Proceeding of 2024 International Conference on Computing and Intelligent Reality Technologies (ICCIRT)
IEEE Xplore Part Number: CFP24OZ6-ART; ISBN: 979-8-3315-1029-9

IOT Botnet Detection using a Hybrid of CNN-LSTM with Blockchain

Aditya Ranganthan Arun
Computer Science and Engineering
National Institute of Technology Puducherry
Karaikal, India
adityaa2302@gmail.com

Annisha R. de Souza
Computer Science and Engineering
National Institute of Technology Puducherry
Karaikal, India
racheldesouza520@gmail.com

Sairam S.
Computer Science and Engineering
National Institute of Technology Puducherry
Karaikal, India
s.r.02.09.2004@gmail.com

4thVaniV
Computer Science and Engineering
National Institute of Technology Puducherry
Karaikal, India
v.vani465@gmail.com

Karthik N
Computer Science and Engineering
National Institute of Technology Puducherry
Karaikal, India
nkarthikapce@gmail.com

2024 International Conference on Computing and Intelligent Reality Technologies (ICCIRT) | 979-8-3315-1029-9/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICCIRT59484.2024.10921767

***Abstract*—The exponential growth of IoT devices has intensified the threat of botnet attacks, creating an urgent need for secure and adaptive detection solutions. Convolutional Neural Networks (CNN) are employed to extract spatial features, while Long Short-Term Memory (LSTM) models record temporal dependencies in network traffic. A blockchain ledger continuously logs all the model parameters and performance matrices by hashing them with SHA-256 thus making the data verifiable. The blockchain layout provides a decentralized, unaltered record of model performance ensuring transparency and reliability in the botnet detection systems. This study integrates the CNN-LSTM model with blockchain technology to enhance the effectiveness and reliability of the botnet detection system in IoT environments. The model is trained in the IoT-23 datasets using SMOTE for data balancing and reducing false positive rates. This study finds that the implementation of CNN-LSTM model without sampling to be the most effective giving an accuracy of 0.999.**

Keywords—Botnet, Blockchain, Hybrid Model, SMOTE

I. INTRODUCTION

The Internet of Things (IoT) ecosystem has expanded rapidly in recent years by connecting billions of devices across diverse applications in various sectors like healthcare, daily life, etc. This increased connectivity has also exposed the IoT network to various cyber threats, botnet attacks which can exploit vulnerabilities in unsecure devices. In the IoT environments, botnets can execute attacks that are sophisticated such as Distributed Denial of Service (DDoS), often escaping detection due to their complex and evolving persona [1]. Traditional detection approaches which rely on rule-based systems, handcrafted features struggle to chase the adaptive speed of modern botnets.

To tackle these challenges, this study proposes a blockchain-integrated deep learning model for IoT botnet detection. It can be a combination of Convolutional Neural Network (CNN) for spatial feature extraction whereas Long Short-Term Memory (LSTM) is used to capture temporal dependencies in the network traffic to detect botnet patterns more effectively and efficiently. Additionally, the integration of blockchain technology provides a secure and immutable framework for logging model parameters and performance metrics [2,3]. This decentralized approach ensures data integrity, facilitating trustworthy and reliable records of detection results that are tamper resistant. The proposed system leverages the IoT-23 dataset to evaluate model performance, using Synthetic Minority Over-sampling Technique (SMOTE) to manage data anomalies and improve detection performance [4]. This blockchain-integrated CNN-LSTM model aims at enhancing the IoT environment security by proving a transparent, resilient, and adaptive solution to counter the evolving botnet threats in the fast-growing technical space.

II. RELATED WORKS

This section includes the pre-existing solutions to IoT botnet detection. The need for IoT botnet detection arises due to the malleable nature of IoT devices leading them to be the perfect conduit for contagion. As manufacturers try to reduce the cost of producing IoT devices namely by running outdated firmware and software. They also tend to not follow basic safety protocols such as two-factor authentication, thereby making them incredibly vulnerable.

