Review on Botnet attack detection using machine learning algorithm

Sandhya Gaikwad[1], Prof R. H. Ambole[2]
[1]*Student VPKBIET BARAMATI, Maharashtra India*
[2]*Prof, VPKBIET BARAMATI, Maharashtra, India*

***Abstract*—The proliferation of the Internet of things (IoT) devices has resulted in a steady rise in the volume of IoT-based assaults. One of the most serious IoT risks is the IoT botnet attack, which tries to commit actual, effective, and profitable cybercrimes. IoT botnets are collections of Internet-connected IoT devices that have been infected with malware and are managed remotely by an attacker. The Internet of things (IoT) systems have significant challenges in offering techniques to detect security vulnerabilities and assaults due to the rapid growth of threats and diversity in attack tactics. earlier identification would enable better IoT Botnet response proposals. As a result, it reduces the harm caused by possible assaults.**

***Index Terms*—Botnet, IoT, XGBoost, Decision tree, SVM, Accuracy, Precision, Recall.**

I. INTRODUCTION

Botnet attacks are a major cyber security threat, enabling malicious activities such as DDoS attacks, data theft, and spam dissemination. Detecting these threats is challenging due to their evolving techniques and stealthy behavior. Machine learning offers robust solutions for botnet detection by identifying patterns in network traffic and system behavior. IoT botnets are collections of Internet-connected IoT devices that have been infected with malware and are managed remotely by an attacker The Internet of things (IoT) systems have significant challenges in offering techniques to detect security vulnerabilities and assaults due to the rapid growth of threats and diversity in attack tactics. There are currently available detection techniques for such stages, thus if a DDoS assault performed by an IoT Botnet has already taken place, identifying the DDoS attack, and the IoT Botnet network by itself at this point is not too challenging. The increasing number of malicious attacks is a serious problem. The detection of botnet from the systems by using various algorithms and techniques is carried out.

II. ALGORITHM

1. Decision tree [1] - Decision trees are the most powerful and popular classification and prediction tools. A decision tree is a flowchart-like tree structure where each inner node specifies a test for an attribute, each branch represents the result of the test, and each leaf node (terminal node) contains a class label Gini Impurity Measures the likelihood of incorrect classification if a random instance is classified according to the distribution of labels:

$$Gini = 1 - \sum_{i=1}^{k} P_i^2$$

where Pi is the proportion of class i in the dataset, and k is the number of classes.

Entropy (Information Gain) Measures the information disorder in a dataset:

$$Entropy = \sum_{i=1}^{k} P_i \log_2(P_i)$$

2. Support Vector Machine - Support vector machines (SVMs) [12] are supervised machine learning algorithms used for both classification and regression. Also known as a regression problem, it is best suited for classification. The goal of the SVM algorithm is to find a hyperplane in the N-dimensional space that uniquely classifies the data points. When the data is not linearly separable, the kernel function must be used to map the data into the Vapnik Chervonenkis dimensional space. Three types of kernel function (Φ) exist: radial basis functions (RBFs), polynomials, and sigmoids. Using the appropriate kernel function for transforming the data is imperative for increasing the classification speed. The three kernel functions are described as follows.

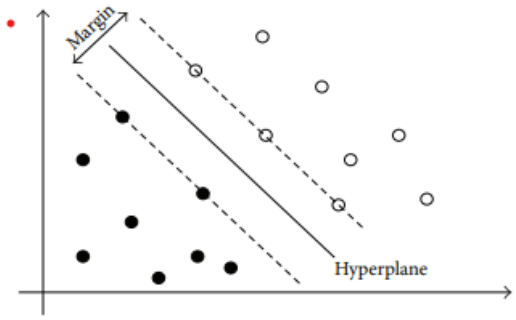

FIGURE 1: The optimal hyperplane.

