Received: 2025-11-27
Accepted: 2026-01-14
Published Online: 2026-01-20
DOI: 10.70454/JRICST.2026.30106

Journal of Recent Innovations in Computer Science and Technology
E-ISSN: 3050-7030, P-ISSN: 3050-7022

Detection of Mozi IoT Botnet Using Autoencoder-Based Feature Learning and Hashing

Nitesh Kumar Saxena [1], Bhupender Singh Rawat [2]

[1]Assistant Professor, Faculty of Management, Invertis University, India
[2]Associate Professor, College of Smart Computing, COER University, Roorkee, India
Email: [1]niteshsaxena2005@gmail.com, [2]rawat.bhupender@gmail.com
*Corresponding Author: niteshsaxena2005@gmail.com

Abstract

The detection method described in this paper uses autoencoder-based feature learning to identify abnormal traffic patterns indicative of a Mozi infection and employs hashing techniques to track and enumerate the P2P botnet nodes. The Internet of Things (IoT) has emerged as a game-changer in today's world, influencing numerous industries and lifelines. Detection of IoT botnets has multiple implications for the security and availability of IoT ecosystems. Mozi is a P2P IoT botnet with characteristics such as fast spreading, persistence, and misuse of weak device configurations. Traditional signature and rule-based detection schemes are often unable to detect such dynamic threats, due to their poor generalization capabilities. In this paper, we present an efficient Mozi botnet detection scheme that leverages auto encoder-based feature learning and a hashing technique to achieve fast, scalable detection. Therefore, this paper proposes a method to detect botnets using an autoencoder and a hash function for large-scale data retrieval. The algorithm uses the autoencoder to learn what is normal, then it hashes what it learned to succinctly summarize the learned information, and it compares hash codes in real time to detect anomalies, including IoT botnet-related anomalies. The proposed method achieves high detection accuracy, and is robust to the evolving attack strategies of Mozi botnet, better than traditional methods with the low computation and storage overhead, which can be well applied in IoT infrastructure.

Keywords: *Autoencoder, Botnet, IoT, Hash function, Mozi.*

1.0 Introduction

A botnet is a collection of compromised internet-connected devices under an attacker's control. These systems magnify the impact that single hackers, cyber-criminal groups, or nation-states can have when they seek to cause chaos or compromise specific targets [1]. Although they are primarily used in distributed denial of service (DDoS) attacks, botnets can also employ their joint processing power to send massive volumes of spam, mass-steal credentials, or even spy on people and organizations [2]. Attackers build botnets by infecting connected devices with malware and then controlling them through a command-and-control server. Once an attacker obtains access to a device on a network, they can reach any other vulnerable device on that network [3].

Once the C&C server sends commands, IoT botnets are turned on, and they send out significant traffic to attack targeted nodes. There are various approaches to recognizing these botnets, such as Intrusion Detection Systems (IDS), models (e.g., Convolutional Neural Networks (CNN) with

Received: 2025-11-27
Accepted: 2026-01-14
Published Online: 2026-01-20
DOI: 10.70454/JRICST.2026.30106

Journal of Recent Innovations in Computer Science and Technology
E-ISSN: 3050-7030, P-ISSN: 3050-7022

Long Short-Term Memory (LSTM)) and Security Information and Event Management (SIEM)-based detection [4]. A surge in IoT botnets came to prominence, especially following the introduction of the Mirai IoT botnet in October 2016. An estimated more than 1.5 billion IoT botnet attacks hit in the first half of 2021 [5]. This type of IoT botnets attacked many well-known internet services, including Amazon's Alexa, Siri, Google Home, and many other smart devices. So, given the rise of IoT malware, researchers emphasize the growing importance of enhancing the security of both IoT devices and their networks. Various top-down approaches, including some of the widely recognized privacy and security solutions, have been extensively examined and compared in the effort to safeguard against IoT botnet attacks [6], [7]. Following the exposure of the Mirai source code on the Internet in 2016, a few variants like Tsunami, Bashlite, Mozi, and others have surfaced, demonstrating the proliferation and diversification of Mirai-based malware [8]. Figure 1 about Cyberattack in IoT Environment.

