MACHINE LEARNING BASED DUAL-DETECTION FOR IOT BOTNET ATTACK PREVENTION

Syed Saniya
21N81A6210
Computer Science and Engineering
(Cyber Security)
Sphoorthy Engineering College,
Nadergul, Hyderabad,501510
syedsaniya2@gmail.com

Katroju Hemasree
21N81A6205
Computer Science and Engineering
(Cyber Security)
Sphoorthy Engineering College,
Nadergul, Hyderabad,501510
hemakatroju2020@gmail.com

Kkshitij Singh Thakur
21N81A6245
Computer Science and Engineering
(Cyber Security)
Sphoorthy Engineering College,
Nadergul, Hyderabad,501510
Kkshitijsingh23@gmail.com

R. Shirisha Reddy
21N81A6214
Computer Science and Engineering
(Cyber Security)
Sphoorthy Engineering College,
Nadergul, Hyderabad,501510
rshirisharshirisha@gmail.com

Dr. K. Subbarao
Ph.D (IIT-Kharagpur), M.Tech(CSE), M.Tech(ECE)
Professor and HOD (Resource and Development)
Sphoorthy Engineering College,
Nadergul, Hyderabad,501510

ABSTRACT:

Botnet attacks are among the most significant and complex cybersecurity threats in the Internet of Things (IoT) landscape, typically unfolding in multiple stages—from initial network scanning to full-scale Distributed Denial of Service (DDoS) attacks. Most current research emphasizes detection after IoT devices have already been compromised and are actively participating in malicious activities like DDoS. Moreover, many existing machine learning (ML) models for botnet detection tend to be overfitted to specific datasets, limiting their generalizability across different attack scenarios due to the variability in attack signatures.

To address these challenges, this study presents a more comprehensive approach by first generating a robust and diverse dataset that includes 33 types of scanning techniques and 60 variations of DDoS attacks. Additionally, samples from three publicly available datasets were selectively integrated to broaden the attack coverage and enhance model training.

Building on this, we introduce a dual-stage machine learning framework for early prevention and detection of IoT-based botnet threats. The first stage utilizes a deep learning model—ResNet-18—to identify suspicious scanning behavior indicative of an impending attack. The second stage employs a separate ResNet-18 instance to detect and classify ongoing DDoS activity. The proposed dual-model approach achieved an impressive performance with 98.89% accuracy, 99.01% precision, 98.74% recall, and an F1-score of 98.87%.

To validate the robustness of our method, additional ResNet-18 models were trained on three separate datasets and benchmarked against the proposed system. Results consistently showed superior detection and prevention capabilities, underscoring the effectiveness and adaptability of the dual-stage architecture in safeguarding IoT networks from botnet intrusions.

I. INTRODUCTION

The Internet of Things (IoT) has revolutionized the way devices interact, but the increasing adoption of IoT devices has also raised significant security concerns. Many IoT devices have limited security features, making them vulnerable to attacks. One of the most prevalent threats is botnet and Distributed Denial of Service (DDoS) attacks, where compromised devices are used to perform malicious activities.

Existing detection methods have limitations. Signature-based detection can only identify known threats, while anomaly-based detection shows promise in detecting unknown attacks. Machine learning approaches, particularly deep learning models, have demonstrated effectiveness in detecting anomalies.

To address these challenges, a two-fold approach can be employed. This approach involves detecting scanning activity to prevent botnet attacks and identifying DDoS attacks performed by compromised devices. By utilizing a deep learning model like ResNet-18, it's possible to detect scanning and DDoS attacks, providing a robust security solution for IoT networks.

The key contributions of this approach include creating a generic dataset with various scan and DDoS attacks and proposing a two-fold machine learning approach to prevent and detect botnet attacks in IoT networks. By adopting this approach, IoT devices and networks can be better protected from devastating cyberattacks.

