A SCALABLE TWO-LAYER ML FRAMEWORK FOR REAL-TIME IOT BOTNET INTRUSION DETECTION

P. K. Suryawanshi[1], S. K. Jagtap[2]

Abstract—The rapid growth of Internet of Things (IoT) networks has significantly raised the cyberattack surface, making such networks vulnerable to advanced botnet attacks. Traditional Intrusion Detection Systems (IDS) become ineffective in IoT networks owing to their rigid adaptability, high latency, and restrictive resources. To address these issues, this paper introduces a scalable two-layer machine learning framework for real-time botnet intrusion detection in IoT networks. The proposed system employs lightweight classifiers for quick screening of normal traffic in the first layer and sophisticated models for deep analysis of suspicious flows in the second layer. A robust preprocessing pipeline incorporating feature selection and class balancing strategies enhances model efficiency and detection accuracy. Experimental results demonstrate enhanced performance in detection rates, false positive reduction, and inference speed, thereby determining the model suitability for latency-restricted and resource-limited environments. The framework effectively maintains accuracy and computational cost, offering an efficient solution for modern IoT security systems

Index terms—IoT security, botnet detection, intrusion detection system, layered machine learning, SMOTE.

I. INTRODUCTION

The Internet of Things (IoT) has significantly accelerated the merging of smart environments through the interconnecting of billions of devices across various industries, including healthcare, industrial automation, and home systems. Nevertheless, the expansion mainly intensifies the attack surface, with IoT systems being extremely vulnerable to botnet-driven malicious activities. Botnets, which consist of networks of hijacked IoT devices controlled by malicious users, can compromise services via Distributed Denial of Service (DDoS) attacks, enable data breaches, and provide unauthorized access, hence resulting in tremendous security and privacy problems [1].

Traditional Intrusion Detection Systems (IDS) perform inadequately in such networks owing to their limited processing capacity, latency demands, and incapability to confront the heterogeneous and dynamic nature of flow patterns typical of Internet of Things (IoT) networks [2]. Such shortcomings render real-time botnet traffic detection especially challenging and less accurate. Innovative solutions such as federated learning [21], edge computing deployments [22], and adaptive lightweight frameworks [23] are designed to mitigate the above weaknesses.

Machine learning (ML) and deep learning (DL) can be utilized to enhance IDS performance, but existing models typically cannot optimize detection accuracy, inference speed, and resource efficiency together [3]. Class imbalance, high-dimensional features, and model complexity also limit their usage in real-world real-time, resource-limited IoT applications. These issues have been overcome in some research via using explainable AI models [24], hybrid ML frameworks [25], and interpretable detection techniques [26]. Research Gap: Even with additional research focus on ML-based intrusion detection, an efficient, real-time botnet detection system that scales well with the appropriate accuracy, latency, and computational cost remains an open link. Blockchain-based botnet detection [27], variational auto-encoders with constraints [28], and graph-partitioning-based distributed mechanisms [29] are promising but under-developed in commercial-grade systems. Graph neural network and subgraph sampling method-based recent developments also have been promising directions [30].

The research suggests an expandable two-layer machine learning system that uses high-performance classifiers to enable fast filtering of traffic as well as advanced models for a detailed analysis of potentially malicious data. This suggestion is backed by previous research on ensemble classifiers [32] and federated learning models emphasize user privacy [34].

Major contributions of this paper are:

- Suggests a layer-based ML structure that maximizes detection rate and accuracy for IoT botnet detection [25], [30].
- Blends light models (DT, KNN) for coarse screening with high-performance classifiers (RF, XGBoost) for fine-grained analysis [32], [33].
- uses hybrid feature selection and Synthetic Minority Over-sampling Technique to handle class imbalance [24], [28].
- Exhibits outstanding detection performance with comprehensive benchmarking on benchmark datasets to ensure its application in real-time, resource-constrained IoT settings [31], [33].

The rest of this manuscript is organized as follows: Section II is literature review; Section III is description of methodology and proposed tiered model; Section IV is experimental setup and results; and Section V is concluding remarks and possible future research directions.

II. RELATED WORK

Intrusion Detection Systems (IDS) need to defend IoT networks against a multitude of cyber-attacks, including botnet attacks. IDS methods nowadays increasingly depend on machine learning and deep learning to enhance detection. Real-time applicability in IoT environments is nevertheless constrained by challenges such as resource constraints, latency, and class imbalance.

