AI Framework for Identifying Anomalous Network Traffic in Mirai And Bashlite IOT Botnet Attack

[1]Dr. M. Shalima Sulthana
Associate Profesor, Dept. Computer Science and Engineering
Vignan's Institute of Management and Technology for Women, Hyd.
Email: m.s.sulthana2012@gmail.com

[2]B. Bhavani
UG Student, Dept. Computer Science and Engineering
Vignan's Institute of Management and Technology for Women, Hyd.
Email: bhavanibandari342@gmail.com

[3]T. Mamatha
UG Student, Dept. Computer Science and Engineering
Vignan's Institute of Management and Technology for Women, Hyd.
Email: tannirumamatha7@gmail.com

[4]V. Chandana
UG Student, Dept. Computer Science and Engineering
Vignan's Institute of Management and Technology for Women, Hyd.
Email: chandanavaldasu@gmail.com

Abstract— This document describes the Artificial Intelligence (AI) framework for accurately detecting anomalous network traffic, with a focus on the well-known and complex IoT botnet attacks Mirai and Bashlite. The widespread use of Internet of Things (IoT) devices has not only ushered in an era of unprecedented convenience and connected ness, but it has also opened up a large attack surface to malicious actors. Examples of IoT botnets that are a major threat to critical infrastructure, business networks, and individual users are Mirai and Bashlite. These botnets use compromised devices to perform destructive tasks like massive Distributed Denial of Service (DDoS) attacks and data exfiltration. It is challenging for conventional security measures to keep up with these botnets' dynamic and polymorphic nature, which allows them to swiftly alter their attack patterns and exploit flaws in different IoT ecosystems. Rule-based intrusion detection systems (IDS) are frequently outmanoeuvred by novel attack vectors, leading to high false positive rates or—more crucially—undiscovered compromises. This study proposes an AI-driven approach that circumvents these limitations by identifying deviations from standard operational patterns using network traffic behavior rather than static signatures.

The primary objective of this framework is to develop a dependable, adaptable, and efficient system that can instantly detect unusual traffic brought on by Mirai and Bashlite. This involves extracting intricate features that point to malicious activity from vast streams of network data using state-of-the-art machine learning. The framework aims to identify both the overt attack phase and its more subtle precursors, including scanning, infection attempts, and Command and Control (C2) communications, to enable proactive mitigation strategies.

keywords—IOT Botnet Detection, Mirai Botnet, Bashlite Botnet, Random Forest, Bernoulli Naive Bayes Classifier, Performance Metrics, Internet of Things (IoT) Devices

I. INTRODUCTION

In the current digital world, the proliferation of Internet of Things devices has transformed numerous industries, leading to an unprecedented increase in network traffic and complexity. According to the Global IoT Security Market Report, the quantity of IoT devices increased by approximately 8.4 billion in 2017 and 30.9 billion by 2025. Both the likelihood and the danger of sophisticated network attacks have increased due to this exponential rise. Among the most well-known examples are the Mirai and BASHLITE botnets. First identified in 2016, the Mirai botnet exploited flaws in IoT devices to launch massive Distributed Denial of Service attacks. Since its initial discovery in 2014, the BASHLITE botnet has been connected to several cyberattacks. Traditional anomaly detection techniques that mainly rely on manual inspection and rule-based systems are ineffective in this dynamic threat landscape. These conventional approaches are less effective due to the vast volume of data, the constantly evolving networks, and the constantly improving tactics of the attackers. Manual approaches are time-consuming, prone to errors, and unable to keep up with the sophistication of contemporary cyber threats. Although rule-based systems can be useful, they are unable to adjust to identify novel or undiscovered attack patterns. In order to overcome these issues, we propose an AI-based framework for identifying anomalous network traffic associated with IoT botnet activity, emphasizing patterns associated with BASHLITE and Mirai. Because machine learning algorithms can adapt to new patterns by learning from historical data, they are a good choice. By analyzing large amounts of network data in real time, machine learning models are able to identify subtle deviations from typical behavior. This improves detection accuracy and expedites reaction times. This approach increases the efficiency and scalability of network security solutions by reducing the need for manual intervention and making it simpler to identify emerging threats. An ai framework could leverage machine learning models and looks at network data quickly, spots strange behavior, and learns to handle new attacks.

