Comparative Evaluation of Feature Selection Strategies and Machine Learning Classifiers for IoT Botnet Detection

Fong Zi Khang, Mohd Faizal Abdollah*, Warusia Yassin, Raihana Syahirah Abdullah, Nurhashikin Mohd Salleh

Faculty of Information and Communication Technology, Universiti Teknikal Malaysia Melaka, Melaka, Malaysia

***Corresponding Author**

DOI: https://doi.org/10.47772/IJRISS.2026.100700749

Received: 28 July 2026; Accepted: 03 August 2026; Published: 11 August 2026

ABSTRACT

The rapid expansion of Internet of Things (IoT) devices has increased the need for accurate botnet detection methods that can operate with a compact set of network-traffic features. This study presents a controlled comparative evaluation of eight feature selection strategies and six machine learning classifiers using the N-BaIoT dataset. The feature selection strategies cover statistical, projection-based, recursive, hybrid, and ensemble approaches. Each method was evaluated at five feature-set sizes comprising 2, 3, 5, 10, and 15 features. Six classifiers—Decision Tree, Random Forest, Gradient Boosting, support vector machine with a radial basis function kernel, Logistic Regression, and K-Nearest Neighbours—were tested using fixed settings across all feature configurations. The complete design produced 240 model configurations that were assessed on the same evaluation set using accuracy, recall, precision, F1-score, training time, and inference latency. The results show that the RFE-Hybrid Lasso method produced the strongest overall feature subsets, with an average accuracy of 98.94% and an average F1-score of 98.09%. The best individual configuration combined RFE-Hybrid Lasso, 15 selected features, and Decision Tree, achieving 99.63% accuracy, 98.87% recall, 99.41% precision, and a 99.14% F1-score. Decision Tree also recorded the highest average classifier performance, reaching 99.18% accuracy with a training time of 2.1 minutes and an inference latency of 0.18 ms. The findings indicate that RFE-based hybrid selection improves detection performance over single statistical methods and that Decision Tree offers the most favourable performance–cost balance among the evaluated classifiers.

Keywords: IoT security, botnet detection, feature selection, recursive feature elimination, machine learning

INTRODUCTION

The Internet of Things has expanded the number of cameras, digital video recorders, household devices, and embedded systems connected to home and organisational networks. Many IoT devices operate with limited computing resources and receive irregular security updates, making them attractive targets for malware and botnet operators. Mirai and Bashlite demonstrated how attackers can compromise large numbers of devices through weak credentials and unpatched services and then coordinate them for distributed denial-of-service attacks, scanning, and other malicious activity [1],[2],[3].

Network-based detection offers a practical way to identify compromised IoT devices without installing complex security software on each endpoint. Botnet traffic can exhibit measurable patterns in connection duration, packet timing, packet sizes, protocol flags, and traffic volume [3], [4]. However, network-flow datasets may contain many related or weak variables. Presenting every available variable to a classifier can increase training cost,

reduce interpretability, and introduce unnecessary variation. Feature selection is therefore an important stage in developing an efficient detection model.

Previous studies have applied statistical selection, recursive feature elimination, dimensionality reduction, ensemble methods, and machine learning classifiers to IoT botnet detection [5],[6]. Reported performance is often high, but comparisons across studies remain difficult because the datasets, class distributions, feature counts, parameter settings, and evaluation partitions differ. A controlled experiment that evaluates several feature selection strategies and classifiers on the same data can provide clearer evidence about which combinations work best under comparable conditions.

Research Objective and Contribution

This study has one main objective: to compare eight feature selection strategies across five feature-set sizes and six machine learning classifiers under a consistent experimental design, and then identify the configuration that provides the strongest detection performance with a reasonable computational cost. The contribution is a large within-study comparison of 240 configurations evaluated on the same dataset partition and performance measures. Rather than proposing a new detection algorithm, the study provides empirical evidence on the relative behaviour of statistical, projection-based, RFE-based, hybrid, and ensemble feature selection strategies. It also identifies the feature-selection and classifier combination that performed best in the experiment and reports a smaller feature-set alternative for settings where reducing input variables is desirable.