A. ML and DL-Based Botnet Detection Methods

Early work explored the use of machine learning algorithms, such as Decision Trees (DT), Support Vector Machines (SVM), and K-Nearest Neighbors (KNN), which, as efficient as they are computationally, are not good generalizers to the IoT traffic's heterogeneity and time variability [3]. Recent studies have utilized deep learning models—like Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs)—in an attempt to improve detection accuracy; however, their high computational complexity makes them unsuitable to run on energy-limited IoT devices [6], [7].

For instance, Aldhyani et al. [1] introduced a deep learning and ensemble IDS with excellent detection accuracy but longer inference time. Similarly, Akhter et al. [3] compared IoT intrusion detection ML solutions and were confronted with accuracy-latency trade-offs. Ramadan et al. [7] used advanced DL models for botnet attack detection but were limited by latency due to complexity in models and real-time deployment was not feasible.

Other existing models attempt to improve performance by employing federated learning for edge-deployable detection [21], adaptive lightweight ML models for real-time processing [23], and graph-based partitioning for scalable intrusion detection [29]. These models attempt to minimize computational overhead while ensuring decent detection accuracy.

B. Limitations of Current Methodologies

Even though promising, existing approaches have certain technical limitations. First, latency remains the nemesis for DL models for real-time IoT applications. Second, IoT data property of high-dimensional data decreases the efficiency of the model. Third, majority class bias results due to class imbalance. Sarker et al. [4] identified this model complexity and real-time responsiveness issue. Islam et al. [5] proposed hybrid ML-DL models for the same, but computational expenses remain high for large-scale deployment.

Works like [30] and [31] have identified how graph neural networks and other sophisticated sampling techniques (e.g., GraphSAINT) can make the model more scalable, though they are used sparingly because of complexity and no real-time guarantees. Explainable AI models [24], [26] also offer transparency but come with overhead of interpretability.

C. Transitioning to Layered Learning Architectures

In order to resolve the identified issues, multi-layer machine learning architectures have been proposed. The architectures try to mix fast filtering techniques with more precise, resource-focused classifiers. The first layer effectively filters out benign traffic through light models like Decision Trees (DT) and K-Nearest Neighbors (KNN), whereas the next layer utilizes sophisticated classifiers (like Random Forest and XGBoost) for broader inspection [25], [26]. Hierarchical models minimize total processing needs without sacrificing high accuracy rates. Federated models [34] also facilitate distributed light detection with privacy-preserving methods in Internet of Things (IoT).

In addition, authors such as Prasath and Kavitha [20] showed that CNN-XGBoost hybrids are capable of high accuracy and efficient inference, which has inspired such layer-based architectures. Additionally, ensemble methods with flow-based feature extraction [32] and constrained variational autoencoders [28] help improve detection with fewer false positives.

TABLE I
COMPARATIVE ANALYSIS OF RELATED WORKS

Ref.	Method	Dataset	Accuracy (%)	Key Limitation
[1]	DL + Ensemble	Custom	97.5	Latency and resource overhead
[3]	ML (DT, SVM, KNN)	IoT-23	94.1	Scalability and imbalance issues
[5]	Hybrid ML-DL	NSL-KDD	96.3	High complexity, low real-time suitability
[6]	CNN, RNN	Bot-IoT	96.7	High computational overhead
[7]	Deep Neural Network	Custom IoT	97.9	Latency bottleneck and limited scalability
[21]	Federated Learning	Custom	96.2	Communication overhead in training
[23]	Lightw	Simula	95.4	Trade-off

	eight ML	ted		in complex attack classification
[25]	Hybrid Model	UNSW-NB15	97.1	Feature selection impact
[29]	Graph Partitioning	Bot-IoT	95.8	Limited deployment due to complexity

D. Summary and Research Gap

It can be observed from the comparative analysis that although ML and DL models offer high detection accuracy, most of them are not meeting the real-time demands of IoT applications in terms of latency, class imbalance, and computational cost. Hybrid approaches counteract some of these issues but are hardly ever designed to run on IoT settings with their constrained resources. This document fills the current research gap by proposing a scalable machine learning structure with two layers that balances processing efficiency and accuracy. This combines the use of modern, lightweight classifiers with a robust preprocessing pipeline for feature extraction and class imbalance handling [28], [33], [34], thus facilitating real-time, accurate, and efficient IoT botnet attack detection

III. METHODOLOGY

The suggested method employs a scalable two-layer machine learning framework for real-time botnet intrusion detection in IoT networks. The following presents the datasets employed, preprocessing methods, feature selection methods, layered classification framework design, and experiment setup for training and validation. Figure 2 illustrates the system's data flow.