II. LITERATURE REVIEW

Meidan et al. [1] introduced the N-BaIoT dataset, comprising network traffic data from IoT devices infected with Mirai and Bashlite. They utilized deep autoencoders to model normal device behavior, enabling the detection of deviations indicative of botnet activity. Haq and Khan [2] developed DNNBoT, a deep neural network-based model designed to detect and classify IoT botnet attacks, including Mirai and Bashlite. Their approach incorporated Principal Component Analysis for feature extraction and achieved high accuracy in real-time detection scenarios. Nguyen et al. [3] proposed DÏoT, a federated self-learning anomaly detection system tailored for IoT environments. By constructing normal communication profiles for each device type, DÏoT effectively identified anomalies without relying on labeled data, demonstrating its efficacy against Mirai infections Koroniotis and Moustafa [4] introduced the Particle Deep Framework, combining particle swarm optimization with deep learning to enhance network forensic capabilities. Their model achieved a 99.9% accuracy rate in detecting botnet-related anomalies within IoT networks. In a study by Hezam et al. [5], a hybrid BiLSTM-CNN model was employed to detect botnet attacks in IoT networks. Utilizing the N-BaIoT dataset, their model demonstrated superior performance compared to traditional deep learning models, achieving high precision and recall rates. Kumar and Lim [6] presented EDIMA, an early detection system for IoT malware network activity. Leveraging machine learning techniques, EDIMA focused on identifying malicious behavior during the scanning and infection phases of botnet attacks, providing timely alerts for mitigation. In a comprehensive evaluation, researchers [7] compared various machine learning and deep learning models for IoT botnet detection. Their findings highlighted the effectiveness of CNN models in accurately classifying complex attack patterns associated with Mirai and Bashlite. A study by researchers [8] proposed a lightweight deep learning method for cross-device IoT botnet detection. By transforming network traffic into fixed-length records, their approach facilitated efficient detection across diverse IoT devices, maintaining high accuracy rates. In another investigation, researchers [9] explored the application of unsupervised deep learning techniques, such as autoencoders, for IoT botnet anomaly detection. Their approach effectively identified deviations in network behavior, contributing to the development of robust intrusion detection systems. Finally, researchers [10] developed a hybrid deep learning model combining CNN and LSTM architectures to detect botnet attacks in IoT environments. Their system demonstrated optimal performance in identifying malicious activities associated with Mirai and Bashlite infections.

III. METHODOLOGY

AI Framework for Identifying Anomalous Network Traffic in Mirai and Bashlite IoT Botnet Attacks:

1. Imports of necessary libraries are made for machine learning (sklearn), data handling (numpy, pandas), visualization (matplotlib, seaborn), and model persistence (joblib).

2.Data Loading and Exploration: A CSV file is used to load the dataset. The first and last few rows are shown, summary statistics are examined, and the 'Attack' column is checked for unique values as part of basic exploratory data analysis (EDA).

3.Preprocessing the data: o Missing values are found.
To see the relationships between features, a heatmap is created. Label Encoder is used to encode the "Attack" labels, while categorical columns such as "Device-Name" and "Attack-subType" are removed.

4.Data Visualization: A count plot shows how the dataset's various attack categories are distributed.

5.Feature and Target Variable Separation: The target variable (y), which stands for attack classes, is isolated from the feature set (x).

6.Data Splitting: To evaluate the model, the dataset is divided into training and testing sets in a 70/30 ratio.

7.Measures of Performance Function: To compute and print accuracy, precision, recall, F1 score, and a confusion matrix, a function called performance-metrics is defined.