II. LITERATURE REVIEW

Nguyen et al. proposed a graph-based approach to detect the IoT botnet via printing string information (PSI) graphs. The authors used PSI graphs to get high-level features from the function call graph and then trained a convolution neural network (CNN), a deep learning model, over the generated graphs for IoT botnet detection. Likewise, Wang et al. proposed an automated model named as BotMark. Their proposed model detects botnet attacks based on a hybrid analysis of flow-based and graph-based network traffic behaviors. The flow-based detection is performed by k-means, which calculates the similarity and stability scores between flows. While the graph-based detection uses the least-square technique and local outlier factor (LOF) which measures anomaly scores. Similarly, Yassin et al. proposed a novel method that compromises a series of approaches such as the utilization of the frequency process against registry information, graph visualization and rules generation. The authors investigated the Mirai attacks using the graph-theoretical approach. In order to identify similar and dissimilar Mirai patterns, the authors used directed graphs. The proposed approach only focuses on the Mirai attack.

Almutairi et al. proposed a hybrid botnet detection technique that detects new botnets implemented on three levels, i.e., host level, network level and a combination of both. The authors focused on focused HTTP, P2P, IRC, and DNS botnet traffic. The proposed technique consists of three components: host analyser, network analyser, and detection report. The authors used two machine learning algorithms, i.e., Naïve Bayes and a decision tree for traffic classification. Similarly, Blaise et al. proposed a bot detection technique named BotFP, for bot fingerprinting. The proposed BotFP framework has two variants, i.e., BotFP-Clus which groups similar traffic instances using clustering algorithms and BotFP-ML is designed to learn from the signatures and identify new bots using two supervised ML algorithms, i.e., SVM and MLP. Likewise, Soe et al. developed a machine learning-based IoT botnet attack detection model. The proposed model consists of two stages: a model builder and an attack detector. In the model builder stage, data collection, data categorization, model training and feature selection are performed step by step. While in the attack detector stage, the packets are first decoded and then the features are extracted in the same way as in the model builder phase. Finally, the features are passed to the attack detector engine where artificial neural network (ANN), J48 decision tree, and Naïve Bayes machine learning models are used for botnet attack detection.

Sriram et al. proposed a deep learning-based IoT botnet attack detection framework. The proposed solution specifically considered network traffic flows, which are further converted into feature records and then passed to the deep neural network (DNN) model for IoT botnet attack detection. Nugraha et al. evaluated the performance of four deep learning models for botnet attack detection by performing a couple of experiments. The experimental results revealed that CNN-LSTM outperformed all deep learning models for botnet attacks detection.

III. METHODOLOGY

A two-phase approach to IoT security provides a comprehensive defense mechanism against botnet threats. The prevention phase proactively identifies vulnerabilities in IoT devices, while the detection phase continuously monitors network traffic to detect and mitigate ongoing attacks. This dual-layered security mechanism enhances the resilience of IoT devices, ensuring they remain protected against both emerging and existing threats.

1. Prevention Phase: Identifying Vulnerable IoT Devices: The prevention phase involves a structured multi-step process that includes data collection, feature extraction, machine learning model training, and enforcement of preventive actions. By monitoring device metadata, security configurations, and network behavior patterns, the system can identify potential vulnerabilities and classify devices as secure or vulnerable. Preventive actions, such as firmware updates and security configurations, are taken to mitigate risks.

2. Detection Phase: Identifying Active Botnet Attacks: The detection phase actively monitors network traffic to identify ongoing botnet attacks and mitigate their impact in real time. By analyzing traffic volume, communication patterns, and protocol distribution, the system can detect anomalous behavior and trigger real-time attack mitigation measures. Machine learning models, including supervised and unsupervised learning approaches, are deployed to distinguish between normal and malicious network traffic.

3. Integration into a Cybersecurity Framework: The two-phase security approach is integrated into a broader cybersecurity framework, allowing centralized monitoring and response. The system employs adaptive learning mechanisms and Edge AI to enhance security and ensure low-latency threat responses. This approach significantly enhances cybersecurity resilience and provides a robust defense mechanism against botnet threats.

