IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed / Refereed journal ⁜ Vol. 12, Issue 2, February 2025

DOI: 10.17148/IARJSET.2025.12211

MACHINE LEARNING-BASED CYBER ATTACK DETECTION: A STUDY ON NETWORK TRAFFIC ANALYSIS AND THREAT PREDICTION

Anusha M[1], Apoorva P[2], Suraj H[3], Praveen M[4], Mr Achyuth Kasyap P[5]

Department of Artificial Intelligence and Machine learning, Dayanand Sagar academy of Technology and Management Bengaluru, India[1-5]

Abstract: This study explores how machine learning can be used to anticipate cybercrimes, with an emphasis on detecting attack techniques and possible offenders. The dataset used includes comprehensive records of criminal activity, including the characteristics of criminals and the methods used in attacks. The study compares these algorithms' performance in order to ascertain how accurate they are at forecasting the kind of cyberattack as well as the characteristics of the attacker.

The study also looks at the potential effects of a number of variables on the forecasts, including gender, income level, work position, and the seriousness of the crime. Additionally, it looks into how feature selection and preprocessing methods can improve model performance. This work's ultimate objective is to assist law enforcement organizations in improving their capacity to foresee and stop cyberattacks.

Keywords: Cyber Attack Prediction using Machine Learning involves cybersecurity concepts such as phishing, malware detection, data breaches, and intrusion detection systems (IDS). It utilizes machine learning techniques like logistic regression, random forests, SVM, and deep neural networks to analyze network traffic and detect anomalies. Key processes include feature extraction, dimensionality reduction (PCA, t-SNE), and data augmentation to enhance model accuracy. Performance evaluation metrics such as precision, recall, F1- score, and cross-validation are crucial for ensuring reliable threat detection. The project also involves tools and techniques like supervised learning, hyperparameter tuning, and behavior-based threat intelligence to improve predictive capabilities.

I. INTRODUCTION

2.1. Overview

Cybercrimes have risen as a result of technology's quick development and our growing reliance on digital platforms. These assaults put people, companies, and governments at considerable danger of suffering monetary losses, data breaches, and a decline in confidence in digital systems.

2.2. Overview of the Issue.

Malicious actions including phishing, malware, data breaches, denial-of-service attacks, and online fraud are all considered forms of cybercrime. Effectively combating these crimes is difficult due to their constantly changing nature. There is a significant gap in predicting and avoiding attacks since traditional cybersecurity solutions frequently concentrate on responding to problems after they occur. The necessity of instruments that can proactively detect such dangers and take action before they worsen is highlighted by this.

2.3 Technology and Its Potential

Artificial intelligence developments, especially in machine learning, provide up new avenues for tackling the problems associated with cybercrime. Large datasets can be analyzed, patterns may be found, and future occurrences can be predicted via machine learning algorithms. Predictive models may be created by utilizing these skills to assist law enforcement and cybersecurity experts in better anticipating and thwarting intrusions.

© IARJSET This work is licensed under a Creative Commons Attribution 4.0 International License

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed / Refereed journal ⁜ Vol. 12, Issue 2, February 2025

DOI: 10.17148/IARJSET.2025.12211

2.4 Challenges with Existing Tools.

Many cybersecurity solutions on the market today have trouble keeping up with the ever-evolving attack techniques. Tools that use signature-based detection are only able to identify known risks; they are unable to identify novel variants. Similar to this, behavior-based systems frequently struggle to establish precise baselines since contemporary networks are so complex, making it more difficult to spot anomalous activity.

2.5 Why is a New Solution Needed?

The limitations of current tools emphasize how crucial it is to take a proactive stance in order to avoid cybercrime. Predictive models based on machine learning can fill these gaps by: analyzing cybercrime data to find patterns and trends in order to forecast and stop future occurrences. using demographic and behavioral information to profile possible attackers. assessing the possible effects of different assaults in order to efficiently prioritize replies. constructing specialized defenses to bolster cybersecurity plans.