8.Model Training: Bernoulli Naive Bayes Classifier: Predictions are made on the test set after the model has been trained on the training data. Rather than retraining, a saved model is loaded if one is available. Random Forest Classifier: The reasoning is similar. The model can be loaded from a saved state or trained from scratch.

9. Model Evaluation: The performance metrics function that was previously established is used to assess the predictions made by both classifiers.

10.Prediction on New Data: A fresh test dataset is loaded and subjected to comparable preprocessing. The Random Forest model is used to make predictions, and the results are printed, identifying the type of attack for each entry.

A. SYSTEM ARCHITECTURE

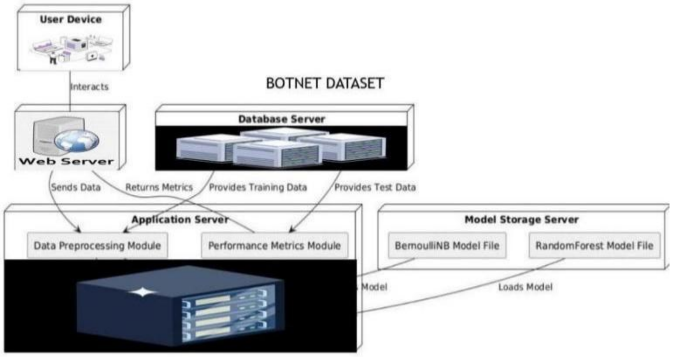


Fig 1: System Architecture

The suggested system is intended to operate as a reliable, real-time pipeline that automates number plate identification, speed estimation, and vehicle detection. It also makes sure that

all outcomes are precisely recorded and shown via a MySQL database interface. The architecture can be divided into a number of interrelated phases

1.The user's device: This is an end-user's device that communicates with the system, such as a computer, laptop, or mobile device. The "Interacts" arrow shows that requests or data exchanges with the Web Server are started by users or other systems.

2.Web Server: Serves as the main conduit for data reception and user interaction. "Sends Data": It sends information to the Application Server for processing after receiving it from the User Device or other sources, most likely network traffic logs or other pertinent data. "Returns Metrics": Following processing, it provides the user or requesting system with performance metrics or detection results that were obtained from the Application Server.

3.Database Server (BOTNET) Dataset: This is an essential part that is at the heart of the machine learning process. It serves as a database for information about botnets. "Provides Training Data": A sizable collection of network traffic data, including examples of both typical (benign) traffic and known botnet traffic (from Mirai, Bashlite, etc.), is stored on the Database Server. The Application Server receives this data in order to train the machine learning models. "Provides Test Data": In order to assess how well the trained models perform and make sure they can correctly detect botnet activity on unseen data it additionally offers a distinct subset of data.

4.Server for Applications: In charge of handling the machine learning models and processing data, this is the system's computational core.

It has several important modules: The Data Preprocessing Module is responsible for converting raw data from the Web Server or Database Server into a format that machine learning algorithms can use. Cleaning data, dealing with missing values, feature extraction (such as identifying packet size, connection time, and protocol types from network traffic), and normalization may all be part of this. Evaluation of the trained machine learning models' performance is the responsibility of the Performance Metrics Module. The Web Server receives the metrics it computes, which include accuracy, precision, recall, F1-score, and false positive/negative rates. "Sends Data": Gets data from the Web server, either raw or preprocessed. In order to load the pre-trained machine learning models (such as Random Forest) for inference (i.e., generating predictions on fresh data), it communicates with the Model Storage Server.

5.Server for Model Storage: The trained machine learning models are stored on this server. Two categories of stored models are clearly displayed:

A file that contains a trained Bernoulli Naive Bayes model is called a Bernoulli NB Model File

types and having a high accuracy rate. Loads Model": When botnet detection is required, the Application Server pulls these trained models from the Model Storage Server.