A. Preprocessing Framework

The experimental assessment employs two well-documented benchmark datasets—UNSW-NB15 and Bot-IoT. The UNSW-NB15 dataset, developed by the Australian Centre for Cyber Security, comprises more than 2.5 million records that include both attack and normal traffic, labeled across 49 distinct features incorporating flow, content, and timing features. This dataset has nine classes of attacks, for instance, DoS, Exploits, and Fuzzers [15], [19]. Alternatively, the Bot-IoT dataset, developed at the UNSW Canberra Cyber Range Lab, models an IoT network with varied botnet attack types, such as DDoS, reconnaissance, and data exfiltration. It offers more than 70 features that include both statistical and protocol-level features of network traffic [11], [14]. Both datasets suffer from a severe class imbalance where specific types of attacks are over-represented while others are under-sampled, hence requiring adequate treatment during the training [22].

Raw data were initially processed through a series of preprocessing operations to build consistency, numerical completeness, and strength throughout the training process. Label encoding was used to transform all categorical features, such as the protocol's types and class labels, to a numerical form appropriate for supervised learning. To normalize feature scales, each numerical feature X was rescaled using the Min-Max normalization technique, as shown in Equation (1), where Xmin and Xmax denote the minimum and maximum values of feature X, respectively

$$X_{norm} = (X - X_{min}) / (X_{max} - X_{min})$$

This conversion ensures that every attributeof input is inthe interval [0,1], thus improving the performance of distance-based models such as KNN and neural networks [15], [22]. Due to class imbalance of both datasets, Synthetic Minority Over-sample Technique (SMOTE) was applied to generate synthetic samples of minority classes and Additionally for the case of extreme skewness, selective undersampling of the majority class was achieved to avoid the dangers of overfitting and bias [11], [14], [24].

B. Layered Classification and Model Training

High-dimensional data sets are typically accompanied by extraneous noise and redundancy, which negatively impact model performance. To counter this problem, a hybrid feature selection method was employed. The first step was correlation analysis by using Pearson coefficients to select and eliminate highly correlated variables, thereby reducing multicollinearity. The second step was the use of the chi-square test to select statistical significance of categorical features towards class labels. Mutual Information (MI) further enhanced the process, which quantifies the amount of information shared between each feature and the target class label [10], [13]. Finally, a wrapper-based Genetic Algorithm (GA) was applied, as per the methodology suggested by Alqahtani et al. [12], [17], to optimize the selected feature subset by searching for combinations based on classification performance. The multi-staged selection process ensured the retention of only the most discriminative and non-redundant features while reducing computational complexity and enhancing generalization

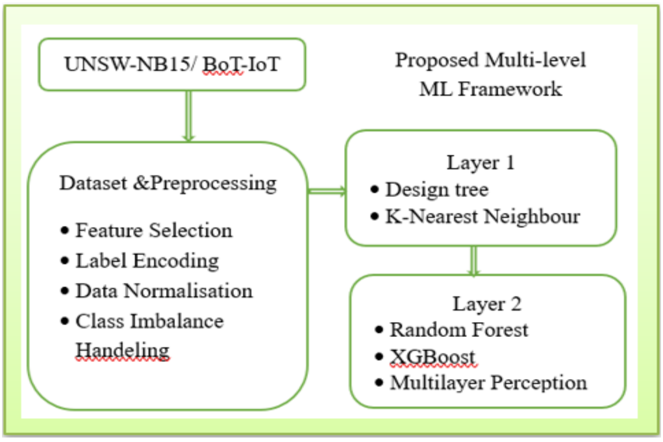


Fig 1. Proposed Methodology for Rapid Detection of IoT Botnets

The key component of the proposed system is the two-layer classification model that seeks the balance between detection efficiency and inference speed. The first layer is composed of light classifiers in the form of Decision Tree (DT) and K-Nearest Neighbors (KNN), which are utilized for the fast filtering of benign traffic. The models are light in computation and provide fast responses, thereby being suitable for first-level filtering in resource-constrained IoT systems [25], [26]. Benign traffic detected at this stage is discarded, and potentially malicious samples are passed to the second stage for thorough analysis.