The rising prevalence of IoT networks has introduced increasing security vulnerabilities, particularly the threat of botnet attacks that exploit weakly protected systems. Traditional detection techniques like the rule-based systems often fail to adapt to the evolving nature of botnet attacks. Therefore, researchers have turned to advanced methods, including machine learning (ML), and blockchain, to enhance detection capabilities with data integrity [2,5,6]. Machine Learning (ML) and Deep Learning (DL) models are essential for identifying IoT-based botnet attacks due to their capability to detect complex and subtle patterns in network

traffic. In [1], ML models were used to ascertain botnet activity by analyzing network traffic, emphasizing the significance of effective feature selection in enhancing detection accuracy. [6] proposed the SMOTE-DRNN approach, which uses Synthetic Minority Oversampling Technique (SMOTE) and deep recurrent neural networks to handle data imbalance issues found in IoT botnet datasets.Hybrid models are effective in capturing spatial as well as temporal patterns, making them ideal for complex sequence data like network traffic. [1] demonstrated that CNN-LSTM architectures can effectively capture hierarchical patterns from the network traffic for more accurate botnet detection compared to CNN or LSTM models applied separately.

Similarly, used bidirectional LSTM models to analyze IoT traffic and achieve high detection accuracy because of the model's ability to retain long-term dependencies [3].

Blockchain technology has gained much attention for improving the security and integrity of IoT systems by providing an immutable, decentralized ledger for records. [1] introduced the "BlockchainBot", a botnet model using blockchain for enhancing resilience against forensic investigations and attack disruptions. Their model leveraged IoT devices in conjunction with blockchain to establish a robust, tamper-resistant botnet infrastructure that mitigated vulnerabilities associated with the traditional P2P-based botnets. Supporting, [3] emphasized the role of blockchain's decentralized nature in IoT, noting its ability to avoid any single point of failure, which is crucial for securing sensitive data and providing transparent, verifiable log records.

The paper explores the application of deep learning methods for botnet detection in IoT environments. McDermott et al. investigate various deep learning techniques, including convolutional neural networks (CNNs) and recurrent neural networks (RNNs), to improve detection accuracy [7].

This research builds on these advancements by integrating CNN-LSTM hybrid model with blockchain to enhance the detection and security of botnet activities in IoT environments. Combining these technologies allows for high detection accuracy and a secure framework for logging and validating performance data. Thus, addressing the multifaceted challenges of IoT botnet detection.

III. PROPOSED METHODOLOGY

The proposed methodology demonstrates a hybrid method for IoT botnet detection, integrating machine learning with blockchain technology and the same have been described in the following sections.

A. Data Preparation

A log file containing the dataset is read. The dataset contains various pertinent data related to network traffic data such that missing values are filled with zeros. Labels are converted into binary labels in order to distinguish between benign and malicious traffic.

Focusing primarily on network characteristics, for instance, duration, packet counts, and bytes transferred, is used to select features. The 'label' column is converted into binary labels to distinguish between benign and malicious traffic (1 for malicious, 0 for benign). The features of the log file are illustrated in Table I.

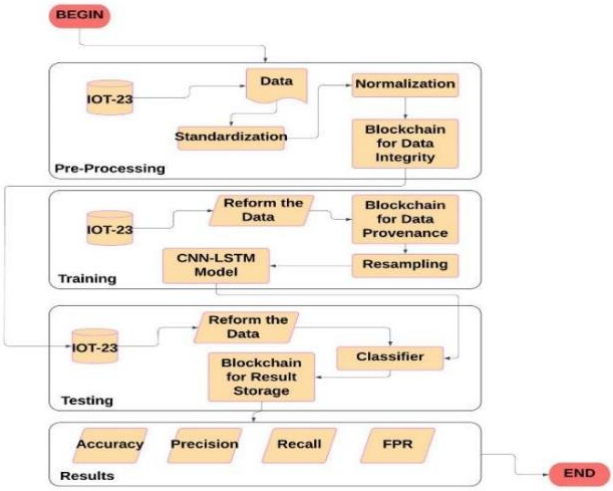

Fig. 1. Architecture of the Proposed System

TABLE I. FILE FEATURES

CATEGORY	Number of Features
Duration	1
Packet Bytes	2 (orig_bytes, resp_bytes)
Packet Count	2 (orig_pkts, resp_pkts)
Protocols	1 (proto, with categories converted to multiple binary features)
Service Type	1 (service, with categories converted to multiple binary features)