Future research should focus on improving model adaptability, reducing dependence on labeled datasets, and integrating emerging technologies like federated learning and blockchain-based authentication. As IoT devices continue to proliferate globally, the adoption of intelligent, automated security solutions will be essential in ensuring the long-term safety, reliability, and resilience of IoT networks worldwide.

IV. EXISTING METHODS

Several approaches have been proposed to detect IoT botnets, including graph-based, hybrid analysis, and machine learning-based methods. Nguyen et al used PSI graphs and CNNs to detect IoT botnets, while Wang et al proposed BotMark, a hybrid model combining flow-based and graph-based network traffic analysis. Other approaches include Yassin et al's graph-theoretical method for Mirai attacks and Almutairi et al's hybrid botnet detection technique using machine learning algorithms.

Machine learning-based approaches have also been explored, such as Blaise et al's BotFP framework, Soe et al's two-stage model using ANN, J48 decision tree, and Naïve Bayes, and Sriram et al's deep learning-based framework using DNN. Nugraha et al evaluated the performance of deep learning models, finding that CNN-LSTM outperformed others.

However, existing systems have limitations. They often focus on detecting scanning activity and DDoS attacks, but IoT botnet attacks can be more complex and not limited to these stages. This highlights the need for more comprehensive and adaptable detection methods.

PROPOSED SYSTEM

To overcome the limitations of traditional IoT security mechanisms, a two-fold machine learning-based approach is proposed, integrating both prevention and detection strategies. The prevention mechanism focuses on proactive security measures to mitigate botnet threats before they compromise IoT devices. Key components include:

Anomaly-Based Access Control: Implementing behavioral analysis techniques to identify and restrict unauthorized access attempts. 16 Enhanced Authentication: Utilizing multi-factor authentication (MFA) and cryptographic techniques to prevent unauthorized access. Secure Firmware Updates: Ensuring timely updates and patches to mitigate vulnerabilities that could be exploited by attackers.

Blockchain-Based Identity Management: Leveraging decentralized authentication to prevent identity spoofing and unauthorized device access.

Federated Learning for Security Training: Enabling IoT devices to collaboratively train security models while preserving privacy.

AI-Powered Traffic Filtering: Employing machine learning algorithms to analyze network traffic in real-time and block potential botnet activity.

Threat Intelligence Sharing: Using decentralized threat intelligence frameworks to allow IoT networks to learn from past security incidents.

Adaptive Firewalls: Deploying AI-enhanced firewalls capable of dynamic rule adjustments based on threat intelligence. Detection Mechanism. The detection mechanism employs machine learning models to identify ongoing attacks and compromised devices.

Key components include:

- **Deep Learning-Based Intrusion Detection Systems (IDS):** Utilizing convolutional neural networks (CNNs) and recurrent neural networks (RNNs) to detect malicious traffic patterns.
- **Real-Time Anomaly Detection:** Implementing unsupervised learning models such as autoencoders and clustering algorithms to identify deviations from normal network behavior.
- **Ensemble Learning Approaches:** Combining multiple classifiers (e.g., decision trees, support vector machines, and random forests) to enhance accuracy and reduce false positives.Behavioral Traffic Analysis: Monitoring network behavior using AI-driven models to detect command-and-control (C&C) botnet communications.
- **Automated Response System:** Deploying AI-driven automated response mechanisms to isolate infected devices and mitigate threats in real-time.
- **Graph-Based Threat Analysis:** Utilizing graph neural networks to map botnet structures and detect coordinated attack campaigns.
- **Zero-Day Attack Identification:** Implementing adversarial learning techniques to recognize previously unseen attack patterns.
- **17 Cloud-Based Threat Analytics:** Integrating with cloud-based security solutions for scalable, real-time botnet threat detection and mitigation.