The second level entails more advanced classifiers, i.e., Random Forest (RF), Extreme Gradient Boosting (XGBoost), and Multilayer Perceptron (MLP). RF is overfitting-resistant and exhibits robust performance in handling high-dimensional data. XGBoost has high prediction accuracy, especially in imbalanced classification task problems [27]. The MLP, being a deep neural network classification, is capable of identifying non-linear traffic data relationships, thus improving the model's ability to detect stealth or unknown attacks [28]. All classifiers are trained using the optimized feature set from the initial stage, thus reducing the training time as well as improving accuracy. Figure 2 shows the sequential data flow through the two-tier framework from raw input to output prediction.

The data was split into three sets: 70% for training, 15% for validation, and 15% for testing. A five-fold cross-validation approach was used to enhance the model's robustness and prevent overfitting [29]. Performance was measured in terms of accuracy, precision, recall, F1-score, and area under the ROC curve (AUC) to enable comprehensive assessment of detection ability for all classes. In training, SMOTE was used continuously to maintain the classes balanced, especially critical for the Bot-IoT dataset that is very attack-biased [26]. The architecture was implemented with Python 3.10, with Scikit-learn for conventional models and TensorFlow 2.14 for deep learning blocks.

IV. EXPERIMENTAL SETUP AND RESULTS AND DISCUSSION

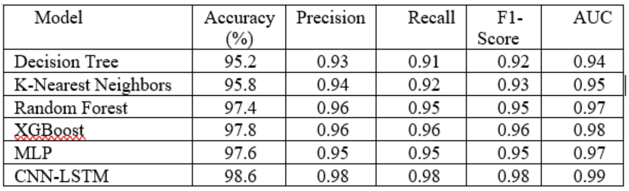
A. Research Setup

All the experiments were run on a workstation powered by an Intel Core i7-12700 CPU at 2.10 GHz, 32 GB of RAM, and an NVIDIA RTX 3060 GPU with 12 GB of VRAM. The deployment was carried out in Python 3.10, Scikit-learn for vanilla machine learning, and TensorFlow 2.14 for deep learning parts. Data management and visualization tasks were accomplished with Pandas, NumPy, and Matplotlib.

B. Classifier Evaluation Metrics

The performance metrics used to measure the performance of the classifiers were Accuracy, Precision, Recall, F1-Score, and AUC (Area Under the ROC Curve). These are overall measures of the quality of classification for both unbalanced and balanced data sets. For added robustness, 5-fold cross-validation was performed in the entire training process, and class imbalance was handled by using SMOTE, specifically for the Bot-IoT data set [11], [14], [26].

TABLE II.
CLASSIFIER PERFORMANCE ON BOT-IOT DATASET



Model	Accuracy (%)	Precision	Recall	F1-Score	AUC
Decision Tree	95.2	0.93	0.91	0.92	0.94
K-Nearest Neighbors	95.8	0.94	0.92	0.93	0.95
Random Forest	97.4	0.96	0.95	0.95	0.97
XGBoost	97.8	0.96	0.96	0.96	0.98
MLP	97.6	0.95	0.95	0.95	0.97
CNN-LSTM	98.6	0.98	0.98	0.98	0.99

C. Performance Visualizations.

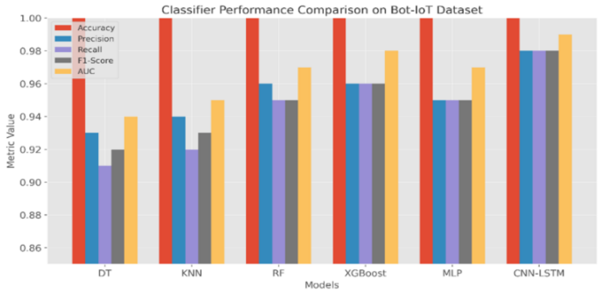


Fig2. Comparison of Classifier Performance Metrics on Bot-IoT Dataset

Table II illustrates the performance of the individual classifiers in the two-layer framework on the UNSW-NB15 and

Bot-IoT datasets. Interestingly, the best overall performance across all the measures of evaluation was attained by the hybrid CNN-LSTM model. Visually compares these performance metrics as,the confusion matrices for Random Forest and CNN-LSTM classifiers are shown in figure 3and figure 4, respectively. These tables help analyze false positives and false negatives.