RBF kernel:

$\phi(x_i - xj) = \exp(-\gamma|x_i - xj|$

Polynomial kernel:

$\phi\ (xi - xj) = (1 + xi \cdot xj)$

Sigmoid kernel:

$\phi\ (xi - xj) = \tanh\ (kxi \cdot xj - \delta)$

3. XGBoost - XGBoost [1] (Extreme Gradient Boosting) is a powerful and efficient implementation of gradient-boosted decision trees, an ensemble learning technique. It builds an ensemble of decision trees incrementally by optimizing a loss function. Each tree attempts to correct the errors of its predecessor by focusing more on the incorrectly predicted samples. Boosting Combines weak learners (trees) to create a strong learner.

$$Obj(\Theta) = \sum_{i=1}^{n} L(y_i, \bar{y}_i) + \sum_{k=1}^{k} \Omega f_k$$

Loss term L(yi,y^i): Measures prediction error.

Ω(fk) term penalizes complex trees.

Predictions are updated iteratively:

$\bar{y}_i'(t) = \bar{y}_i(t-1) + \eta f(x_i)$

where η is the learning rate.

3. Artificial Fish-Swarm Algorithm (AFSA) - The Artificial Fish Swarm Algorithm (AFSA**)** [8] is an optimization technique inspired by the social behaviors of fish, such as preying, swarming, and following. It is widely used for solving complex optimization problems because of its simplicity, flexibility, and ability to avoid local optima.

The preying behavior mimics the fish's tendency to move toward areas with higher food concentration (better solutions). Select a random position Xj within Visual.

If f(Xj)>f(Xi), the fish moves toward Xj

$$Xi' = Xi + rand.Step.(Xj - Xi)/|Xj - Xi|$$

where rand () is a random number in [0,1]

If f(Xj)≤f(Xi), randomly generate a new position and repeat.

The swarming behaviour models the fish's tendency to gather in groups. Compute the centre of mass Xc and average objective value in the vicinity

$$x_c = \frac{\sum_{k\in n} x_k}{N_n} \quad , \quad \bar{f} = \frac{\Sigma_{k\epsilon n} F(x_k)}{N_n}$$

If f¯>f(Xi) and the crowding condition ‖Xc−Xi‖ < δ is satisfied, move toward Xc

$Xi' = Xi + rand.Step.(Xj - Xi)/|Xj - Xi|$

The following behaviour simulates a fish following another with better fitness.

Identify the neighbour Xj with the best fitness (f(Xj)>f(Xi))

If the crowding condition ‖Xj−Xi‖<δ is satisfied, move toward Xj

$Xi' = Xi + rand.Step.(Xj - Xi)/|Xj - Xi|$

In random behaviour If no improvement is found in preying, swarming, or following, the artificial fish moves randomly:

Xi′=Xi + rand () · Step ·D

where D is a random direction vector.

4. Logistic Regression- Logistic Regression [1] is a statistical and machine learning technique used for classification tasks. It predicts the probability of a binary or categorical outcome based on one or more input features. Despite its name, logistic regression is not used for regression tasks but rather for classification. Logistic regression models the relationship between a set of independent variables (x) and a dependent binary outcome (y) using a linear equation:

$z = w^T x + b$

Sigmoid Function the linear output (z) is passed through a sigmoid function to map it to a probability P(y=1|x)):

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

5. Random Forest - Random Forest [7] is a popular machine learning algorithm related to supervised learning techniques. It can be used for both ML classification and regression problems. It is based on the concept of ensemble learning, which combines multiple classifiers to solve complex problems and

improve model performance. each iteration and makes the “best” leaf into a rule

For a dataset $(x_i, y_i)_{i=1}^n$where xi is the feature vector and yi is the target. A bootstrap sample is created by randomly selecting n instances with replacement. A decision tree Tk is trained on this sample using a random subset of features at each split. For classification prediction are given as

y^=mode {Tk(x): k=1, K}

6. K- Nearest Neighbor - K-Nearest Neighbors (KNN) [7] is a simple, non-parametric, and versatile supervised learning algorithm used for both classification and regression tasks. It makes predictions based on the similarity between data points. Sort the distances and select the k-nearest points. Predicts the output by:

$\hat{y} = mode(y_i : i \in N_k)$

7. Artificial neural network - The term "artificial neural network"[8][5] describes a biologically-inspired sub-area of artificial intelligence modeled after the brain. Artificial neural networks are computer networks usually based on biological neural networks that build the structure of the human brain. Just like the human brain has neurons connected to each other, artificial neural networks have neurons connected to each other at different layers of the network. These neurons are called nodes. The mathematical formulation of an ANN is described as follows:

The output of a single neuron is given by:

$y = f(\sum_{i=1}^{n} w_i x_i + b)$

8. Hidden Markov Model (HMM) - A hidden Markov model (HMM) is a statistical model that can be used to describe the evolution of observable events that depend on internal factors, which are not directly observable. Call for the observed event a `symbol' and the invisible factor underlying the observation a `state'.

