decision trees were more efficient than alternative approaches in classifying botnet-related activities. Nevertheless, restrictions include relying on

sub -sets of certain data that may not be applied to a wider range of actual scenarios. In particular, this cannot be applied to a wider range of real scenarios by determining the model.

5. Botnet detection using machine learning: k. B. Aswathi, s. Jayadev, J. (n.d.). Summary of our findings. Krishna, r. Krishnan, and G. Sarath (2021) performed a comparative study on botnet detection using supervised techniques (svm, random forest, knn) and unsupervised techniques (k-means, pca), alongside rnns for temporal analysis of network flow data. While the research highlights the capabilities of various models, it also notes limitations such as the reduced accuracy of unsupervised methods without clean labels, the high computational demand of deep learning models, and the challenges posed by feature scaling, data imbalance, and real-time application difficulties due to processing overhead.

3. METHODOLOGY

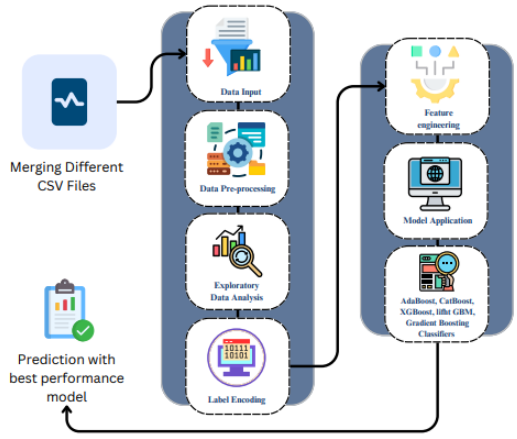

Figure 1: Pipeline for CSV Data

Integration and Prediction

The proposed methodology for botnet detection is designed around a structured and systematic pipeline that emphasizes data quality, model robustness, and precision in threat identification. The approach integrates comprehensive data preprocessing techniques with powerful ensemble learning algorithms, aiming to effectively detect botnet attacks while minimizing false alarms.

The process begins with data pre-processing, an essential stage that prepares raw data for efficient and meaningful analysis. The first step involves the removal of duplicate entries and null values, which ensures that the dataset is both clean and consistent. Duplicates can distort the learning process by overemphasizing specific patterns, while null values can impede the effectiveness of most machine learning algorithms. After organizing the data, the process of label encoding is employed to transform categorical features into numerical representations. Since most machine learning algorithms operate on numerical data, categorical variables are converted into numerical format using label encoding, where each category is assigned a unique integer value. This **adjustment enables** the models to **understand** and **assess** these **qualities effectively.**

Subsequently, normalization is carried out using the MinMax Scaler, which scales all numerical features to a fixed range, typically between 0 and 1. This step is crucial for algorithms sensitive to feature scales,

such as boosting models, as it ensures that no single feature dominates due to its magnitude. Normalization also speeds up the convergence of gradient-based learning algorithms and helps improve the stability of the entire model.

After pre-processing, the subsequent step is feature selection. This phase aims to streamline the data, enhance computational efficiency, and boost model accuracy by focusing on the most crucial features. Two types of features are systematically eliminated: features that are highly correlated with each other and features that have a single value. When different aspects of a model are closely related, it can result in multi-collinearity, which can make it challenging to understand the model's outcomes and decrease its reliability. In contrast, single-value features lack diversity and fail to provide any meaningful insights for classification tasks. The outcome is a cleaner and more concise dataset that facilitates faster training and improved generalization.

Following the process of feature selection, the dataset is split into two sections: 80% for training and 20% for testing. This 80-20 data split ensures that the models have enough data to learn from, while also allowing for evaluation on data that the models have not seen before. The test set acts as a standard to assess the models' ability to perform well in unfamiliar, real-life situations.

The foundation of the proposed methodology lies in the application of ensemble learning, employing five commonly used boosting algorithms: XGboost, ADAboost, CATboost, Gradient boosting, and LightGBM. This model is chosen because it has a track record of effectively handling structured data and has the potential to enhance the accuracy of the classification by continuously learning. These algorithms are built upon the concept of merging several weak learners, like decision trees, to form a strong predictive model.

- **XGBoost** (Extreme Gradient Boosting) is known for its scalability and superior performance, particularly on large and complex datasets.
- **Adaboost** (adaptive boosting) focuses on correcting errors made in earlier iterations by assigning greater importance to misclassified instances, resulting in improved classification accuracy as each subsequent model is trained.
- **CatBoost** is optimized for datasets with categorical features and eliminates the need for extensive preprocessing while maintaining high accuracy.
- Gradient boosting constructs models in a sequential manner, with each model refining the mistakes of its predecessor, making it particularly efficient for non-linear problems.
- **LightGBM** (Light Gradient Boosting Machine) is designed for high efficiency and speed, making it ideal for large-scale datasets with high dimensionality.

Each boosting model is trained using the prepared training dataset and then evaluated on the test dataset. The evaluation center analyzes various important aspects such as accuracy, accuracy, review, F1 and speed. Among these, special attention is given to achieving high precision and minimizing false positives, as these are crucial in the context of botnet detection. False positives can lead to unnecessary alerts, a heavier workload for security analysts, and a decrease in confidence in the detection system. Consequently, it is crucial to minimize such errors while still achieving a high level of accuracy in detection.

4. IMPLEMENTATION

The implementation of the machine learning system relies heavily on Python due to its simplicity, versatility, and the robust ecosystem of libraries that it offers for data science and machine learning. In this project, several key libraries were utilized, including pandas, SCIkit-Learn, XGBoost, CatBoost, and LightGBM. Each of these tools plays a critical role in different stages of the machine learning workflow, from data preprocessing to model training and deployment.