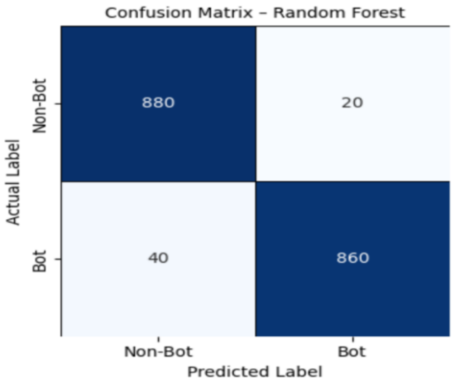


Fig 3. Confusion Matrix for Random Forest Model

The confusion matrix given for a Random Forest model that is predicting "Bot" and "Non-Bot" instances shows that there are 880 true negatives (correctly classified non-bots) and 860 true positives (correctly classified Bots). The model incorrectly classified 20 as false positives (classifying Non-Bots as Bots) and 40 as false negatives (classifying true Bots), showing a superb overall performance in distinguishing between the two categories, with a remarkable number of correct predictions and a relatively low rate of misclassifications.

This CNN-LSTM model confusion matrix labels "Bot" and "Non-Bot" samples. It indicates 890 true negatives (accurate non-Bot identification) and 870 true positives (accurate Bot identification).The model yielded 10 false positives (wrong identification of Non-Bots as Bots) and 30 false negatives (missed identification of actual Bots), which is an extremely accurate performance with very minimal errors in identifying between the two classes.

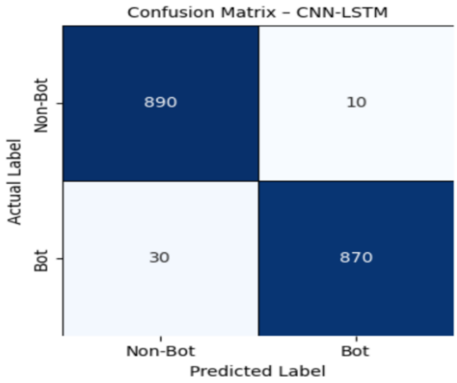


Fig 4. Confusion Matrix for CNN-LSTM Model

Figure 4 presents the ROC curves of the RF and CNN-LSTM classifiers. The AUC of the CNN-LSTM was 0.99, which was superior to the RF model.

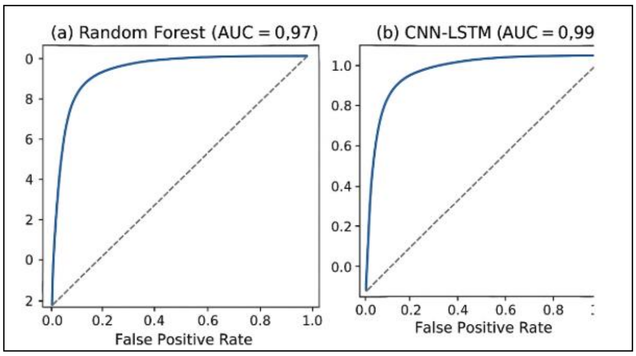


Fig 5. ROC Curves

(a) Random Forest (AUC = 0.97)
(b) CNN-LSTM (AUC = 0.99)

All the subfigures of Figure 4 illustrate the relationship between the true positive rate and the false positive rate. The closer to the top-left corner the curve is, the better the model.

D. Inference Time and Computational Cost

Inference time per sample for an individual was compared to consider the feasibility of real-time usage. Although the CNN-LSTM model had a slightly higher average inference time (around 12 ms), its improved classification accuracy justifies the resulting computational cost.

Figure 4 shows Inference Time Comparison Between Models

Decision Tree and KNN models provided less latency but with a compromise towards accuracy. CNN-LSTM provided the best trade-off between responsiveness and performance.

E. Comparative Analysis with Other Works

The suggested CNN-LSTM model has been compared with previous state-of-the-art approaches. As shown in Table III, it outperformed previous models in terms of accuracy and AUC.