9. Naive Bayes Classifiers-- Naive Bayes Classifier is a collection of classification algorithms based on Bayes' theorem. This is not a single algorithm, but a family of algorithms, all sharing common principles. Each pair of classified features is independent of each other

III. DETAILS OF PAPERS FOR BOTNET ATTACK DETECTION

1. Botnet Attack Detection in IoT Using Machine Learning by Khalid Alissa and Shadman Sakib, Tahir Alyas [1]**:** In this research A UNSW-NB15 dataset was cast off. An extreme gradient is one of many machine learning algorithms Extreme gradient is one of many machine learning algorithms boosting (XGBoost), which delivers extremely efficient and precise data. A subset of the results was chosen. 23 of the 39 useable characteristics were achieved through information gain. Various classifiers such as neural network, multi-logistic regression, nonlinear SVM, XGBoost, Naive Bayes, and random forest are trained and evaluated. From all the XGBoost outperformed with 88% test accuracy, followed by random forest which reported 87.89% accuracy. Similarly, numerous other studies have been conducted in Which experts adopt deep learning to detect intrusion.

The proposed methodology consists of feature extraction, traffic reduction, and a multi-layer network classifier to detect the botnet from the normal traffic. In the first phase, traffic is reduced by filtering the TCP control packet after the feature extraction. After extracting the features, the model is trained for botnet detection and legitimate traffic. Algorithm used are XGBOOST, Decision tree, Random Forest.

The evaluation parameter used are precision, F1-Score and Recall for calculating the model performance.

2. SUTMS: Designing a Unified Threat Management System for Home Networks [2]: The proposed integrated SUTMS design consists of flow detection, intrusion detection, and firewall core engines, as well as optional routing and log collection engines. The flow detection engine in SUTMS discovers anomalies and detects the active protocols, which are a basis for signature optimization in the SUTMS intrusion detection system (IDS). By ingesting Indicator of Compromise (IoC) feeds, the SUTMS firewall engine provides dynamic anti-bot protection. Evaluations indicate that SUTMS with IDS signature optimization can provide 99% accuracy with approx. 55% memory utilization reduction compared to traditional signature-based IDS.

Hardware Requirement: Raspberry Pi 4 with 8 GB of RAM, 32 Gigabyte (GB)of Secure Digital (SD) memory storage, and a Quad-core Cortex-A72 1.5GHz processor.

Software Requirement: We run the Ubuntu 22.04 LTS operating system on the Raspberry Pi 4 and the following software modules: a) GNU Ntop as flow detection engine for protocol and anomaly detection, b) Suricata version 6.0.4 as intrusion detection(IDS) engine for real-time scanning and detection/prevention

of malicious traffic, c) Linux IP tables as firewall engine) STIX/TAXII feeds from Anomaly for integrating anti-bot capabilities into the firewall engine, e) Routing and access point services, g) Apriori algorithm (optional component)that can be applied to IDS outputs.

Dataset: CICIDS2017 dataset. It is built up of a unique profiling mechanism, i.e., the B-Profile system. It allows the simulation of human behavioral traffic patterns and benign attacks.

It generates data from 25 users and commonly used protocols, e.g., HTTP, HTTPS, SSH, FTP, and email. The number of users and protocols closely match a typical home network. It includes common home network attacks, e.g. Dis-tributed DoS (DDoS), brute force attacks, HTTP/HTTPS exploitation, C&C communication, and DDoS. The data in pcap format is easier to manage and simulate than actual user traffic. The dataset is designed for IDS evaluation. However, the size and quality of the dataset are significant enough for stress testing and for evaluating the inspection capabilities of SUTMS IDS with other modules enabled.

Challenges: Effectiveness relies heavily on the quality of training data and the robustness of AI models. And also, data collection for monitoring raises questions about user privacy and data security.