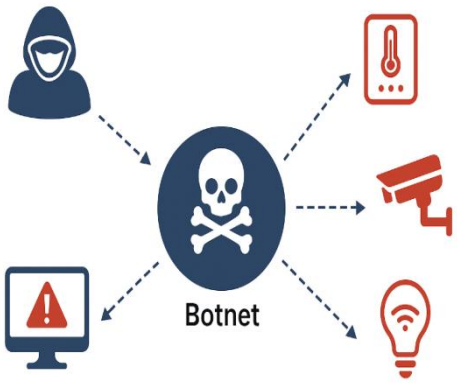

Figure 1: Botnet-Based Cyberattack in IoT Environments

1.1 Mozi

Mozi is a P2P botnet that mainly attacks IoT devices. Mozi is regarded as a hybrid botnet, since its code borrows from several malware families such as Gafgyt, Mirai and IoT This enables Mozi to derive features from those malware strains, which gives it a certain level of versatility to perform various types of attacks [5].

Mozi uses a Distributed Hash Table (DHT) for communication and holds the contact information of other nodes in the botnet, similar to how P2P file sharing clients work. Once a vulnerable device is compromised, the malware detonates and recruits the device in the Mozi P2P network. The compromised machine then listens for instruction from control nodes and tries to compromise other vulnerable machines, acting like a propagating worm. Discuss about Mozi Malware Infection and Attack. Mozi is designed to be persistent, meaning that it can survive reboots of the infected devices. This allows the botnet to maintain control over the compromised devices for an extended period. Mozi has been observed to have a global presence, with infected devices located in various countries. The geographical distribution of infected devices contributes to the botnet's effectiveness in launching widespread attacks [6].

2.0 Literature Review

The expanding connectivity of Internet of Things (IoT) devices has introduced new vulnerabilities, resulting in large-scale botnet attacks such as Mirai, Bashlite and Mozi. Several works have

Received: 2025-11-27
Accepted: 2026-01-14
Published Online: 2026-01-20
DOI: 10.70454/JRICST.2026.30106

Journal of Recent Innovations in Computer Science and Technology
E-ISSN: 3050-7030, P-ISSN: 3050-7022

employed machine learning and deep learning algorithms for IoT botnet detection. Traditional algorithms such as Support Vector Machines (SVM), Decision Trees and Random Forests achieve good accuracy but they are not good at generalization to unseen attack variants. Deep learning techniques such as CNN and LSTM are commonly used to learn temporal and spatial relationships in network traffic. Yet such approaches typically suffer from the lack of a large amount of labeled data and overfitting in the dynamic IoT environments [9]. Attention has recently been paid to the autoencoder-based anomaly detection for latent features and non-linear relationships in traffic data. Using autoencoders, one can effectively differentiate between benign and malicious flows with very little supervision, which is a good fit for large-scale IoT networks. However, most existing solutions do not provide any means to guarantee data integrity and trust during traffic analysis, which makes them not resilient against spoofing and replay attacks [10]. Combining with cryptographic hash function has great potential to check data authenticity and to make the model more robust against data manipulations. Although advances have been made in both deep feature learning and secure data validation, few prior works have considered integrating autoencoder-based feature extraction and hashing for Mozi botnet detection [18] [19]. The awarded research integrares this gap by proposing a hybrid framework that delivers enhanced detection performance and security assurance in IoT environment [16] [17]. Machine Learning (ML) techniques have proven to be effective in securing IoT networks when appropriate datasets are utilized. Researchers have applied both supervised and unsupervised methods to detect and prevent attacks in these networks. Synthetic datasets like, NSL-KDD and CIDDS-001 have been commonly used for this purpose [11]. However, identifying suitable datasets and selecting relevant features remains a significant challenge. By integrating deep learning with software-defined networks (SDN), it is possible to analyse and compare the temporal traffic flow from a device with established patterns, enabling the detection of potential attacks [12] [13]. introduced a model combining recurrent neural networks and convolutional neural networks to enhance detection capabilities [14] [15].