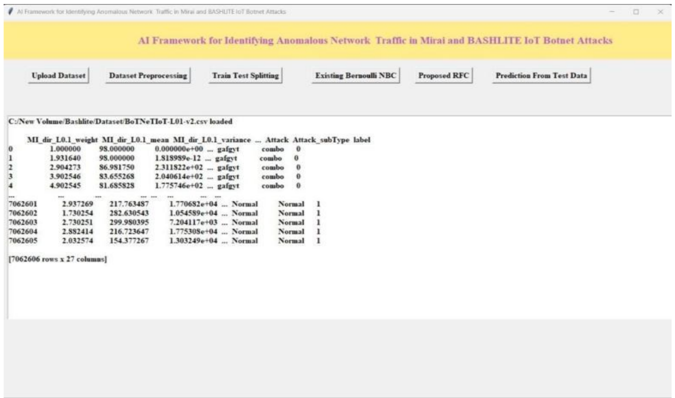
B. IMPLEMENTATION

A hybrid approach is used by the current intrusion detection system (IDS) to find questionable network activity. To identify threats linked to well-known malware families like Mirai and Bashlite, it mainly uses a signature-based intrusion detection system (IDS) to identify known attack patterns. Nevertheless, this element has trouble with new or changing attack types. The system also includes an anomaly-based intrusion detection system (IDS) that makes use of less complex statistical techniques. It makes use of a Bernoulli Naive Bayes Classifier that works with binary features. The Bernoulli model's intrinsic information loss and Naive Bayes' independence assumption limit this method's efficacy in identifying sophisticated and complex botnet activity, even though it provides a baseline for identifying departures from typical network behavior. This suggested system targets IoT botnet attacks such as Mirai and BASHLITE and provides a strong methodology for detecting unusual network traffic. Careful data preprocessing is the first step in the process, which involves cleaning and standardizing the raw internet traffic from different devices. The dataset's attack labels are subsequently numerically encoded to make machine learning algorithms easier to use. The most important characteristics or "clues" from the processed traffic data are efficiently identified and chosen using a "heatmap" visualization technique, guaranteeing that the analysis that follows concentrates on extremely pertinent signs of malicious activity. The fine-tuned dataset is then subjected to a critical division after feature selection: 70% is set aside for machine learning model training, and the remaining 30% is set aside for independent testing. Common performance indicators, such as precision, accuracy, and Recall, F1 score, and the confusion matrix are all precisely defined to offer a thorough framework for assessing the models' efficacy. Machine learning models form the basis of the detection mechanism, and the Random Forest algorithm is a top contender. Using the large training dataset, this model is trained to identify patterns suggestive of botnet attacks. Then, using the testing data that is not visible, its performance is thoroughly assessed. The Random Forest model is ultimately chosen for its capacity to predict attack types in fresh, incoming network traffic, thereby successfully detecting and classifying IoT botnet attacks like Mirai and BASHLITE with high confidence. It was chosen due to its well-established dependability and powerful predictive capabilities.

IV. EXPERIMENTAL RESULT AND ANALYSIS

AI Framework for Identifying Anomalous Network Traffic in Mirai and BASHLITE IoT Botnet Attacks
Upload Dataset
Dataset Preprocessing
Train Test Splitting
Existing Bernoulli NBC
Proposed RFC
Prediction From Test Data

Fig 2: AI Framework


AI Framework for Identifying Anomalous Network Traffic in Mirai and BASHLITE IoT Botnet Attacks
AI Framework for Identifying Anomalous Network Traffic in Mirai and BASHLITE IoT Botnet Attacks
Upload Dataset
Dataset Preprocessing
Train Test Splitting
Existing Bernoulli NBC
Proposed RFC
Prediction From Test Data

```
C:/New Volume/Bashlite/Dataset/BoTNeTIoT-L01-v2.csv loaded

     MI_dir_L0.1_weight  MI_dir_L0.1_mean  MI_dir_L0.1_variance  ...  Attack  Attack_subType  label
0            1.000000         98.000000          0.000000e+00  ...  gafgyt       combo      0
1            1.931640         98.000000          1.818989e-12  ...  gafgyt       combo      0
2            2.904273         86.981750          2.311822e+02  ...  gafgyt       combo      0
3            3.902546         83.655268          2.040614e+02  ...  gafgyt       combo      0
4            4.902545         81.685828          1.775746e+02  ...  gafgyt       combo      0
...               ...               ...                   ...  ...     ...         ...    ...
7062601      2.937269        217.763487          1.770682e+04  ...  Normal      Normal      1
7062602      1.730254        282.630543          1.054589e+04  ...  Normal      Normal      1
7062603      2.730251        299.980395          7.204117e+03  ...  Normal      Normal      1
7062604      2.882414        216.723647          1.775308e+04  ...  Normal      Normal      1
7062605      2.032574        154.377267          1.303249e+04  ...  Normal      Normal      1

[7062606 rows x 27 columns]
```