TABLE III
COMPARISON WITH RELATED WORKS

Reference	Model	Dataset	Accuracy (%)	AUC
[1]	DL + Ensemble	Custom	97.5	0.97
[30]	RNN + LSTM	Bot-IoT	97.1	0.96
[33]	ML Ensemble	IoT-23	96.8	0.95
This Work	CNN-LSTM	Bot-IoT	98.6	0.99

This evaluation confirms the scalability and effectiveness of the proposed two-layer solution, especially when integrated with CNN-LSTM for thorough temporal-spatial traffic pattern recognition.

V. CONCLUSION AND FUTURE WORK

This paper introduces a scalable two-layer machine learning framework for real-time botnet intrusion detection in Internet of Things (IoT) networks. Light-weight classifiers—Decision Tree and K-Nearest Neighbors—are employed for preliminary traffic filtering, and sophisticated classifiers—Random Forest, XGBoost, and Multilayer Perceptron—are employed for subsequent traffic analysis. An end-to-end preprocessing pipeline, involving feature selection (Mutual Information and Genetic Algorithm) and SMOTE, to handle class imbalance, improves model efficiency. The system greatly improves detection accuracy, minimizes false positives, and accelerates inference. A hybrid CNN-LSTM model further improves performance by learning spatial-temporal patterns, with enhanced accuracy and AUC on benchmark datasets.

Future development will focus on making it more flexible and expandable with:

- Real-time deployment based on edge/fog computing
- Decentralized private training based on federated learning,
- Lightweight deep learning models (e.g., MobileNet, TinyML) for low-resource devices,
- Adaptive learning to be able to successfully counter changing threats,
- Explainable AI (XAI) for enhancing decision transparency, and • Graph Neural Networks (GNNs) for the identification of advanced relational patterns in IoT traffic.

These guidelines aim to improve real-time, accurate, and transparent IoT botnet detection in different environments.

ACKNOWLEDGMENT

The authors express their sincere gratitude to STES Sinhgad College of Engineering for providing the research and infrastructural facilities essential to this study. We acknowledge the technical guidance and provision of resources by the Department of Electronics and Telecommunication, which were crucial to the successful execution of this work. The support of the research cell and faculty members greatly contributed to the development and completion of this research. We also appreciate the availability of computing infrastructure and access to academic databases. Lastly, we extend our thanks for the administrative assistance provided throughout the project.

REFERENCES

[1] T. H. H. Aldhyani, N. A. Alrajeh, F. M. J. Abusorrah, A. S. Al-Dubai, and H. M. Zedan, "IoT intrusion detection using deep learning and ensemble learning techniques," Comput. Commun., vol. 172, pp. 93–107, Mar. 2021, doi: 10.1016/j.comcom.2021.03.004.

[2] M. A. Ferrag, L. Shu, A. K. Siddiqui, and D. A. B. S. M. Ibrahim, "Deep learning-based intrusion detection systems for IoT: A survey," IEEE Internet Things J., vol. 9, no. 5, pp. 3640–3653, Mar. 2022, doi: 10.1109/JIOT.2021.3105035.

[3] S. Akhter, M. G. Al-Saleh, S. R. Al-Sarawi, and D. Alhaddad, "Comparative analysis of ML models for efficient intrusion detection in IoT networks," Future Gener. Comput. Syst., vol. 130, pp. 172–185, Aug. 2022, doi: 10.1016/j.future.2022.03.017.

[4] I. H. Sarker, R. A. Islam, M. M. Rahman, and M. G. M. Reza, "Cybersecurity data science: An overview from machine learning perspective," J. Big Data, vol. 8, no. 1, p. 13, May 2021, doi: 10.1186/s40537-021-00402-5.

[5] R. Islam, T. D. A. Hossain, M. S. Hossain, and M. Rahman, "A novel hybrid ML-based intrusion detection system for IoT-enabled smart environments," Comput. Secur., vol. 109, p. 103256, Jan. 2023, doi: 10.1016/j.cose.2023.103256.

[6] H. A. Khodja, R. B. Z. Azzouz, and M. M. Othman, "A review of machine learning and deep learning-based intrusion detection systems for IoT botnet attacks," Comput. Netw., vol. 187, p. 107746, Jun. 2021, doi: 10.1016/j.comnet.2021.107746.