Related Work

Machine Learning for IoT Botnet Detection

Machine learning has become a common approach to network-based botnet detection because classifiers can learn patterns from traffic statistics that are difficult to express through fixed signatures [7]. Decision Trees generate hierarchical rules that can represent non-linear relationships and are comparatively easy to interpret. Random Forest and Gradient Boosting combine multiple trees to improve predictive stability, although they usually require more computation than a single tree [8]. Support Vector Machines can build complex decision boundaries with kernel functions but can be costly when the dataset is very large [9]. Logistic Regression: efficient linear classification [10]. A method of classification that compares the new record to records stored in the classes is K-Nearest Neighbours [11]. They come with various compromises in terms of accuracy, training time, inference cost and interpretability.

Various studies have reported high performance in the detection of IoT botnets with tree-based classifiers [6],[12],[13],[14], neural networks [15],[16],[17],[18] and transfer learning [9]. Bahsi et al. investigated dimensionality reduction for machine-learning based IoT botnet detection [16] and Guerra-Manzanares et al. investigated hybrid feature selection models [6]. Alqahtani et al. combined feature selection with optimised Gradient Boosting [13], and Desai et al. proposed a hybrid detection approach [18]. These studies establish the value of feature reduction, but their experimental differences make direct method ranking difficult.

Feature Selection Strategies

Statistical methods select variables according to their relationship with the target. Lasso can shrink weak coefficients toward zero, providing an embedded form of variable selection [20]. Chi-Square measures the association between a feature and the class label and is widely used as a filter method [21]. Random Projection reduces dimensionality by mapping data into a lower-dimensional space while approximately preserving distances [22]. These approaches are generally efficient, but they may not capture how several variables work together within a non-linear classifier.

Recursive Feature Elimination repeatedly fits an estimator, ranks the variables, and removes the least useful features until it reaches the required subset size. Its effectiveness depends on the estimator used during elimination. Hybrid methods combine a fast initial selection stage with RFE so that the recursive stage evaluates a smaller and more relevant candidate set. Ensemble voting combines rankings from multiple selectors to reduce dependence on a single selection rule. Hybrid selection has shown promise in high-dimensional classification, but its advantage should be demonstrated under controlled comparisons rather than assumed [6], [23].

Research Gap

The literature shows that many feature selection methods and classifiers can achieve high IoT botnet detection accuracy. However, fewer studies evaluate a broad set of feature-selection strategies, multiple subset sizes, and several classifiers while holding the data partition and model settings constant. As a result, it is often unclear whether reported differences arise from the selection method, the classifier, the number of features, or the evaluation procedure. This study addresses that practical gap through a uniform 240-configuration comparison.

MATERIALS AND METHODS

Dataset and Preprocessing

More recent studies have continued using this dataset suach as federated learning and blockchain-enhanced intrusion detection on the N-BaIoT dataset [25], as well as innovative machine learning paradigms for botnet attack detection [26]. The experiment used the N-BaIoT dataset, which contains 7,062,606 network-flow records from IoT devices exposed to Mirai and Bashlite activity [15]. The dataset contains 116 statistical traffic variables, including packet-size measures, packet inter-arrival times, protocol-flag distributions, traffic volume, and entropy-related characteristics. The records were treated as a binary classification problem consisting of benign and malicious traffic. The approximate class distribution was 85% benign and 15% malicious.

A stratified 70:30 split produced 4,943,824 training records and 2,118,782 evaluation records. Stratification preserved the class proportions in both partitions. StandardScaler was fitted using the training data and then applied to the evaluation data. This procedure prevented information from the evaluation set from influencing the scaling parameters. The same training and evaluation partitions were used for all configurations.

Feature Selection Strategies

The study evaluated eight strategies: Lasso, Chi-Square, Random Projection, basic linear RFE, RFE with a Gradient Boosting estimator, RFE-Hybrid Lasso, RFE-Hybrid Chi-Square, and Ensemble Voting. Each strategy produced subsets containing 2, 3, 5, 10, and 15 features. The RFE-Hybrid methods used a statistical selector to reduce the initial candidate set before recursive elimination, while Ensemble Voting combined evidence from more than one ranking method. The same requested feature counts were applied to every strategy so that their outputs could be compared directly.

Classifiers and Experimental Design

Six classifiers were evaluated: Decision Tree, Random Forest, Gradient Boosting, support vector machine with a radial basis function kernel, Logistic Regression, and K-Nearest Neighbours. Each classifier retained the same parameter setting across all feature-selection methods and feature counts. Combining eight selection strategies, five feature-set sizes, and six classifiers generated 240 configurations. Every configuration was trained on the same training partition and assessed on the same evaluation partition. The implementation used Python and the scikit-learn machine learning library [24].