Fig 3: Upload Dataset

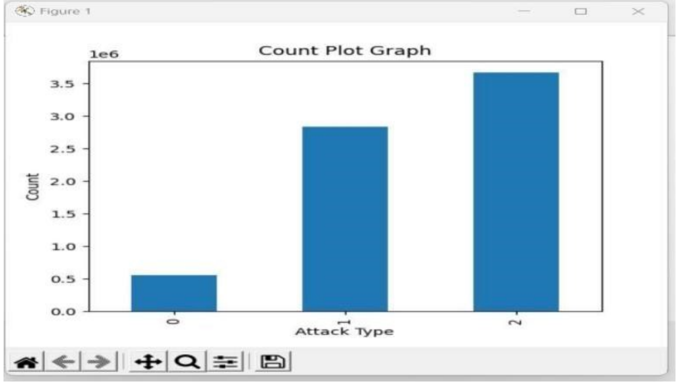
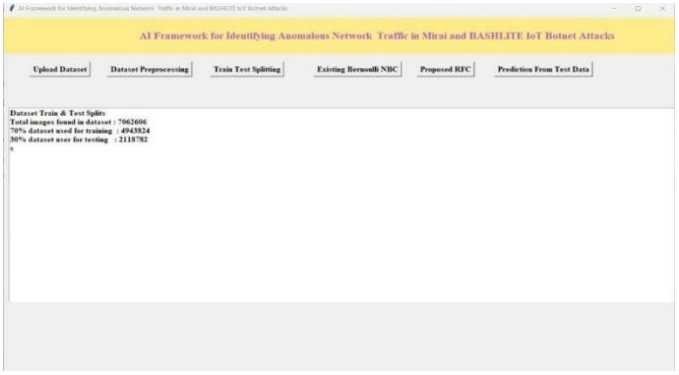
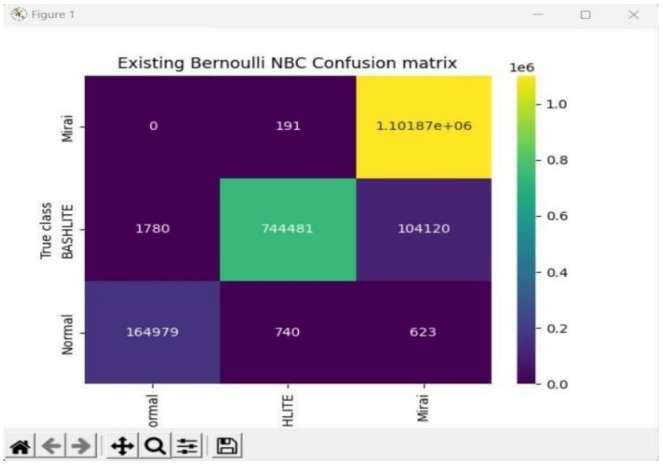


Fig 4: Count Plot Graph


AI Framework for Identifying Anomalous Network Traffic in Mirai and BASHLITE IoT Botnet Attacks
AI Framework for Identifying Anomalous Network Traffic in Mirai and BASHLITE IoT Botnet Attacks
Upload Dataset
Dataset Preprocessing
Train Test Splitting
Existing Bernoulli NBC
Proposed RFC
Prediction From Test Data

```
Dataset Train & Test Splits
Total images found in dataset : 7062606
70% dataset used for training  : 4943824
30% dataset user for testing   : 2118782
s
```