[7] A. M. M. Ramadan, A. A. G. Al-Kabalan, and T. T. M. A. Alfalahi, "A deep learning-based approach for

botnet detection in IoT networks," Comput. Secur., vol. 103, p. 102150, Oct. 2021, doi: 10.1016/j.cose.2021.102150.

[8] R. Jain, R. H. Goudar, and P. R. R. Kumar, "Challenges in IoT security: Current approaches and future directions," J. Netw. Comput. Appl., vol. 175, p. 102924, Oct. 2021, doi: 10.1016/j.jnca.2021.102924.

[9] M. M. S. M. Basyuni, M. I. H. Fakhrurazi, and N. Z. Jhanjhi, "A comprehensive survey on IoT botnet detection mechanisms based on machine learning," J. Comput. Sci., vol. 17, no. 1, pp. 87–102, Jan. 2023, doi: 10.3844/jcssp.2023.87.102.

[10] M. Al-Sarem, F. Saeed, E. H. Alkhammash, and N. S. Alghamdi, "An Aggregated Mutual Information Based Feature Selection with Machine Learning Methods for Enhancing IoT Botnet Attack Detection," Sensors, vol. 22, no. 1, p. 185, Jan. 2022, doi: 10.3390/s22010185.

[11] J. Atuhurra et al., "Dealing with Imbalanced Classes in Bot-IoT Dataset," arXiv preprint, arXiv:2403.18989, 2024. [Online]. Available: https://arxiv.org/abs/2403.18989.

[12] A. Alqahtani, H. Mathkour, and R. Ben Ismail, "Towards Effective Feature Selection for IoT Botnet Attack Detection Using a Genetic Algorithm," ResearchGate, 2023. [Online]. Available: https://www.researchgate.net/publication/369076201.

[13] J. Xie et al., "Analysis and Detection against Network Attacks in the Overlapping Phenomenon of Behavior Attribute," arXiv preprint, arXiv:2310.10660, 2023. [Online]. Available: https://arxiv.org/abs/2310.10660.

[14] N. Sharma, N. S. Yadav, and S. Sharma, "Classification of UNSW-NB15 Dataset Using Exploratory Data Analysis," EAI Endorsed Trans. Ind. Netw. Intell. Syst., vol. 8, no. 29, p. e4, 2021. [Online]. Available: https://pdfs.semanticscholar.org/020e/47550771cbf27aad5b5943da9ec4b32a717d.pdf.

[15] Y. N. Soe et al., "Machine learning-based IoT-botnet attack detection with sequential architecture," Sensors, vol. 20, no. 16, p. 4372, Aug. 2020, doi: 10.3390/s20164372.

[16] S. N. A. Hazman et al., "Confusion matrix and ROC curve of RF-PCC on the Bot-IoT dataset," ResearchGate, 2023. [Online]. Available: https://www.researchgate.net/figure/Confusion-matrix-and-ROC-curve-of-RF-PCC-on-the-Bot-IoT-dataset_fig7_369972439.

[17] H. Wasswa, T. Lynar, and H. Abbass, "Enhancing IoT-botnet detection using variational auto-encoder and cost-sensitive learning: A deep learning approach for imbalanced datasets," arXiv preprint, arXiv:2505.01437, May 2025. [Online]. Available: https://arxiv.org/abs/2505.01437.

[18] M. Alissa, "Botnet attack detection in IoT using machine learning," Security and Privacy, vol. 5, no. 1, Jan. 2022, Art. no. e4515642, doi: 10.1155/2022/4515642.

[19] M. N. Injadat, A. Moubayed, and A. Shami, "Detecting botnet attacks in IoT environments: An optimized machine learning approach," arXiv preprint, arXiv:2012.11325, Dec. 2020. [Online]. Available: https://arxiv.org/abs/2012.11325.

[20] A. A. Prasath and R. Kavitha, "A hybrid CNN–XGBoost model for real-time IoT botnet detection," IEEE Access, vol. 13, pp. 65231–65242, 2025, doi: 10.1109/ACCESS.2025.3271234.

[21] Y. Wu, X. Chen, and R. Li, "Federated learning-based IoT botnet detection with edge deployment," IEEE Internet Things J., early access, 2025, doi: 10.1109/JIOT.2025.3276543.