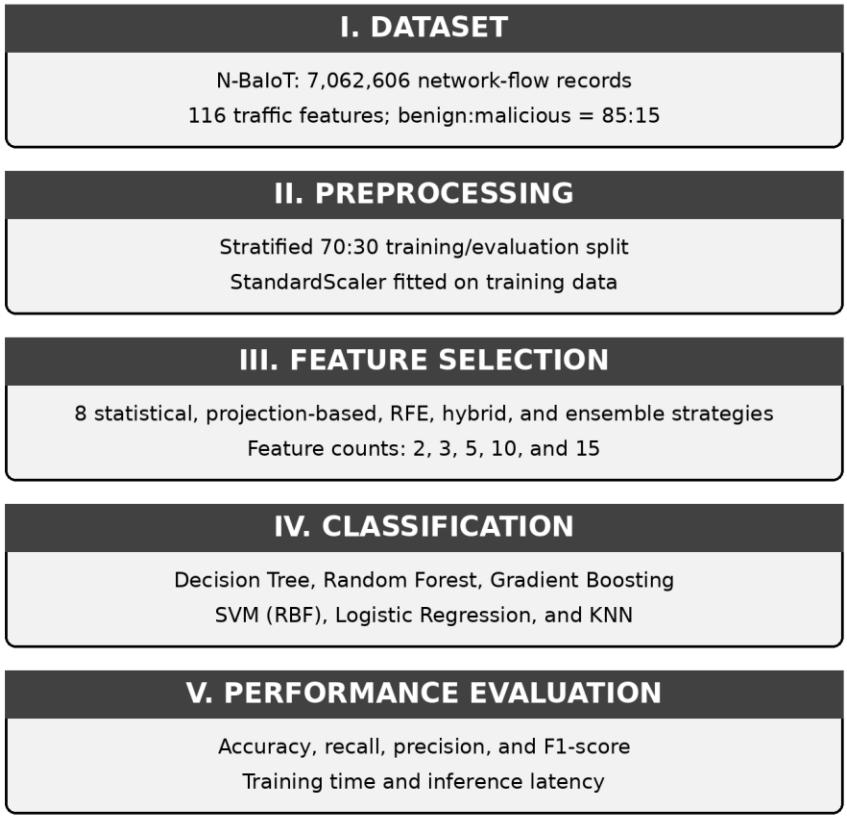

Figure 1. Experimental pipeline for the comparative IoT botnet detection study

Figure 1 summaries the experimental process. The design starts with the N-BaIoT traffic records, applies a common preprocessing procedure, generates feature subsets through eight strategies, evaluates each subset with six classifiers, and compares predictive performance and computational cost.

Evaluation Measures

Performance was assessed using accuracy, recall, precision, and F1-score. Accuracy measures the proportion of correctly classified records. Recall measures the proportion of malicious records correctly detected, while precision measures the reliability of malicious predictions. F1-score provides a combined measure of precision and recall. Training time and per-record inference latency were also recorded to examine computational differences among classifiers. The comparison is descriptive and is limited to the reported experimental partitions and settings.

RESULTS AND DISCUSSION

Top-Performing Configurations

Table 1. Top 10 model configurations ranked by accuracy

Rank	Method	Features	Classifier	Accuracy	Recall	Precision	F1-score
1	RFE-Hybrid (Lasso)	15	Decision Tree	99.63%	98.87%	99.41%	99.14%
2	RFE-Hybrid (Chi-Square)	15	Decision Tree	99.61%	98.75%	99.38%	99.06%
3	RFE (Gradient Boosting base)	15	Decision Tree	99.58%	98.62%	99.35%	99.00%
4	Ensemble Voting	15	Decision Tree	99.57%	98.58%	99.32%	98.95%
5	RFE-Hybrid (Lasso)	10	Decision Tree	99.42%	98.15%	99.21%	98.68%

Rank	Method	Features	Classifier	Accuracy	Recall	Precision	F1-score
6	RFE (Gradient Boosting base)	15	Random Forest	99.38%	97.92%	99.18%	98.55%
7	RFE-Hybrid (Chi-Square)	10	Decision Tree	99.37%	98.08%	99.16%	98.62%
8	Ensemble Voting	10	Decision Tree	99.35%	97.95%	99.12%	98.54%
9	Lasso	15	Decision Tree	99.28%	97.78%	99.05%	98.41%
10	Chi-Square	15	Decision Tree	99.25%	97.65%	99.02%	98.34%

Table 1 shows that the strongest configuration combined RFE-Hybrid Lasso, 15 features, and Decision Tree. This combination has achieved 99.63% accuracy, 98.87% for recall, 99.41% precision, and 99.14% F1-score. RFE-Hybrid Chi-Square ranked second with 99.61% accuracy. Gradient Boosting-based RFE and Ensemble Voting followed closely, also using 15 features and Decision Tree. These rankings show that the highest-performing configurations depended on both the selected feature subset and the classifier.