Fig 5: Train Test Splitting Dataset



	ormal	HLITE	Mirai
Mirai	0	191	1.10187e+06
BASHLITE	1780	744481	104120
Normal	164979	740	623

Fig 6: Existing Bernouli NBC Confusion Matrix

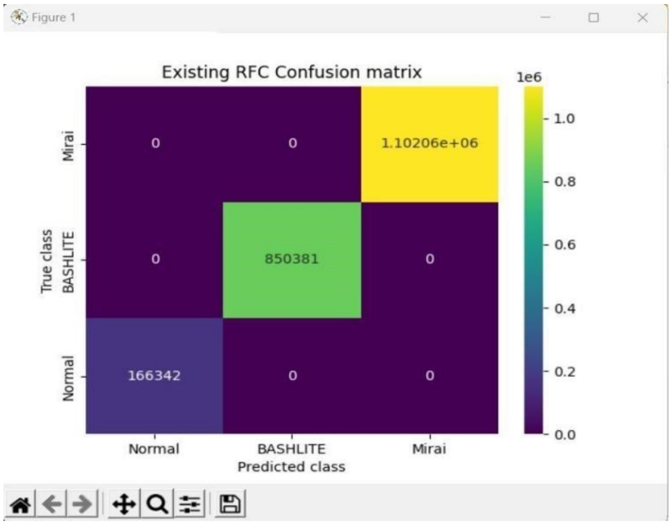
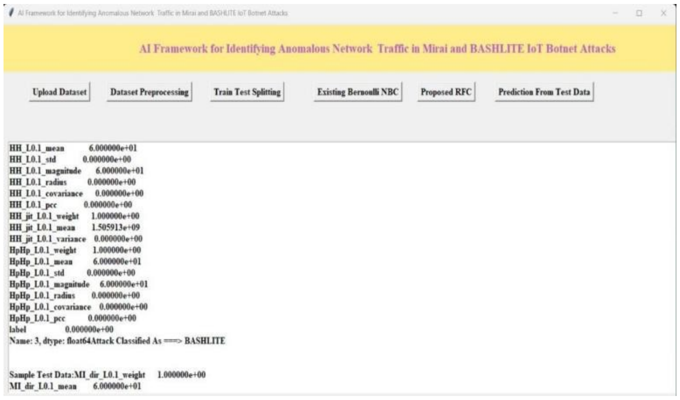


Fig 7: Existing RFC Confusion Matrix


AI Framework for Identifying Anomalous Network Traffic in Mirai and BASHLITE IoT Botnet Attacks
AI Framework for Identifying Anomalous Network Traffic in Mirai and BASHLITE IoT Botnet Attacks
Upload Dataset
Dataset Preprocessing
Train Test Splitting
Existing Bernoulli NBC
Proposed RFC
Prediction From Test Data

```
HH_L0.1_mean          6.000000e+01
HH_L0.1_std         0.000000e+00
HH_L0.1_magnitude      6.000000e+01
HH_L0.1_radius        0.000000e+00
HH_L0.1_covariance     0.000000e+00
HH_L0.1_pcc         0.000000e+00
HH_jit_L0.1_weight     1.000000e+00
HH_jit_L0.1_mean       1.505913e+09
HH_jit_L0.1_variance    0.000000e+00
HpHp_L0.1_weight       1.000000e+00
HpHp_L0.1_mean        6.000000e+01
HpHp_L0.1_std        0.000000e+00
HpHp_L0.1_magnitude     6.000000e+01
HpHp_L0.1_radius       0.000000e+00
HpHp_L0.1_covariance    0.000000e+00
HpHp_L0.1_pcc        0.000000e+00
label             0.000000e+00
Name: 3, dtype: float64Attack Classified As ===> BASHLITE


Sample Test Data:MI_dir_L0.1_weight     1.000000e+00
MI_dir_L0.1_mean      6.000000e+01
```