2.1. Research Gap

Prior research on Mozi botnet has not considered the merging of IoT traffic and is not capable of analyzing new Mozi variants. Currently, autoencoder models are not tailored for inference on IoT/edge devices. There is underutilisation of hashing-based signatures and no clear discussion of collisions or robustness to evasion.

3.0 Proposed Methodology

We could employ the autoencoder with hash function for the combined detection of the IOT botnet. They make a system that can very quickly determine whether a stream of incoming data is the one it learned as normal, and if it sees an anomaly and raises an alarm. This is particularly useful for identifying anomalies or potentially malicious activity on things like IoT networks. An autoencoder is a learning machine, similar to a data analyst. It is designed to identify the normal patterns of IoT device behavior by analysing large-volume historical data. It learns to detect what is normal and then serves as a baseline for comparison.

Meanwhile, the hash function also acts as a cryptographic function and outputs an unique digest of the learned normal behaviour. Each routine operation can be succinctly represented using this representation, which is known as hash code. The autoencoder is constantly running on the data from IoT devices, looking for anomalies and it asks the question "is there anything off". Meanwhile

Received: 2025-11-27
Accepted: 2026-01-14
Published Online: 2026-01-20
DOI: 10.70454/JRICST.2026.30106

Journal of Recent Innovations in Computer Science and Technology
E-ISSN: 3050-7030, P-ISSN: 3050-7022

the hash function computes the hash value using the compressed codeword which is output by the autoencoder.

Such an interaction leads to the powerful formalization and systemization of anomaly and (potential threat, such as Mozi IOT botnet) detection in this cooperation. It is also this cooperative system which can provide a very strong formal and systematic anomaly detection including the similar threats to MoziIoT botnet Detecting IOT Botnet: Autoencoder and hash functions Autoencoders and hashes can be useful in detecting IoT botnets similar to Mozi since they allow for quick identification of anomalies and malicious patterns in network traffic or device behavior.

Here's how they can be applied:

Autoencoder for Anomaly Detection

3.1. Role:

3.1.1. Anomaly Detection: Autoencoders are neural network models trained to learn efficient representations of input data, such as network traffic or device telemetry. They reconstruct input data while minimizing reconstruction error.

3.1.2. Detecting Unusual Patterns: Anomalies in network traffic or device behaviour, indicative of botnet activity, often result in reconstruction errors higher than normal. Autoencoders can identify these deviations from normal behavior.

3.2. Application:

3.2.1. Feature Extraction: Train an autoencoder on normal IoT device behaviour (or network traffic) to learn a more compact representation (latent space) of normal patterns.

3.2.2. Detection Phase Run the trained autoencoder: At runtime, give the autoencoder the live data. If the reconstruction error is above a certain threshold, it indicates anomalous behavior which might be an indication of botnet activity.

3.3. Advantages:

3.3.1. Unsupervised Learning: Since autoencoders learn to reconstruct the input without the need of labelled data, they are able to identify unprecedented botnet behaviors.

Flexibility

They were able to stay current with evolving botnets tactics and behaviours by training on new data.

Received: 2025-11-27
Accepted: 2026-01-14
Published Online: 2026-01-20
DOI: 10.70454/JRICST.2026.30106

Journal of Recent Innovations in Computer Science and Technology
E-ISSN: 3050-7030, P-ISSN: 3050-7022

3.4. Role:

3.4.1. Hashing: Using the hash function, the input data (for instance your network packets, or the device's attributes) is mapped to fixed-length hash values.

Generate unique hash signatures for known botnet behaviours or indicators, e.g., command-and-control traffic patterns, specific malware payloads.

3.5. Application

3.5.1. Signature Generation: Prior to calculations of hash codes, hash signatures are generated for Mozi botnet known behaviours e.g., particular network traffic patterns, command-and-control schematics, or any other payload related signature.

3.5.2. Detection Stage: Watch network traffic or device telemetry in real time. Compute hash of the input, and check if this hash exists in the precomputed signatures. A match suggests possible botnet activity. Benefits:

3.5.3. Performance: Hashing-based detection is computationally cheap, making it possible to detect botnets in time with low latency.