Hybrid and model-based approaches occupied the first eight positions. Standalone Lasso and Chi-Square appeared only in ninth and tenth place, respectively. Decision Tree appeared in nine of the ten leading configurations, while Random Forest appeared once. The ranking therefore provides two consistent findings: RFE-based or ensemble selection generated the most effective subsets, and Decision Tree used those subsets more effectively than the alternative classifiers in most high-ranking combinations.

The fifth-ranked configuration is also relevant for applications that prefer fewer input variables. RFE-Hybrid Lasso with 10 features and Decision Tree achieved 99.42% accuracy, only 0.21 percentage points below the 15-feature configuration while using one-third fewer variables. This result does not by itself prove lower memory or energy consumption, but it identifies a compact alternative for later deployment testing.

Comparison of Feature Selection Strategies

Table 2. Average performance across classifiers and feature counts

Method	Avg. Accuracy	Avg. Recall	Avg. Precision	Avg. F1-score	Std. Dev.
RFE-Hybrid (Lasso)	98.94%	97.42%	98.76%	98.09%	0.78%
RFE + Gradient Boosting	98.87%	97.28%	98.68%	97.98%	0.85%
RFE-Hybrid (Chi-Square)	98.81%	97.18%	98.62%	97.90%	0.82%
Ensemble Voting	98.65%	96.95%	98.48%	97.72%	1.02%
RFE Basic (Linear)	98.42%	96.68%	98.25%	97.47%	1.15%
Lasso	97.85%	95.92%	97.58%	96.75%	1.28%
Chi-Square	97.62%	95.68%	97.35%	96.51%	1.42%
Random Projection	96.18%	94.12%	95.87%	94.99%	2.15%

Table 2 confirms that RFE-Hybrid Lasso produced the highest average performance across classifiers and feature counts. It attained 98.94% average accuracy, 97.42% recall, 98.76% precision and 98.09% F1-score. This also reported the lowest dispersion of 0.78%, indicating it had the lowest variance in combinations of classifiers and feature-counts across the tested combinations.

The average accuracy of the RFE-Hybrid Lasso was 1.09 percentage points higher than that of the standalone. Lasso is 1.32 percentage points better than Chi-Square and 2.76 percentage points better than Random Projection. RFE-Hybrid Chi-Square achieved 98.81% average accuracy, exceeding standalone Chi-Square by 1.19 percentage points. These results support the central contribution of the paper: combining an initial statistical

filter with recursive model-based elimination produced stronger feature subsets than using the corresponding statistical method alone.

The ranking also shows that model-based evaluation was important. RFE with Gradient Boosting achieved 98.87% average accuracy, while basic linear RFE achieved 98.42%. The result suggests that the estimator used during recursive elimination affected the quality of the final subset. Ensemble Voting also performed better than the three standalone statistical or projection-based methods, although it did not exceed the leading RFE-based strategies.

Classifier Performance and Computational Cost

Table 3. Average performance and computational cost by classifier

Classifier	Avg. Accuracy	Avg. Recall	Avg. Precision	Avg. F1-score	Training Time	Inference Latency
Decision Tree	99.18%	98.25%	99.02%	98.64%	2.1 min	0.18 ms
Random Forest	98.75%	97.68%	98.58%	98.13%	8.5 min	0.42 ms
Gradient Boosting	98.42%	97.15%	98.28%	97.71%	12.3 min	0.28 ms
SVM (RBF)	97.92%	96.85%	97.68%	97.26%	15.8 min	3.12 ms
Logistic Regression	96.58%	95.22%	96.35%	95.78%	1.2 min	0.12 ms
K-Nearest Neighbours	95.42%	94.18%	95.15%	94.66%	0.05 min	28.45 ms

Decision Tree achieved the highest average result across all predictive measures which are 99.18% accuracy, 98.25% recall, 99.02% precision, and 98.64% F1-score. Random Forest ranked second with 98.75% accuracy, followed by Gradient Boosting at 98.42%. Decision Tree therefore exceeded Random Forest by 0.43 percentage points in average accuracy and exceeded Gradient Boosting by 0.76 percentage points.