3**.** Network Traffic Visualization Coupled with Convolutional Neural Networks for Enhanced IoT Botnet Detection [3]by DAVID ARNOLD, (Member, IEEE), MIKHAIL GROMOV, (Member, IEEE), AND JAFAR SANIIE, (Life Fellow, IEEE): This paper proposed firstly a novel network traffic visualization methodology for transforming network traffic into a visual format then Convolutional Neural Network model for classifying visualized network traffic. And then compare them on the accuracy base. The Dataset Used - N-BaIoT and IoT-23.

Algorithms used: Deep Neural Network: DLL are composed of interconnected multilayer perceptron layers in which output of each layer is determined by multiplying and adding a set of weights and biases respectively followed by activation function

- Layer 1 – Input: 115, Output: 115, Activation: tanh, Regularization: L1 Regularization
- Layer 2 – Input: 115, Output: 64, Activation: tanh, Regularization: None
- Layer 3 – Input: 64, Output: 32, Activation: tanh, Regularization: None
- Layer 4 – Input: 32, Output: 12, Activation: SoftMax, Regularization: None

4. Convolutional Neural Network:

The CNN is composed of series of Convolutional function with trained kernels against the input. In between each convolutional layer Max Pooling is used to reduce dimensionality and highlight important features in image Layer, Max Pooling is used to reduce dimensionality and highlight important features in our images. For classification, the image is flattened and passed through a set of multilayer perceptron layers. During testing, we adjusted the number of Convolutional, Max Pooling, and MLP layers along with the Convolutional Kernel Size, the Convolutional Kernel Stride, the Convolutional Activation Function, the Max Pooling Kernel Size, the Output size of each MLP layer, and the MLP Activation Functions. After testing, we arrived at the following model using an input 20 × 20×3 image:

- Convolutional Layer 1 – Kernel: 2 × 2×12, Stride: 2, Activation: ReLU
- Max Pooling Layer 1 – Kernel: 2 × 2
- Convolutional Layer 2 – Kernel: 2 × 2×16, Stride: 1, Activation: ReLu
- Max Pooling Layer 2 – Kernel: 2 × 2
- Flatten Layer
- MLP Layer 1 – Input:64, Output: 128, Activation: Sigmoid
- MLP Layer 2 – Input: 128, Output: 64, Activation: Sigmoid
- MLP Layer 3 – Input: 64, Output: 12, Activation: Sigmoid

5. Autoencoder:

The design of an Autoencoder is broken into two stages:

1) Encoder and 2) a Decoder. The Encoder compresses the input while the Decoder decompresses the image to their constructed class. Based on our testing, we found that the Autoencoder worked best when Convolutional Layers were used for compressing and decompressing the input. Our Encoder was composed of 2 Convolutional Layers, which decreased the input size to 75% and then 50%. The Decoder was composed of 2 Transpose Convolutional Layers. We modified the size of our Convolutional kernels, the Stride of our Kernels, and the Activation Function

during testing. The following describes our model's construction:

- Encoder:

Convolutional Layer 1 – Kernel: 2 × 2×9, Stride: 2, Activation: ReLU
Convolutional Layer 2 – Kernel: 6 × 6×24, Stride 1, Activation: ReLU

- Decoder:

Transpose Convolutional Layer 1 – Kernel: 6 × 6×9, Stride: 1, Activation: ReLU
Transpose Convolutional Layer 2 – Kernel: 2 × 2×3, Stride: 1, Activation: ReLU.

Performance is evaluated by using the performance metrics as accuracy, precision, recall, F1-score, and computational efficiency.

There are the challenges like Data dependency and computational overhead to visualize and train the CNN research includes the deep learning models ANN, DNN, and RNN as an interruption detection system. The dataset **UNSW-NB15** was established in diverse files and then categorized into binary classifications with deep learning models to measure abnormal patterns. In this study, the whole dataset was combined in a solo folder for models being tested more fairly than separately for a separate file. (e dataset attack families were then used as new labels, resulting in a multi-classification labelled dataset.