V. IMPLEMENTATION

Step 1: Data Collection and Preparation

- Capture network traffic using tools like **Wireshark**, **tcpdump**, and logs from IoT devices.
- Generate a **custom dataset** containing:
  - **33 types of scanning attacks**
  - **60 types of DDoS attacks**

- Integrate samples from **three public datasets** to maximize diversity and coverage.

Step 2: Data Preprocessing

- Extract relevant **network and behavioral features** (e.g., packet sizes, IPs, flow duration).
- Clean the data by:
    - Handling **missing values**
    - **Removing outliers**
- Normalize the data using techniques like **Min-Max scaling** or **Standardization**.

Step 3: Two-Fold ML-Based Model Design

- Implement **ResNet-18**, a deep learning model, in two folds:
    - **Fold 1**: Train ResNet-18 to detect **scanning activities** to **prevent** botnet attacks at early stages.
    - **Fold 2**: Train another ResNet-18 model to detect **DDoS attacks** to **identify** active botnet infections.

Step 4: Feature Engineering and Model Training

- Use tools such as **TensorFlow**, **PyTorch**, and **Scikit-learn**.
- Perform:
    - **Model selection and tuning**
    - **Cross-validation**
- Evaluate using metrics like:
    - **Accuracy**
    - **Precision**
    - **Recall**
    - **F1-score**

Step 5: Real-Time Monitoring and Detection

- Implement real-time monitoring of network traffic via dashboards.
- Display results visually using **bar charts**, **status ratios**, and **detection logs**.
- Generate real-time **alerts and notifications** upon detecting suspicious activity.

Step 6: Web Application and User Interface

- Backend: **Python Django + MySQL**
- Frontend: **HTML, CSS, JavaScript**
- Users (remote and admin) can:
    - **Login/Register**
    - **Upload data**
    - **Trigger botnet detection**
    - **View detection reports and download logs**

Step 7: Output Generation and Reporting

- After prediction:
    - Display **detection results** (e.g., botnet present or not).
    - Allow users to **download predictions and reports**.
- Generate summary charts comparing detection accuracy across datasets.

Step 8: Evaluation & Comparative Analysis

- Train and test the model on **three different datasets**.
- Compare the performance of the two-fold approach against other ML models.
- Confirm robustness and generalizability across attack types and traffic patterns.

Step 9: Integration and Testing

- Conduct:
    - **Black-box testing**
    - **White-box testing**

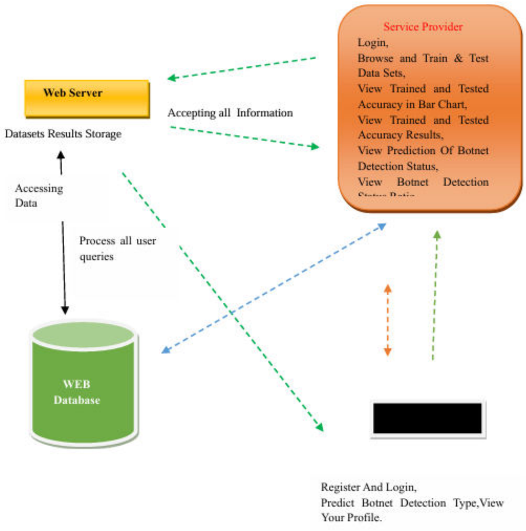
- Ensure each module—data input, ML prediction, UI interaction, and alert generation—works seamlessly.

Step 10: Deployment and Future Enhancement

- Deploy the system on local or cloud servers.
- Plan enhancements like:
    - **Federated learning**
    - **Encrypted traffic handling**
    - **Real-time device isolation**

SYSTEM ARCHITECTURE

The system architecture consists of multiple interconnected components that work together to analyze, prevent, and detect botnet attacks in IoT networks.