2.6 The Influence on Society

A effective predictive system for cybercrime prevention can yield significant social advantages. It is capable of:Safeguard private data to lower the chance of identity theft and monetary damages.By protecting vital infrastructure, you can make sure that vital services like healthcare and banking remain resilient.

2.7 Research Goal

The purpose of this study is to develop and assess machine learning models for accurate cybercrime prediction. Among the main goals are:Creating predictive models with methods such as Logistic Regression, Random Forest, Support Vector Machine, and KNN. analyzing the capabilities of each model and assessing its performance using pertinent indicators. investigating how elements like as targeted systems, attacker characteristics, and tactics affect forecast accuracy. addressing issues including discovery of new threats and data imbalances. The ultimate objective is to provide cutting- edge resources to law enforcement and cybersecurity teams so they can combat cybercrime proactively, lessening its negative effects on society and enhancing digital security in general.

II. REVIEW OF LITERATURE

The use of machine learning to examine network data and spot malicious activity has been the main focus of research on cybercrime prediction. High accuracy in differentiating between malicious and benign traffic has been demonstrated by algorithms such as Random Forest and Support Vector Machines (SVM). Research also emphasizes the usefulness of NetFlow data for anomaly detection, providing insights into network behavior while protecting privacy.

The significance of feature extraction and selection for improving model performance is emphasized in the literature. Methods such as recording botnet temporal activities and summarizing data in time windows have shown promise.However, because there are few datasets that accurately reflect the entire spectrum of harmful activity, it is still difficult to adjust models to changing threats. Furthermore, a major problem is data imbalance, where there are considerably fewer malicious cases than benign ones.

III. AIM OF THE PROJECT

3.1 Objectives

The main purpose of this project is to detect malware or botnet activity in NetFlow datasets using different Machine Learning techniques. The approach focuses on:

Identifying malicious or botnet traffic within NetFlow data. The system should handle datasets of any size, whether clean or containing malware, and classify the traffic as either normal or attack.

Comparing various Machine Learning models to determine the most suitable one for particular use cases.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed / Refereed journal ⁜ Vol. 12, Issue 2, February 2025

DOI: 10.17148/IARJSET.2025.12211

3.2 Methodology

1. Dataset Selection: The CTU-13 dataset was selected due to its accessibility and frequent use in similar studies. Key features such as StartTime, Duration, Proto, SrcAddr, and Label were examined.
2. Feature Extraction: NetFlow data was processed using a 2-minute time window with a 1-minute stride, extracting both categorical and numerical attributes.
3. Feature Selection: Dimensionality reduction was performed using methods like Pearson Correlation, Backward Feature Elimination, Random Forest, PCA, and t-SNE.
4. Algorithm Comparison: Logistic Regression, SVM, Random Forest, Gradient Boosting, and Dense Neural Network models were trained and compared.
5. Botnet Detection: The models were evaluated on the CTU-13 dataset for identifying botnet traffic, with performance assessed through the F1 score.

Main issue with Network Security Data: Network security data poses challenges like class imbalance, making it difficult for models to detect malicious traffic. Overfitting is a concern as network structure affects learning, while a structure-independent approach is preferred. The dynamic nature of networks adds complexity to identifying unknown botnets. Proper data analysis and methods like cross-validation are crucial to overcoming these issues.

IV. DATA ANALYSIS

4.1 CTU-13 Dataset

The CTU-13 dataset is a labeled collection frequently utilized in research for training botnet detection models, such as in the works of Garcia et al. (2014) and Rafal, Choras, and Keller (2019). Created by CTU University in 2011, it includes actual botnet traffic alongside normal and background traffic. This section will provide an overview of the dataset to identify key features for extraction and model training.