5. Latent Semantic Analysis and Graph Theory for Alert Correlation [4]: This paper proposes a system for IoT botnet detection that comprises two phases. The first phase aims to identify IoT botnet traffic, the input to this phase is the IoT traffic, which is subjected to feature selection and classification model training to distinguish malicious traffic from normal traffic. The second phase analyses the malicious traffic from stage one to identify different botnet attack campaigns. The second stage employs an alert correlation approach that combines the Latent Semantic Analysis (LSA) unsupervised learning and graph theory-based techniques. LSA, driven by Singular Value Decomposition (SVD). The proposed system utilizes LSA in the clustering phase to generate clusters of candidate botnet categories. These clusters are then mapped to stages of a botnet attack. SVD takes a matrix of numbers as input and decomposes it into three matrices that capture linear transformation, namely rotation, stretch, and rotation. The SVD formula is given by equation:

$$SVD = U\sum VT$$

SVD is a powerful mathematical tool that can be used to obtain a compact representation of an input matrix. The decomposition results in three matrices: a left singular matrix U, a diagonal matrix, and a right singular matrix VT. The matrix U is of dimensions m × r, where m is the number of rows in the input matrix, and r is a user-defined parameter. Matrix is a diagonal matrix of dimensions r×r, containing the singular values of the input matrix arranged in descending order. Finally, matrix VT is of dimensions r × n, where n is the number of columns in the input matrix. The proposed IDS operates in two phases. Equation to Equation illustrate different measures of interests from the graph-based theory approach from In-Degree through to the computation of betweenness centrality. Equation is a piecewise function that assigns 1 or 0 depending on whether the edge ei,j is from i to j, therefore 0 is assigned, otherwise if ei,j then 1 is assigned if ej,i, This function is utilized in Equation and Equation to differentiate between incoming edges and outgoing edges of a respective node for which the measures are computed.

$$\mathcal{F}(e_{i,j}) = \begin{cases} 0, \text{ if } e_{i,j} \in E \\ 1, \text{ otherwise} \end{cases}$$

TABLE 5. Evaluation of the detection of IoT botnet stages.

Attack Stage	Accuracy	Precision	TPR	FPR	F-Score
Scanning	99.9944%	100%	99.9944%	0%	99.9972%
C&C	99.9998%	100%	97.4089%	0%	98.6874%
DDoS	100%	100%	100%	0%	100%

Evalution parameter used : Detection accuracy,Efficiency,Graph visualization.

6. Hybrid Machine Learning Model for Efficient Botnet Attack Detection in IoT Environment: MUDASIR ALI, MOBEEN SHAHROZ , MUHAMMAD FAHEEM MUSHTAQ ,SULTAN ALFARHOOD, MEJDL SAFRAN , AND IMRAN ASHRAF [5]: The botnet detection is carried out using a hybrid deep learning model proposed in this research. The proposed approach is based on model stacking where the output of ANN, CNN, LSTM, and RNN is used for the final prediction. Additionally, it is estimated how well deep learning classification models perform when employed to analyze botnet attack detection using the UNSW-NB15 dataset. The preprocessing is carried out to remove null values and

handle categorical data using label encoding. To expedite the process, a variety of deep learning algorithms including ANN, CNN, LSTM, and RNN are applied.

Challenges: Generalization to detect new botnet variants.

Scalability for large-scale IoT networks.

IV. CONCLUSION

In conclusion, the reviewed literature highlights significant advancements in the detection of botnet attacks, emphasizing the evolution of techniques from traditional rule-based methods to more sophisticated machine learning and deep learning approaches. While methods such as anomaly detection, traffic pattern analysis, and behavior-based techniques have proven effective, challenges remain in addressing the dynamic and adaptive nature of modern botnets. Gaps in scalability, real-time detection, and accuracy in identifying novel attack patterns underline the need for continuous innovation.

REFERENCES

[1] Botnet Attack Detection using Machine Learning Mustafa Alshamkhany, Wisam Alshamkhany, Mohamed Mansour, Mueez Khan, Salam Dhou, Fadi Aloul

[2] SUTMS: Designing a Unified Threat Management System for Home Networks ASIF SIDDIQUI, BHASKAR P. RIMAL, (Senior Member, IEEE), MARTIN REISSLEIN 3, (Fellow, IEEE), DEEPAK GC 4, (Senior Member, IEEE), AND YONG WANG 1