Service Provider
Login,
Browse and Train & Test Data Sets,
View Trained and Tested Accuracy in Bar Chart,
View Trained and Tested Accuracy Results,
View Prediction Of Botnet Detection Status,
View Botnet Detection Status Ratio
Web Server
Accepting all Information
Datasets Results Storage
Accessing Data
Process all user queries
WEB Database
[redacted]
Register And Login,
Predict Botnet Detection Type,View Your Profile.

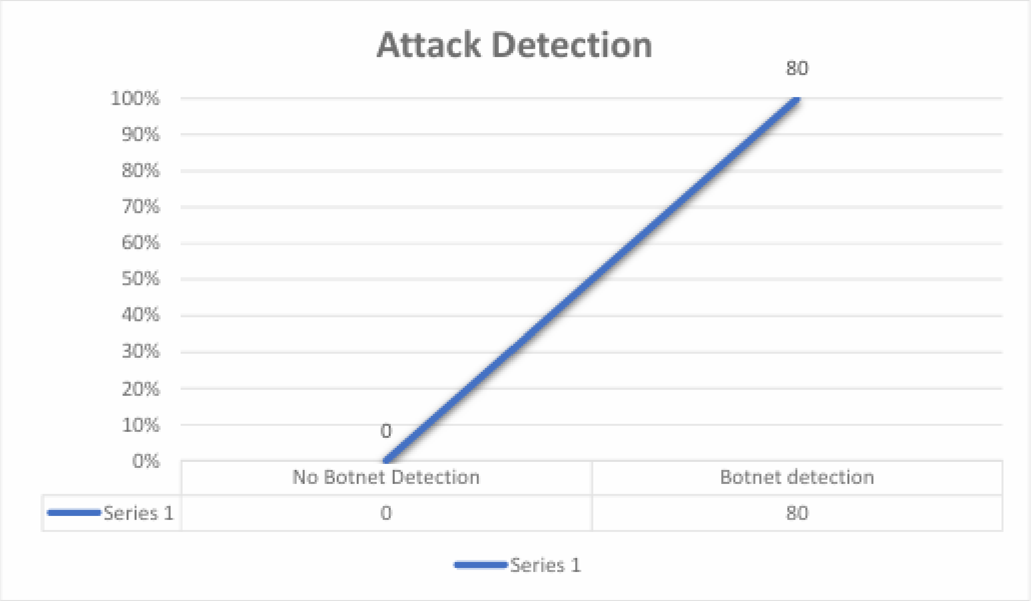
VI. RESULT ANALYSIS

Durati on	PB S	PB R	TB S	PACKE TS	SRP R	PREDCTI ON
12	0	0	60	1	2	Botnet DDOS Detection
20	0	0	88	2	0.5	No Botnet DDOS Detection
67	0	0	120	2	0	Botnet DDOS Detection
54	55	100	0	0	5	Botnet DDOS Detection

Prediction of Attacks

Detect IOT Botnet Attacks Found Ratio Details

Attack Type	Ratio
No Botnet DDOS Detection	20.0
Botnet DDOS Detection	80.0



Attack Detection

VII. CONCLUSION

In this work, we proposed a two-fold machine learning approach to prevent and detect IOT botnet attacks. In the first fold, we trained a state-of-the-art deep learning model, i.e., ResNet-18 for scanning attack detection, and named it ResNetScan-1 model. While in the second fold, we trained another ResNet-18 model (named as ResNetDDoS-1 model) in order to detect the DDOS attack in case if the scanning detection model fails to prevent a botnet attack. In order to authenticate the performance of the proposed ResNetScan-1 model and ResNetDDoS-1 model, we performed a couple of experiments in which we take the scan and DDOS traffic samples from three publicly-available datasets, trained the ResNet-18 model over these datasets, and saved the resultant Res Net Scan and Res Net DDOS models. We then tested each resultant Res Net Scan and Res Net DDOS model over the test set of other datasets on which they were not trained. The experimental results revealed that the performance of all Res Net Scan and Res Net DDOS models except the proposed ResNetScan-1 and ResNetDDoS-1 model crucially reduced when tested over the datasets on which they were not trained. Furthermore, the experimental results proved that the proposed ResNetScan-1 and ResNetDDoS-1 models persisted in their performance and outperformed all other models for detecting the scan and DDOS attacks. Hence, the proposed two-fold approach is efficient and robust to prevent and detect IOT botnet attacks with a large attack patterns coverage.