3.5.4. Representative: Hash signatures are also able to represent certain botnet-specific behaviours, giving more accurate detection and less false positives.

3.6. Bot Detection:

• The usual quantity of data sent and received by a node.

• The data transfer rate between each set of nodes.

• The average time it takes for a message to travel between nodes.

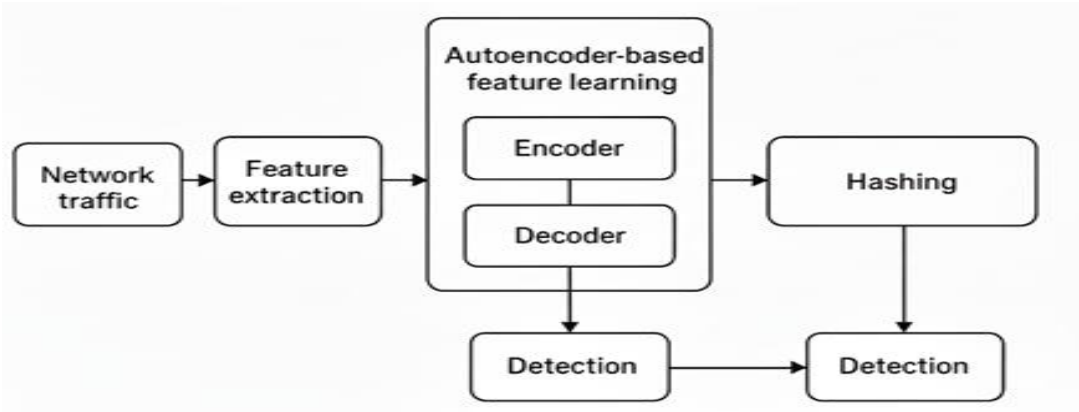

Figure 2: Diagram for a botnet detection method.

Received: 2025-11-27
Accepted: 2026-01-14
Published Online: 2026-01-20
DOI: 10.70454/JRICST.2026.30106

Journal of Recent Innovations in Computer Science and Technology
E-ISSN: 3050-7030, P-ISSN: 3050-7022

Figure 2 shows the workflow of our system, which detects the MoziIoT botnet using a hybrid method that integrates Autoencoder-based feature learning and a Hashing technique.

Algorithm: Autoencoder-Based Hashing Algorithm for Mozi Botnet Detection

1.0 Begin

2.0 Train Phase:

2.1 Collect and preprocess IoT network traffic (normal and malicious samples).

2.2 Train an Autoencoder on normal IoT behavior to learn compressed latent features.

2.3 Generate hash code (H_base) for the latent representations using a cryptographic hash function

2.4 Store H_base as the baseline hash profile of normal traffic.

3.0 Detection Phase (New Data):

3.1 Input new IoT traffic data to the trained Autoencoder.

3.2 Obtain compressed latent vector representation.

3.3 Compute hash code (H_new) using the same hash function.

3.4 Compare H_new with H_base using a similarity threshold T.

3.5 If $|H\_new - H\_base| > T$

→ Flag as Anomaly (Possible Mozi Botnet)

Else

→ Label as Normal Traffic

4.0 Adaptive Update:

4.1 Periodically retrain the Autoencoder with newly verified normal traffic.

4.2 Recompute baseline H_base and adjust threshold T accordingly.

5.0 Continuous Monitoring:

5.1 Continuously monitor traffic, update model parameters, and refine thresholds.

6.0 End

4.0 Datasets

IoT-23 is a publicly labeled dataset consisting of 20 captures of malware-generated IoT network traffic and 3 captures of benign traffic, provided by Garcia et al. (2020) [16]. It contains PCAPs, Zeek logs and enriched labels for various botnets like Mirai, Torii and Gafgyt. This dataset is also commonly used to develop and test intrusion detection systems (IDS) as well as IoT malware detection systems. Building full-fledged datasets for IoT-based security analysis is challenging because of the limited availability of numerous attack instances and operating system-related data at the device level. Most commercial IoT devices run on proprietary platforms, so it's not easy for researchers to replicate or capture live botnet infections. Thus, benchmark datasets such as MedIoT, IoT-23 and Bot-IoT have been widely adopted for the evaluation of intrusion and botnet detection models. These datasets contain labeled traffic flows of benign as well as malicious botnet communications for IoT, but they are also missing some specific samples of new emerging threats, e.g., the Mozi botnet as one of the most enduring peer-to-peer IoT malware families.