Data preprocessing is primarily managed through pandas, a robust data manipulation library in Python, which facilitates efficient loading, cleaning, and transformation of datasets. In addition to pandas, label-encoder from the SCIkit-Learn library is utilized to transform categorical variables into numerical representations, which is a prerequisite for most machine learning algorithms.

For the model training phase, the system leverages three advanced gradient boosting libraries: XGBoost, CatBoost, and LightGBM. These libraries are widely recognized for their performance and scalability, especially in structured data problems. XGBOOST is famous for its impressive speed and performance of the model, and LightGBM is famous for reducing rapid training and memory requirements. Using a combination of these algorithms, the system is suitable for practical use in various scenarios by creating an accurate and universal model.

Following the training process, the model is preserved using the salt water library, allowing it to be stored and utilized without the need for retraining. This is a crucial stage for introducing a model into real-world production conditions. The system is equipped with the best conveyor to receive reserves in real time and provide immediate predictions. This pipeline provides instant predictions for new data because the machine learning system can easily be incorporated into a larger application or user interface.

The system underwent comprehensive testing on modest hardware, specifically utilizing an Intel i5 processor and 8GB of RAM. Despite the absence of high-end computational resources, it delivered commendable performance, showcasing the efficiency and optimization of the underlying tools and techniques. This outcome highlights the lightweight nature of the solution, ensuring that it does not rely on specialized or powerful hardware to function effectively.

Consequently, the system proves to be highly accessible and practical for a broader user base, including those with limited hardware capabilities. Its adaptability and resource-efficient design make it suitable for deployment in a wide range of environments, from individual users to organizations with constrained technological infrastructure.

5. RESULTS & DISCUSSION

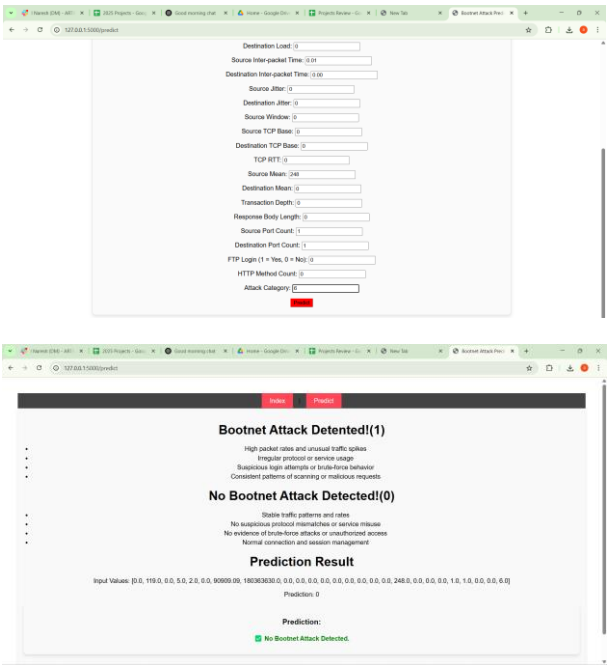

Figure 2: Results

All boosting models achieved high accuracy and fast prediction times, with XGBoost performing the best.
Evaluation Index:

- **F1-Indicator:** ~ 98%
- **Learning time model**: The system is suitable for environments with a large amount of data, such as distributed clusters or cloud platforms, which can be classified in real time to diagnose the system and detect threats.
- **Limitations:** In complex magazines due to loss of deformation, accuracy can be slightly reduced. Future studies can combine ensemble in deep training or adaptive tokenization methods.

REFERENCES

1. M. Ali, M. Shahroz, M. F. Mushtaq, S. Alfarhood, M. Safran and I. Ashraf, "Hybrid Machine Learning Model for Efficient Botnet Attack Detection in IoT Environment," in IEEE Access, vol. 12, pp. 40682-40699, 2024, doi: 10.1109/ACCESS.2024.3376400.
2. P. Saxena and R. B. Patel, "Efficient Hybrid Model for Botnet Detection Using Machine Learning," 2023 4th International Conference on Computation, Automation and Knowledge Management (ICCAKM), Dubai, United Arab Emirates, 2023, pp. 1-7, doi: 10.1109/ICCAKM58659.2023.10449643.
3. N. Mohamed, "Botnet Detection: A Review of Machine Learning and AI Strategies," 2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT), Kamand, India, 2024, pp. 1-6, doi: 10.1109/ICCCNT61001.2024.10724496.

4. Alshamkhany, Mustafa & Alshamkhany, Wisam & Mansour, Mohamed & Dhou, Salam & Aloul, Fadi. (2020). Botnet Attack Detection Using Machine Learning. 10.1109/IIT50501.2020.9299061.
5. M. Gupta, S. K. Kalhotra, H. Reddy Gantla, G. K. Arora, S. Vaishnodevi and A. Yashwanth Reddy, "Role of Machine Learning for Solving Satisfiability Problems and its Applications in Cryptanalysis," pp. 577-581, doi: 10.1109/ICACITE57410.2023.10182617.
6. Reddy, Harish., Kr, Sunil., Mantha, Shailaja, Goyal,Priya. , Jabeen,Asmath., Fatima,Shameem. , Mamodiya, Udit. Fusion of Real-Time Traffic and Environmental Sensor Data with Machine Learning for Optimizing Smart City Operations. Fusion: Practice and Applications, vol.,no.19 , Issue 2 , PP: 328-340, 2025 **DOI:** https://doi.org/10.54216/FPA.190224
7. K. B. Aswathi, S. Jayadev, N. Krishna, R. Krishnan and G. Sarath, "Botnet Detection using Machine Learning," 2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT), Kharagpur, India, 2021, pp. 1-7, doi: 10.1109/ICCCNT51525.2021.9579508.