Fig 8: Prediction from Dataset

V.CONCLUSION

This study does a good job of showing how machine learning techniques can be used to enhance the detection of intricate Internet of Things-based botnet attacks. The framework addresses the challenges caused by the increasing volume and complexity of data by utilizing the Random Forest. Mirai and BASHLITE, two well-known botnets that have a history of orchestrating massive Distributed Denial of Service (DDoS) attacks and other cyberthreats by taking advantage of vulnerabilities in Internet of Things devices, are the targets of the implementation. Machine learning enables the system to recognize anomalous patterns in network traffic that traditional rule-based systems might overlook. Specifically, the Random Forest classifier is a wise selection due to it offers a high degree of precision and robustness.

VI. FUTURE SCOPE

Even if the existing system is reliable, it can yet be expanded and improved. The following improvements could be investigated in further iterations

Integration with RealTime Systems: One of the primary future directions for this project is to integrate the trained machine learning models into real-time network monitoring systems. This would involve deploying the models within network security infrastructures to continuously monitor and analyze incoming traffic for anomalies, providing instant alerts and automated responses to potential threats. Real-time deployment would also require optimizing the models for speed and efficiency, ensuring minimal latency in detection. more scalable and infrastructure expenses can be decreased by optimizing the model to run on edge devices such as the Raspberry Pi or Jetson Nano.

Incorporating Advanced Machine Learning Techniques: While Bernoulli Naive Bayes and Random Forest classifiers provide a solid foundation, future enhancements could explore more advanced machine learning techniques, such as deep learning. Convolutional Neural Networks (CNNs) or Recurrent Neural Networks (RNNs) could be employed to capture more complex patterns in network traffic data, potentially improving detection rates for sophisticated or novel attack vectors.

Expanding the Scope to Include Other Botnets: The current framework focuses on detecting Mirai and BASHLITE botnets. Future work could expand the scope to include other emerging IoT botnets, such as Hajime, Amnesia, or Reaper. By incorporating a broader range of attack types, the framework could be made more versatile and capable of handling a wider array of threats.

VII. REFERENCES

[1] Meidan, Y., et al. "N-BaIoT: Network-based Detection of IoT Botnet Attacks Using Deep Autoencoders." arXiv preprint arXiv:1805.03409 (2018).

[2] Haq, M. A., & Khan, M. A. R. "DNNBoT: Deep Neural Network-Based Botnet Detection and Classification." Computers, Materials & Continua 71.1 (2022):45394.

[3] Nguyen, T. D., et al. "DÏoT: A Federated Self learning Anomaly Detection System for IoT." arXiv preprint arXiv:1804.07474 (2018).

[4] Koroniotis, N., &Moustafa, N. "Enhancing Network Forensics with Particle Swarm and Deep Learning: The Particle Deep Framework." arXiv preprint arXiv:2005.00722 (2020).

[5] Hezam, M. A., et al. "Combining Deep Learning Models for Enhancing the Detection of Botnet Attacks in Multiple Sensors Internet of Things Networks." JOIV: International Journal on Informatics Visualization 6.2 (2022):733.

[6] Kumar, A., & Lim, T. J. "EDIMA: Early Detection of IoT Malware Network Activity Using Machine Learning Techniques." arXiv preprint arXiv:1906.09715 (2019).

[7] Researchers. "Intelligent Detection of IoT Botnets Using Machine Learning and Deep Learning." Applied Sciences 10.19 (2020):7009.

[8] Researchers. "A Deep Learning Method for Lightweight and Cross-Device IoT Botnet Detection." Applied Sciences 13.2 (2023):837.

[9] Researchers. "IoT Botnet Anomaly Detection Using Unsupervised Deep Learning." Electronics 10.16 (2021): 1876.

[10] Researchers. "Hybrid Deep-Learning Model to Detect Botnet Attacks over Internet of Things Environments." Soft Computing (2022).