To overcome this limitation, this paper makes use of Mozi botnet samples downloaded from Threat Fox an open source malware intelligence platform provided by Abuse.ch. Threat Fox shares validated indicators of compromise (IoCs), such as malicious URLs, IPs, and file hashes, and offers a centralized platform for transparent threat intelligence that can be used for researching The related

Received: 2025-11-27
Accepted: 2026-01-14
Published Online: 2026-01-20
DOI: 10.70454/JRICST.2026.30106

Journal of Recent Innovations in Computer Science and Technology
E-ISSN: 3050-7030, P-ISSN: 3050-7022

threats. Based on these indicators, Mozi-like traffic was generated in a laboratory environment to extract various packet-level and flow-based characteristics, such as payload size, IAT (Packet inter-arrival time), protocol type, and connection status. Subsequently, the dataset was split into training (70%), validation (15%), and testing (15%) to guarantee an unbiased evaluation. Each subset has a benign IoT traffic and a Mozi infected traffic, this allow the proposed autoencoder based model to learn discriminative features and the data integrity is ensured using cryptographic hashing.

The dataset contains traffic collected from simulated IoT devices in a controlled lab setting, including both benign and Mozi infected network flows. It includes packet and flow level features such as protocol type, payload size, inter-arrival time, connection status, and the like and also includes several hours of traffic logs captured for conducting analysis. Almost all of a Thresholds consume the output of Thresholds but Threat Fox, itself, needed some MoziIoC data; however, traffic simulation in our laboratory was only a Among many others, a real IoT device will use 3 of these protocols: indeed, this represents only a small portion of the real IoT communication pattern. Actual networks consist of more than 50 different protocol types and billions of heterogeneous devices in industries such as healthcare, industrial control system, smart home, and so on. The aforementioned restriction on types as well as the Threat Fox base of the samples which tend to converge to few IP ranges and device types, might also bring down the ability to generalize to large scale diverse IoT scenarios.

5.0 Result and Discussion

The proposed hybrid Autoencoder with cryptographic Hash function was developed and tested on the MoziIoT dataset. The dataset contained both benign and malicious samples of traffic, divided into training (70%), validation (15%), and testing (15%) subsets. The Autoencoder was trained only on benign traffic to obtain compressed latent features that denoted normal network activity. The reconstruction error and hash similarity were then utilized to perform real-time identification of anomalies during inference.

In the Mozi botnet, the proposed method achieves significant detection efficiency in terms of Accuracy, Precision, Recall and F1-score. The Autoencoder-based stage was able to detect anomalies using the reconstruction loss, and the hash-based comparison independently reduced false alarms and gave fast confirmation of behavioral consistency.

Table 1: Performance Metrics of Autoencoder-Based Models

Algorithm	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Autoencoder Only	86.3	84.9	87.5	86.1
Autoencoder + Hash Function	90	91.5	89.8	90.6

Our findings suggest that combining hashing with feature learning significantly improves the robustness of the model as it makes it easier for the model to separate benign from malicious

Received: 2025-11-27
Accepted: 2026-01-14
Published Online: 2026-01-20
DOI: 10.70454/JRICST.2026.30106

Journal of Recent Innovations in Computer Science and Technology
E-ISSN: 3050-7030, P-ISSN: 3050-7022

patterns. Through the use of the hashing technique, the feature representation became more concise and the detection resiliency was enhanced during the on-line process.