B. Data Normalization and Reshaping

The implementation of a hybrid model leverages StandardScaler to standardize feature values. Data is then reshaped to fit the input requirements of the hybrid model, particularly the shape. The shape contains attributes such as samples, time steps, and features. Due to the absence of temporal dependencies, the time steps are set to 1.

C. Model Building and Training

The model's architecture consists of a hybrid model combining CNN and LSTM, which is illustrated in Figure 1 below. Convolutional Neural Network (CNN) is primarily used to implement tasks related to image or visual data analysis. It is beneficial in recording extracting meaningful features. It consists of multiple steps of which the main three are: convolution, pooling, and fully connected layers. These layers are created so that they incrementally learn spatial hierarchies through a backpropagation algorithm. Long Short-Term Memory (LSTM) networks are a form of

repetitive neural network that integrates memory units in their hidden layers, allowing them to selectively retain long-term patterns. This feature makes LSTMs especially effective for modeling sequential data.

D. Blockchain Integration

A simple blockchain is implemented to log model parameters and results securely. The trained model's parameters are then hashed, and a transaction containing this hash and model performance metrics is added to the blockchain. After which, a new block is mined to secure the model's performance data as well as making it tamper-proof. Therefore, integrating a blockchain provides a permanent and transparent record of the model's performance.

E. Model Hashing

The parameters of the hybrid model are then flattened and hashed with the help of SHA-256. This guarantees that model performance can be reproduced and validated against the stored hash in the blockchain, thus providing a trust mechanism for the model's integrity.

IV. EXPERIMENTAL SETUP

This section contains a comprehensive view of the key components that allow us to prepare and test the data on our system of choice. Data preparation was done using the IoT-23 dataset. using a series of performance metrics that were used to compare and evaluate the values. Table II contains a snapshot of malicious IoT scenarios.

A. Dataset

The IoT-23 dataset, launched by the Stratosphere Laboratory at the Czech Technical University, Prague, is extensively used in IoT security research [7]. It offers a comprehensive set of network traffic data to address IoT security issues, particularly botnet detection. The dataset encompasses a variety of botnet activities observed in traffic from different IoT devices. These activities include various types of infections, and malicious behaviors of compromised devices. Notable activity labels are as follows:

- **Heartbeat**: C&C servers monitor infected hosts through heartbeat packets.
- **Mirai and Okiru**: Connections exhibit behavior specific to Mirai and Okiru botnets.
- **Attack**: Infected devices exploit vulnerabilities in other hosts.
- **Benign**: Connections show no malicious indicators.
- **C&C**: Connections to a Command & Control server.
- **DDoS**: Distributed Denial of Service attacks through compromised botnet devices.
- **File Download**: File downloads by infected devices.
- **PartOfAHorizontalPortScan**: Connections used for horizontal port scanning, marked by patterns like consistent port use, similar byte transmission, and multiple destination IPs [8].