[3] Network Traffic Visualization Coupled with Convolutional Neural Networks for Enhanced IoT Botnet Detection DAVID ARNOLD, (Member, IEEE), MIKHAIL GROMOV, (Member, IEEE), AND JAFAR SANIIE, (Life Fellow, IEEE)

[4] Latent Semantic Analysis and Graph Theory for Alert Correlation: A Proposed Approach for IoT Botnet Detection MOEMEDI LEFOANE 1 (Member, IEEE), IBRAHIM GHAFIR1, SOHAG KABIR 1, IRFAN-ULLAH AWAN1, KHALIL EL HINDI 2, AND ANAND MAHENDRAN3

[5] Hybrid Machine Learning Model for Efficient Botnet Attack Detection in IoT Environment MUDASIR ALI 1, MOBEEN SHAHROZ, MUHAMMAD FAHEEM MUSHTAQ SULTAN ALFARHOOD, MEJDL SAFRAN, AND IMRAN ASHRAF 4

[6] Appaswamy, Niranjan & M., Akshobhya & Shenoy, P. & K R, Venugopal. (2018). EKNIS: Ensemble of KNN, Naïve Bayes Kernel and ID3 for Efficient Botnet Classification Using Stacking. 1-6. 10.1109/ICDSE.2018.8527791

[7] M. Alshamkhany, W. Alshamkhany, M. Mansour, M. Khan, S. Dhou and F. Aloul, "Botnet Attack Detection using Machine Learning," 2020 14th International Conference on Innovations in Information Technology (IIT), 2020, pp. 203-208, doi: 10.1109/IIT50501.2020. 9299061..

[8] Kuan-Cheng Lin,1 Sih-Yang Chen,1 and Jason C. Hung2 Research Article Botnet Detection Using Support Vector Machines with Artificial Fish Swarm Algorithm

[9] Mahardhika, Yesta & Sudarsono, Amang & Barakbah, Ali. (2017). An implementation of Botnet dataset to predict accuracy based on network flow model. 33-39. 10.1109/KCIC.2017.8228455.

[10] Kuan-Cheng Lin, Sih-Yang Chen, Jason C. Hung, "Botnet Detection Using Support Vector Machines with Artificial Fish Swarm Algorithm", Journal of Applied Mathematics, vol. 2014, Article ID 986428, 9 pages, 2014. https://doi.org/10.1155/2014/986428

[11] N. Moustafa, B. Turnbull and K. -K. R. Choo, "An Ensemble Intrusion Detection Technique Based on Proposed Statistical Flow Features for Protecting Network Traffic of Internet of Things," in IEEE Internet of Things Journal, vol. 6, no. 3, pp. 4815-4830, June 2019, doi: 10.1109/JIOT.2018.2871719.

[12] A. Alharbi and K. Alsubhi, "Botnet Detection Approach Using Graph-Based Machine Learning," in IEEE Access, vol. 9, pp. 99166-99180, 2021, doi: 10.1109/ACCESS.2021.3094183.

[13] D. Nanthiya, P. Keerthika, S. B. Gopal, S. B. Kayalvizhi, T. Raja and R. S. Priya, "SVM Based DDoS Attack Detection in IoT Using Iot-23 Botnet Dataset," 2021 Innovations in Power and Advanced Computing Technologies (i-PACT),

2021, pp. 1-7, doi: 10.1109/iPACT52855.2021.9696569.

[14] D. Zhuang and J. M. Chang, "Enhanced PeerHunter: Detecting Peer-to-Peer Botnets Through Network-Flow Level Community Behavior Analysis," in IEEE Transactions on Information Forensics and Security, vol. 14, no. 6, pp. 1485-1500, June 2019, doi: 10.1109/TIFS.2018.2881657

[15] P, J., Shareena, J., Ramdas, A. et al. Intrusion Detection System for IOT Botnet Attacks Using Deep Learning. SN COMPUT. SCI. 2, 205 (2021). https://doi.org/10.1007/s42979-021-00516-9

[16] A STUDY OF MACHINE LEARNING CLASSIFIERS FOR ANOMALY-BASED MOBILE BOTNET DETECTION Ali Feizollah1, Nor Badrul Anuar2, Rosli Salleh3, Fairuz Amalina4, Ra'uf Ridzuan Ma'arof5, Shahaboddin Shamshirband