4.2 Feature extraction

NetFlow data often consists of categorical features, which can cause large matrix sizes and memory problems. To solve this, new features are derived from network traffic analysis studies. Data is summarized using time windows, as botnets typically show temporal patterns. The time window is set to 2 minutes, with a stride of 1 minute. For categorical data, features such as unique occurrences and normalized subgroup entropy are calculated. For numerical data, features like sum, mean, standard deviation, maximum, and median are extracted.

4.3 Feature selection

The filter method selects features based on statistical measures such as Pearson Correlation, which assesses the relationship between features and the target. Highly correlated features are removed to retain the most relevant ones. The wrapper method, on the other hand, tests feature subsets by training a model and adjusting the features iteratively until no further improvements are observed. Here, backward feature elimination is applied, starting with all features and progressively removing them to optimize the F1 score.

V. RESULTS

5.1 Metrics for Algorithm Evaluation

To evaluate the performance of algorithms, metrics like false positives (background communications mislabeled as botnets) and false negatives (botnet communications mislabeled as background) are crucial. Three key scores are used:

1. **Recall (R)**: Measures the proportion of actual botnet communications correctly identified.
2. **Precision (P)**: Measures the proportion of detected botnet communications that are truly botnets.
3. **F1 Score**: A harmonic mean of recall and precision, calculated as: F1=2×R·PR+PF1 = 2 \times \frac{R\cdot P}{R + P}F1=2×R+PR·P

For detecting malicious software, high recall is prioritized to minimize undetected botnets, even at the expense of precision. The goal is to maximize the F1 score while maintaining acceptable recall. Standard accuracy is unsuitable for imbalanced datasets; instead, weighted accuracy, aligned with F1 scores, is more relevant.

© IARJSET This work is licensed under a Creative Commons Attribution 4.0 International License

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed / Refereed journal ⁜ Vol. 12, Issue 2, February 2025

DOI: 10.17148/IARJSET.2025.12211

5.2 Overview of Algorithms Logistic Regression

Logistic Regression classifies data using a linear combination of features. Cross- validation determined optimal parameters as C=550C = 550C=550 and Weightnon- botnet=0.044\text{Weight}_{\text{non- botnet}}= 0.044Weightnon- botnet =0.044.

Support Vector Machine (SVM)

SVM employs kernels to transform data space and separate classes. Cross- validation optimized parameters for:

- **Linear kernel**: λ=10−9\lambda = 10^{-9}λ=10−9.
- **RBF kernel**: Best gamma value = 0.03567.
- **Polynomial kernel**: Degree = 2 (selected as best for botnet detection in CTU-13).

Random Forest

This method uses an ensemble of 100 decision trees to classify data.

Gradient Boosting

Gradient Boosting incrementally improves predictions using decision trees. Optimal parameters include:

- **Loss function**: Exponential (better than deviance).
- **Max depth**: 4.

Dense Neural Network

A simple dense neural network with two hidden layers (256 and 128 neurons) was tested. It uses ReLU activation (sigmoid for output), batch-normalization, no dropout, and a binary cross-entropy loss function. The model has 39,681 trainable parameters and was trained over 10 epochs with a batch size of 32, as shown by the learning curve.

5.3 Comparison and Generalization of Algorithms

Algorithm Performance

Models were trained on 2/3 of the dataset and tested on the remaining 1/3. Table 1 highlights that Random Forest, Gradient Boosting, and Dense Neural Network achieved the best F1 scores (0.97) in detecting botnets. Logistic Regression and SVM showed lower F1 scores (0.85) due to imbalances in precision and recall. Random Forest was preferred for its faster training and lower overfitting tendency, and it was tested on additional scenarios.

Random Forest Results Across Scenarios

Random Forest performed well in detecting botnets in 8 out of 13 scenarios (Table 2). However, smaller datasets resulted in poor scores for 5 scenarios. Statistical analysis (Table 3) confirmed that smaller datasets caused significant deviations, but average precision across scenarios remained near 95%, with mean recall above 50%.