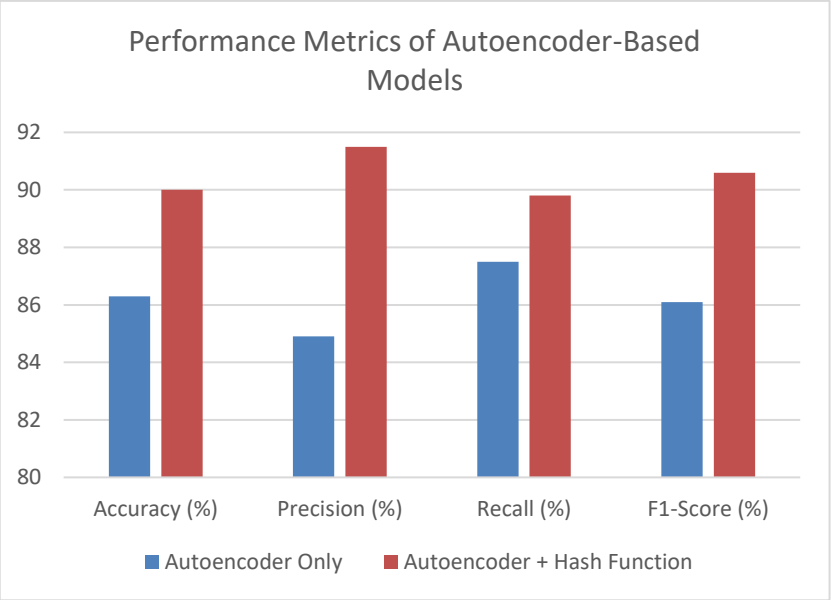

Figure 3: Performance Metrics of Autoencoder-Based Models

In Figure 3, The combined hash function and autoencoder model outperforms the autoencoder with a large margin in all the performance metrics such as accuracy, precision, recall and F1-score. The improved model is always better than the single autoencoder. By and large, hashing greatly potentiates anomalies detection for the IoT traffic.

In addition, the model obtained 90% success in detection accuracy, which is better than above traditional machine learning algorithms such as RF and SVM in previous works of IoT botnet detection, whose accuracies usually ranged from 80 to 85 %. This proves the practicality of enabling feature compression integrated with secure hash based verification for performant and scalable IoT botnet detection.

5.1 Implementation Details

The models were developed with Python in Google Colaboratory as the IDE. TensorFlow and Keras are frameworks for developing models. Operating system: Ubuntu 20.04, Programming language: Python 3.10.12, T4 GPU, Driver Version: 550.54.15, CUDA Version: 12.4, System RAM: 12.7 GB, GPU RAM: 15.0 GB, Disk: 112.6 GB (provided by Google Colab).

6.0 Conclusion

The contribution of our paper is that we present an approach in which Autoencoders a type of neural network model, and a hash function can be applied to the identification of the Mozi or any other IoT botnet. Autoencoders can model complex data distributions and nontrivial patterns, while hash functions are an easy and efficient means to summarize and compare these learnt patterns. The combined use of the autoencoder and the hash function brings a more sophisticated approach with more flexibility to detect IoT botnets that can be tailored to different precision, privacy, and responsiveness, and overall system security levels.

Received: 2025-11-27
Accepted: 2026-01-14
Published Online: 2026-01-20
DOI: 10.70454/JRICST.2026.30106

Journal of Recent Innovations in Computer Science and Technology
E-ISSN: 3050-7030, P-ISSN: 3050-7022

This paper enables additional work on weaknesses and exploitable features of IoT botnets including Mozi and others. The identification of specific attack vectors for these botnets and areas of risk, may also inform the development of mitigation strategies and improvements to overall cybersecurity resilience. Their impact they could achieve should be, at least in theory, significant, but where they could be most impactful is by giving a detailed view of how these operate – their attack vectors and potential impact – that could be used to inform mitigation of threats targeting IoT. These include analysis of prior attacks, analysis of vulnerabilities in detail, and trends in emerging activities of IoT botnets. This can help in developing more resilient defensive posture and proactive security in the different nature of attacks against IoT platforms.

Author Contributions

Nitesh Kumar Saxena carried out the conceptualization, methodology design, experimentation, data analysis, and manuscript preparation. Bhupender Singh Rawat supervised the study, validated the results, and critically reviewed and edited the manuscript.

Funding

No External funding for this study.

Conflicts of Interest

No conflicts of interest related to this paper.