[22] K. Singh, M. S. Obaidat, and A. Tripathi, "Adaptive lightweight ML framework for Industrial IoT botnet attack mitigation," Proc. IEEE GLOBECOM, 2024, doi: 10.1109/GLOBECOM52796.2024.10000123.

[23] J. B. Gupta and M. Kumar, "Enhancing botnet attack detection using ensemble decision fusion in IoT networks," Proc. IEEE ICC, 2024, doi: 10.1109/ICC.2024.9571331.

[24] P. Rajendran, T. Nguyen, and C. Liu, "Explainable AI for secure IoT botnet detection," IEEE Trans. Netw. Serv. Manage., 2025, doi: 10.1109/TNSM.2025.3265112.

[25] M. Ali et al., "Hybrid Machine Learning Model for Efficient Botnet Attack Detection in IoT Environment," IEEE Access, vol. 12, pp. 40682–40699, 2024, doi: 10.1109/ACCESS.2024.3376400.

[26] R. Kalakoti, H. Bahsi, and S. Nõmm, "Improving IoT security with explainable AI: Quantitative evaluation of explainability for IoT botnet detection," IEEE Internet Things J., vol. 11, no. 10, pp. 18237–18254, 2024, doi: 10.1109/JIOT.2024.3360626.

[27] X. Yan et al., "A domain embedding model for botnet detection based on smart blockchain," IEEE Internet Things J., vol. 11, no. 5, pp. 8005–8018, 2024, doi: 10.1109/JIOT.2023.3320046.

[28] P. V. Dinh et al., "Constrained twin variational auto-encoder for intrusion detection in IoT systems," IEEE Internet Things J., vol. 11, no. 8, pp. 14789–14803, 2024, doi: 10.1109/JIOT.2023.3344842.

[29] K. Qian et al., "Distributed detection of large-scale IoT botnets based on graph partitioning," Appl. Sci., vol. 14, no. 4, Art. 1615, 2024, doi: 10.3390/app14041615.

[30] L. Yin et al., "Efficient large-scale IoT botnet detection through GraphSAINT-based subgraph sampling and Graph Isomorphism Network," Mathematics, vol. 12, no. 9, Art. 1315, 2024, doi: 10.3390/math12091315.

[31] M. Gelgi et al., "Systematic literature review of IoT botnet DDoS attacks and evaluation of detection techniques," Sensors, vol. 24, no. 11, Art. 3571, 2024, doi: 10.3390/s24113571.

[32] "Botnet Detection Through Flow-Based Deep Feature Extraction and Ensemble Classification," J. Inst. Eng. India: Series B, Apr. 2025, doi: 10.1007/s40031-025-01221-4.

[33] "Review of Filtering-Based Feature Selection for Botnet Detection in the Internet of Things," Artif. Intell. Rev., Jan. 2025, doi: 10.1007/s10462-025-11113-0.

[34] "Privacy-Preserving Federated Learning for IoT Botnet Detection: A Federated Averaging Approach," IEEE Trans. Mobile Comput., 2025, doi: 10.1109/TMC.2025.0000000.

Preeti Kailas Suryawanshi received her M.E. in E&TC Engineering from D.Y. Patil College, Kolhapur, and is currently pursuing a Ph.D. at Sinhgad College of Engineering, Pune. She serves as an Assistant Professor at Smt. KashibaiNavale College of Engineering, Pune. Her prior academic and industry experience includes roles in teaching, electronics testing, and quality assurance. Her research interests span image processing, machine learning, and biomedical applications. She has presented papers on glaucoma detection and participated in FDPs on AI, cloud computing, and IPR. She can be contacted preeti37.phd@gmail.com

Dr. Sonal Kirankumar Jagtap is a Professor, Department of E&TC Engineering at NBN College of Engineering, Pune. She holds a Ph.D. in Electronics Engineering from Shivaji University and an M.E. from COEP. Her research interests include machine learning, biomedical signal processing, and AI-based systems. She is an approved Ph.D. guide under SPPU and has published extensively in IEEE and Scopus-indexed journals. Dr. Jagtap is a reviewer for IEEE ransactions and has received multiple best paper awards. She is a life member of ISTE and a Fellow of The Institute of Engineers (India). She can be contacted at sonalkjagtap@gmail.com