TABLE II. SUMMARY OF THE MALICIOUS IOT SCENARIOS

#	Name of Dataset	Duration (hrs)	#Packets	#ZeekFlows	Pcap Size	Name
1	CTU-IoT-Malware-Capture-34-1	24	233,000	23,146	121 MB	Mirai
2	CTU-IoT-Malware-Capture-43-1	1	82,000,000	67,321,810	6 GB	Mirai
3	CTU-IoT-Malware-Capture-44-1	2	1,309,000	238	1.7 GB	Mirai
4	CTU-IoT-Malware-Capture-49-1	8	18,000,000	5,410,562	1.3 GB	Mirai
5	CTU-IoT-Malware-Capture-52-1	24	64,000,000	19,781,379	4.6 GB	Mirai
6	CTU-IoT-Malware-Capture-20-1	24	50,000	3,210	3.9 MB	Torii
7	CTU-IoT-Malware-Capture-21-1	24	50,000	3,287	3.9 MB	Torii
8	CTU-IoT-Malware-Capture-42-1	8	24,000	4,427	2.8 MB	Trojan
9	CTU-IoT-Malware-Capture-60-1	24	271,000,000	3,581,029	21 GB	Gagfyt
10	CTU-IoT-Malware-Capture-17-1	24	109,000,000	54,659,864	7.8 GB	Kenjiro
11	CTU-IoT-Malware-Capture-36-1	24	13,000,000	13,645,107	992 MB	Okiru
12	CTU-IoT-Malware-Capture-33-1	24	54,000,000	54,454,592	3.9 GB	Kenjiro
13	CTU-IoT-Malware-Capture-8-1	24	23,000	10,404	2.1 MB	Hakai
14	CTU-IoT-Malware-Capture-35-1	24	46,000,000	10,447,796	3.6G	Mirai
15	CTU-IoT-Malware-Capture-48-1	24	13,000,000	3,394,347	1.2G	Mirai
16	CTU-IoT-Malware-Capture-39-1	7	73,000,000	73,568,982	5.3GB	IRCBot
17	CTU-IoT-Malware-Capture-7-1	24	11,000,000	11,454,723	897 MB	Linux,Mirai
18	CTU-IoT-Malware-Capture-9-1	24	6,437,000	6,378,294	472 MB	Linux.Hajime
19	CTU-IoT-Malware-Capture-3-1	36	496,000	156,104	56 MB	Muhstik
20	CTU-IoT-Malware-Capture-1-1	112	1,686,000	1,008,749	140 MB	Hide and Seek

B. Performance Metrics

For evaluation purposes, we use a confusion matrix. It contains four elements:

True Positive (TP): The data which correctly identifies the attack's characteristics, meaning it accurately detected and classified the attack type.

True Negative (TN): The data accurately identified normal traffic as non-threatening.

False Positive (FP): The data classified normal traffic incorrectly as an attack.

False Negative (FN): The data fails to detect an attack, classifying it instead as normal traffic [9].

The hybrid model is evaluated using the following metrics:

1) Accuracy: Accuracy represents the proportion of correct classifications, covering both positive and negative cases. The equation for accuracy is shown in the equation below (1).

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

2) Precision: Precision is the proportion of all the model's positive classifications that are positive. The equation for Precision is mentioned below in equation (2).

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

3) Recall: Recall is the ratio of original positive cases that were identified as positive correctly by the classifier. Also referred to as the True Positive Rate (TPR), the formula of Recall is provided in the equation below (3).

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

4) F1-score: The F1-score is formulated as the harmonic mean of Recall and Precision. The formula for the F1-score is provided in the equation below (4).

$$F1 - score = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (4)$$

5) False Positive Rate (FPR): The False Positive Rate is the ratio of actual negative cases that were classified as positive.. The equation for False Positive Rate is mentioned below in equation (5).

$$False\ Positive\ Rate = \frac{FP}{FP + TN} \quad (5)$$

C. System Specifications

The model has been implemented in Python. The systems hardware and software are listed in Table III.

TABLE III. SYSTEM SPECIFICATIONS

HW/SW	Settings
Clock speed	1.70 GHz
Processor	12Th Gen Intel® Core™ i7-1255U
RAM	16 GB
GPU	NVIDIA GeForce MX550
Python	3.10.0
Operating System	Windows 11
Sci-Kit Learn	0.24.1
TensorFlow	2.17.0

V. VI. RESULTS

The proposed CNN-LSTM-blockchain integration model was tested with the IoT-23 datasets to assess its effectiveness and performance in detecting botnet activity.
The CNN-LSTM model's performance was evaluated using three sampling strategies: without sampling, random under sampling, and SMOTE for addressing the class imbalance in the IoT-23 dataset.