The current work only covers 33 types of scanning and 60 types of DDOS attacks. In future, we aim to cover more scanning and DDOS attacks techniques in order to well train the proposed framework for more efficient prevention and detection of IOT botnet and DDOS attacks. Further, we can deploy the proposed two-fold approach in an IDS to investigate its effectiveness on live network traffic.

FUTURE SCOPE

To further enhance the Two-Fold Machine Learning Approach for IoT botnet detection, several future directions are proposed. One key area is the development of standardized IoT datasets that reflect real-world traffic patterns, device diversity, and varying attack types. This would enable more effective model training and evaluation.

Another important aspect is the integration of Explainable Artificial Intelligence (XAI) techniques, such as LIME or SHAP, to provide insights into the model's decision-making process. This would foster trust and accountability in automated security systems. Additionally, federated learning can be adopted to preserve data privacy and reduce centralized processing, enabling IoT devices to collaboratively train models without transmitting sensitive raw data.

Future systems should also incorporate blockchain technology for device authentication and secure communication, as well as real-time automated threat mitigation mechanisms, such as isolating compromised devices or blocking malicious IPs. Optimizing models for resource-constrained devices using techniques like pruning or quantization would ensure deployment feasibility without compromising detection accuracy.

Moreover, the system's robustness against adversarial machine learning attacks should be improved through defensive strategies like adversarial training. Continuous learning and online model updating would enable the system to adapt to evolving threat patterns, reducing reliance on static datasets. Incorporating multi-modal features, such as CPU usage and system logs, would provide richer context-aware anomaly detection. Finally, achieving cross-platform and protocol interoperability would ensure broader applicability in heterogeneous IoT environments.

REFERENCES

[1] Ali, A. I. A. Ahmed, A. Almogren, M. A. Raza, S. A. Shah, A. Khan, andA. Gani, ``Systematic literature review on IoT-based botnet attack,'' IEEE Access, vol. 8, pp. 212220_212232, 2020.

[2] S. Ghazanfar, F. Hussain, A. U. Rehman, U. U. Fayyaz, F. Shahzad,and G. A. Shah, ``IoT-Flock: An open-source framework for IoT traf_cgeneration,'' in Proc. Int. Conf. Emerg. Trends Smart Technol. (ICETST),Mar. 2020, pp. 1_6.

[3] M. Safaei Pour, A. Mangino, K. Friday, M. Rathbun, E. Bou-Harb, F. Iqbal,S.Samtani, J. Crichigno, and N. Ghani, ``On data-driven curation, learning,and analysis for inferring evolving Internet-of-Things (IoT) botnets in thewild,'' Comput. Secur., vol. 91, Apr. 2020, Art. no. 101707.

[4] F. Hussain, S. G. Abbas, M. Husnain, U. U. Fayyaz, F. Shahzad, andG. A. Shah, ``IoT DoS and DDoS attack detection using ResNet,'' in Proc.IEEE 23rd Int. Multitopic Conf. (INMIC), Nov. 2020, pp. 1_6.

[5] S. Dange and M. Chatterjee, ``IoT botnet: The largest threat to the IoTnetwork,'' in Data Communication and Networks. Singapore: Springer,2020, pp. 137_157.

[6] F. Hussain, S. G. Abbas, U. U. Fayyaz, G. A. Shah, A. Toqeer, and A. Ali,``Towards a universal features set for IoT botnet attacks detection,'' in Proc.IEEE 23rd Int. Multitopic Conf. (INMIC), Nov. 2020, pp. 1_6.