Generalization Challenges

Testing the Random Forest Classifier on scenarios not included in the training set showed poor generalization (Tables 4–6). Training on one botnet scenario did not effectively detect other botnets due to differences in attack characteristics and dataset limitations. This emphasizes the need for botnet-specific training data for accurate detection.

Data Augmentation and Algorithm Comparison for Botnet Detection

Data Augmentation:
To address the small dataset sizes, bootstrap resampling was applied to expand training data by 10x and 30x.

- **10x Resampling**: Improved recall and F1 scores by approximately 10 points across scenarios, highlighting dataset size as a critical factor in botnet detection.

© IARJSET This work is licensed under a Creative Commons Attribution 4.0 International License

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed / Refereed journal ⁜ Vol. 12, Issue 2, February 2025

DOI: 10.17148/IARJSET.2025.12211

- **30x Resampling**: Performance gains plateaued, indicating limitations in the bootstrap method. Standard deviations remained unchanged since only training data was augmented.
- Algorithm Comparison for Challenging Scenarios:

Scenarios where Random Forest performed poorly (4, 5, 7, 11, and 12) were re-evaluated using alternative algorithms:

Logistic Regression:

- Struggled with limited and non- representative data.
- Poor F1 scores across all scenarios, indicating ineffectiveness in detecting complex botnet behaviors.

Gradient Boosting:

- Comparable results to Random Forest.
- Notable improvement in Scenario 5, but Scenario 12 continued to underperform.

Dense Neural Network:

- Demonstrated high performance in simpler scenarios but failed to generalize to complex botnet patterns.
- Statistical analysis using 50 random test datasets showed low precision and recall in difficult scenarios.
- Increasing epochs instead of resampling did not significantly enhance performance.

VI. CONCLUSION

This project focused on building and comparing models to detect botnets in real network traffic using Netflow datasets.

After extensive data analysis and feature extraction, no feature was deemed irrelevant for training. Algorithms such as Logistic Regression, SVM, Random Forest, Gradient Boosting, and Dense Neural Networks were evaluated.

Random Forest achieved over 95% detection accuracy in 8 out of 13 scenarios. For the 5 challenging scenarios, bootstrap resampling improved detection rates in scenarios 5, 7, and 11 to over 55%, while scenarios 4 and 12 remained difficult due to possibly inadequate feature representation or the need for more advanced models like recursive deep neural networks.

Future improvements include experimenting with feature extraction parameters, exploring more hyperparameters, training across multiple scenarios simultaneously, and applying unsupervised learning to detect botnet behaviors without labeled data.

REFERENCES

[1]. Bapat, R., et al. (2018). Identifying malicious botnet traffic using logistic regression. ***IEEE Systems and Information Engineering Design Symposium***, 266–271.
[2]. Brid, R. (2018). Regression Analysis. Retrieved from GreyAtom.
[3]. Claise, B. (2004). RFC 3954: Cisco systems NetFlow services export version 9. ***Internet Engineering Task Force***.
[4]. CTU University (2011). The CTU- 13 Dataset: Labeled Botnet, Normal, and Background Traffic. Retrieved from **Stratosphere IPS**.
[5]. Dsouza, M. (2018). Building botnet detectors using Python. Retrieved from PacktPub.
[6]. Ertoz, L., et al. (2004). The MINDS - Minnesota Intrusion Detection System.
[7]. Fruehwirt, P., Schrittwieser, S., & Weippl, E. R. (2014). Machine learning techniques for traffic classification and attacker profiling. ***Privacy, Security, Risk and Trust Conference***.
[8]. Gandhi, R. (2018). Support Vector Machines: Introduction. Retrieved from Towards Data Science.
[9]. Garcia, S., et al. (2014). Empirical comparison of botnet detection methods. ***Computers & Security***, 45, 100–123.

© IARJSET This work is licensed under a Creative Commons Attribution 4.0 International License