The disproportionate distribution of data between classes is called class imbalance which can be addressed by randomly resampling the training dataset by under sampling or oversampling. Undersampling involves removing instances from the majority class, whereas oversampling multiplies the instances from the minority class. This enables the model to focus more on minority patterns but risks losing important information, potentially lowering overall accuracy (0.9626 in this study) due to a narrower dataset [2,6].

Without sampling refers to the non-probability sampling which uses no random criteria to sample the data. This often results in high overall accuracy (0.9999 in this study) as the model primarily learns patterns of the majority class. However, relying on accuracy alone can be misleading, as it may not reflect effective detection of the minority class [5, 9].

SMOTE on the other hand balances the data by multiplying Sthe number of instances in the minority class, generating new samples. Here, SMOTE improved recall to 0.998 and F1-score to 0.971, indicating better detection of malicious activity. It's important to acknowledge that SMOTE can sometimes introduce noise, especially close to the decision boundary of the minority class, potentially resulting in a higher rate of false positives for imbalanced datasets like IoT-23 [1, 2].
SMOTE enhances recall by creating synthetic examples for underrepresented classes, but it can lead to more false positives by adding noisy instances. This could result in "alert fatigue," where security teams and systems are overwhelmed with excessive alerts, possibly slowing down the response to real threats. Achieving a balance between improving detection and maintaining system efficiency requires careful threshold tuning, combined sampling methods, and post-detection validation to ensure effective threat detection without overburdening the system.
Key metrics were used for evaluating the performance: The Blockchain ledger effectively logged model parameters and metrics, ensuring data integrity and verifiability. Additionally, this approach facilitates seamless auditing of historical model performance. The integration of Blockchain also enhances trust in the model's deployment process by providing a tamper-proof record. Overall, the model performed with high accuracy and low false positive rate, thus making it a promising tool for real-time detection.

A. CNN

The IoT-23 dataset was processed with the stand-alone CNN model with all these sampled data and the results are illustrated below in Table IV.

TABLE IV. RESULTS USING CNN

	Accuracy	Precision	Recall	F1-Score	FPR
Without Sampling	0.9994	0.9994	0.999	0.999	0.83
Random Under Sampling	0.5931	0.5496	0.999	0.709	0.80

SMOTE	0.5889	0.5487	0.999	0.708	0.82

B. CNN-LSTM

The IoT-23 dataset was processed with the integrated CNN-LSTM model with all these sampled data and the results are illustrated below in table V.

TABLE V. RESULTS USING CNN-LSTM

	Accuracy	**Precision**	**Recall**	**F1-Score**	**FPR**
Without Sampling	0.9999	0.9999	0.999	0.999	0.06
Random Under Sampling	0.9626	0.9380	0.990	0.963	0.06
SMOTE	0.9711	0.945	0.998	0.971	0.05

Figure 2 shows a side-by-side comparison of each metric of both the models to depict a clearer understanding of the results.