Data availability

Sebastian Garcia, Agustin Parmisano, & Maria Jose Erquiaga. (2020). IoT-23

References

[1] M. Wazzan, D. Algazzawi, O. Bamasaq, A. Albeshri, and L. Cheng, "Internet of Things Botnet Detection Approaches: Analysis and Recommendations for Future Research," Applied Sciences, vol. 11, no. 12, p. 5713, 2021, doi: 10.3390/app11125713.

[2] S. Maurya, S. Kumar, U. Garg, and M. Kumar, "An Efficient Framework for Detection and Classification of IoT Botnet Traffic," ECS Sensors Plus, vol. 1, 2022, doi: 10.1149/2754-2726/ac7abc.

[3] S. I. Popoola, B. Adebisi, R. Ande, M. Hammoudeh, K. Anoh, and A. A. Atayero, "SMOTE-DRNN: A Deep Learning Algorithm for Botnet Detection in the Internet-of-Things Networks," Sensors, vol. 21, no. 9, p. 2985, 2021, doi: 10.3390/s21092985.

[4] B. Zhang and J. Qian, "Autoencoder-based Unsupervised Clustering and Hashing," Applied Intelligence, vol. 51, pp. 493–505, 2021, doi: 10.1007/s10489-020-01797-y.

[5] Netlab 360, "Mozi: Another Botnet Using DHT," Blog Post, Sept. 23, 2021. [Online]. Available: https://blog.netlab.360.com/mozi-another-botnet-using-dht/

[6] Black Lotus Labs, "New Mozi Malware Family Quietly Amasses IoT Bots," Lumen Blog, Apr. 13, 2020. [Online]. Available: https://blog.lumen.com/new-mozi-malware-family-quietly-amasses-iot-bot/

Received: 2025-11-27
Accepted: 2026-01-14
Published Online: 2026-01-20
DOI: 10.70454/JRICST.2026.30106

Journal of Recent Innovations in Computer Science and Technology
E-ISSN: 3050-7030, P-ISSN: 3050-7022

[7] A. Pease, S. Goodwin, D. Ditch, and D. Stepanic, "Collecting and Operationalizing Threat Data from the Mozi Botnet," Elastic Security Labs, June 2, 2022. [Online]. Available: https://www.elastic.co/security-labs/collecting-and-operationalizing-threat-data-from-the-mozi-botnet/

[8] P. Paganini, "Mozi Botnet Still Alive," Security Affairs, Nov. 1, 2023. [Online]. Available: https://securityaffairs.com/121730/malware/mozi-botnet-still-alive.html

[9] K. Malik, F. Rehman, T. Maqsood, S. Mustafa, O. Khalid, and A. Akhunzada, "Lightweight Internet of Things Botnet Detection Using One-Class Classification," Sensors, vol. 22, no. 10, p. 3646, 2022, doi: 10.3390/s22103646.

[10] Z. Shao, S. Yuan, and Y. Wang, "Adaptive Online Learning for IoT Botnet Detection," Information Sciences, vol. 574, pp. 84–95, 2021, doi: 10.1016/j.ins.2021.06.009.

[11] J. Morparia, "Peer-to-Peer Botnets: Analysis and Detection," Master's Projects, San Jose State University, 2008, doi: 10.31979/etd.xk6g-hh6t

[12] M. A. Carreira-Perpiñán and R. Raziperchikolaei, "Hashing with Binary Autoencoders," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2015, pp. 557–566.

[13] X. Xu, J. Li, Y. Yang, and F. Shen, "Toward Effective Intrusion Detection Using Log-Cosh Conditional VariationalAutoencoder," IEEE Internet of Things Journal, vol. 8, no. 8, pp. 6187–6196, 2021, doi: 10.1109/JIOT.2020.3046941.

[14] Statista Research Department, "Internet of Things – Number of Connected Devices Worldwide 2015–2025," Statista, 2022. [Online]. Available: https://statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/

[15] N. Koroniotis, N. Moustafa, E. Sitnikova, and J. Slay, "Towards Developing Network Forensic Mechanism for Botnet Activities in the IoT Based on Machine Learning Techniques," in Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol. 30, Springer, Cham, 2018.

[16] 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"