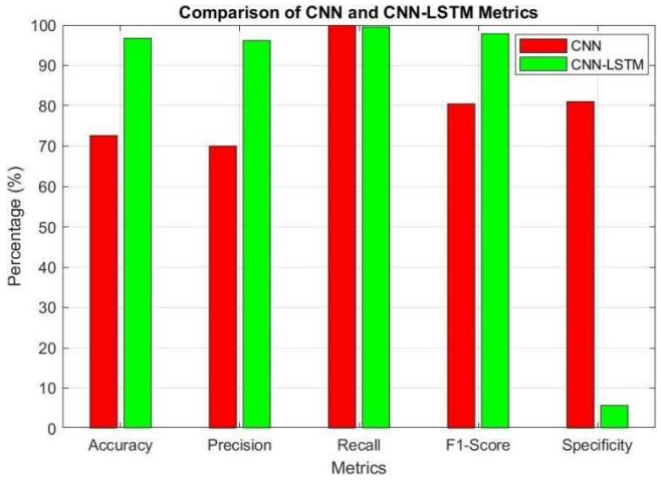

Fig. 2. Comparison of CNN and CNN-LSTM metrics

VII. CONCLUSION

This study presents an innovative approach to IoT botnet detection by integrating CNN-LSTM hybrid model with blockchain technology that addresses both the accuracy of detection and the security of model performance data. By leveraging CNN layers for extracting spatial features while LSTM layers capture temporal dependencies. This model achieves a high classification capability for distinguishing between benign and malicious network traffic. This integration proved to be effective on testing with IoT-23 by offering high accuracy in botnet detection while mitigating issues related to imbalanced data.

Additionally, integrating with blockchain introduces a secure, reliable method for logging model parameters and performance metrics. Using blockchain to store hashed model parameters provides a decentralized ledger that ensures data integrity and makes the results reproducible and verifiable. This approach also provides protection against tampering of records.

Overall, the hybrid CNN-LSTM model with blockchain integration provides a viable solution that protects IoT environments against evolving botnet threats. Future works on this model may explore the scope of optimizing the blockchain's scalability to accommodate larger datasets and integrating other machine learning architectures to further improve detection performance. This research contributes a secure and transparent framework that combines deep learning with blockchain thus pushing the boundaries of IoT botnet detection techniques.

REFERENCES

[1] Popoola, S.I.; Adebisi, B.; Ande, R.; Hammoudeh, M.; Anoh, K.; Atayero, A.A. SMOTE-DRNN: A Deep Learning Algorithm for Botnet Detection in the Internet-of-Things Networks. Sensors 2021, 21, 2985. https://doi.org/10.3390/s21092985I.

[2] Zheng, Zibin & Xie, Shaoan & Dai, Hong-Ning & Chen, Xiangping & Wang, Huaimin. (2017). An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends. 10.1109/BigDataCongress.2017.85.

[3] Habib, G.; Sharma, S.; Ibrahim, S.; Ahmad, I.; Qureshi, S.; Ishfaq, M. Blockchain Technology: Benefits, Challenges, Applications, and Integration of Blockchain Technology with Cloud Computing. Future Internet 2022, 14, 341. https://doi.org/10.3390/fi14110341

[4] Anwar, Shahid & Mohamad Zain, Jasni & Zolkipli, Mohamad & Inayat, Zakira. (2014). A Review Paper on Botnet and Botnet Detection Techniques in Cloud Computing.

[5] Gao, H.; Li, L.; Chang, X.; Wan, J.; Li, J.; Du, J.; Zhang, X. BlockchainBot: A Novel Botnet Infrastructure Enhanced by Blockchain Technology and IoT. *Electronics* **2022**, *11*, 1065. https://doi.org/10.3390/electronics11071065

[6] Sebastian Garcia, Agustin Parmisano, & Maria Jose Erquiaga. (2020). IoT-23: A labeled dataset with malicious and benign IoT network traffic (Version 1.0.0) [Data set]. Zenodo. http://doi.org/10.5281/zenodo.4743746

[7] McDermott, Christopher & Majdani, Farzan & Petrovski, Andrei. (2018). Botnet Detection in the Internet of Things using Deep Learning Approaches. 10.1109/IJCNN.2018.8489489.

[8] Van Houdt, Greg & Mosquera, Carlos & Nápoles, Gonzalo. (2020). A Review on the Long Short-Term Memory Model. Artificial Intelligence Review. 53. 10.1007/s10462-020-09838-1.

[9] Djenna, A.; Barka, E.; Benchikh, A.; Khadir, K. Unmasking Cybercrime with Artificial-Intelligence-Driven Cybersecurity Analytics. Sensors 2023, 23, 6302. https://doi.org/10.3390/s23146302