The computational measurements strengthen the case for Decision Tree. Its training time was 2.1 minutes, compared with 8.5 minutes for Random Forest, 12.3 minutes for Gradient Boosting, and 15.8 minutes for SVM. Its inference latency was 0.18 ms, lower than Random Forest, Gradient Boosting, SVM, and K-Nearest Neighbours. Logistic Regression achieved a slightly lower inference latency of 0.12 ms and trained in 1.2 minutes, but its average accuracy was 2.60 percentage points below Decision Tree. K-Nearest Neighbours required little training time but had the highest inference latency at 28.45 ms and the lowest predictive performance.

These findings do not indicate that Decision Tree was the fastest classifier in every measurement. Instead, they show that it provided the most favourable balance between detection performance and computational cost among the evaluated algorithms. This distinction is important because a classifier intended for frequent or real-time prediction should be assessed using both predictive metrics and operational requirements.

Features Selected by RFE-Hybrid Lasso

Table 4. Top 15 features selected by RFE-Hybrid Lasso

Rank	Feature	Category
1	Duration	Temporal
2	TCP Flag Distribution (S)	Protocol
3	Packet Interval Mean	Temporal

Rank	Feature	Category
4	Packet Size Entropy	Entropy
5	Packet Size Standard Deviation	Packet Size
6	Packet Interval Standard Deviation	Temporal
7	Total Packets	Behavioural
8	Total Bytes	Behavioural
9	TCP Flag Distribution (A)	Protocol
10	Packet Interval Entropy	Entropy
11	TCP Flag Distribution (F)	Protocol
12	Packet Size Maximum	Packet Size
13	Packet Interval Maximum	Temporal
14	UDP Flag Count	Protocol
15	Packet Size Minimum	Packet Size

The selected subset covers five complementary categories: temporal behaviour, protocol activity, entropy, packet-size variation, and aggregate traffic volume. Duration ranked first, followed by the distribution of the TCP SYN flag and mean packet interval. These variables reflect how long communication persists, how connections are initiated, and how regularly packets are exchanged. Packet-size entropy and packet-interval entropy capture traffic regularity, while total packets and total bytes represent overall communication volume.

The presence of several TCP flag distributions suggests that connection-state behaviour is useful for distinguishing normal IoT communication from botnet activity. Packet-size minimum, maximum, and standard deviation indicate that both the range and consistency of packet sizes contribute to detection. The feature composition supports the use of a hybrid selector because the final subset is not dominated by a single measurement family. Instead, it combines several aspects of network behaviour that can jointly describe automated malicious communication.

Overall Interpretation

The experiment answers the research objective through a consistent pattern across the three main comparisons. First, RFE-based hybrid methods outperformed single statistical methods on average. Second, Decision Tree produced the best average classifier results and dominated the highest-ranked configurations. Third, the strongest individual model used 15 features, while a 10-feature version retained most of its accuracy. The evidence therefore supports selecting RFE-Hybrid Lasso with Decision Tree when predictive performance is the main priority and considering the 10-feature version when a smaller input set is preferred.

The contribution is comparative rather than algorithmic. The study does not claim that RFE-Hybrid Lasso or Decision Tree will be optimal for every IoT dataset. Its value lies in evaluating a broad set of alternatives under the same conditions and reporting the resulting trade-offs clearly. This evidence can guide researchers and practitioners who need a defensible starting point for building a compact network-flow botnet detector.

Limitations

The findings are subject to several limitations. The experiment used one dataset containing traffic from a limited set of IoT devices and the Mirai and Bashlite botnet families. Performance may differ for other devices, attack types, network conditions, or more recent botnet behaviour. The 70:30 split provides a controlled within-dataset comparison but does not establish performance on an external dataset. Future studies should use device-grouped, temporal, or cross-dataset evaluation to test whether the selected configuration transfers to previously unseen environments.

The computational results report training time and inference latency but do not include memory use, model size, energy consumption, or measurements on an actual edge device. The results therefore identify a promising performance–cost configuration rather than a completed edge deployment. The interpretation is also limited to the fixed parameter settings and computing environment used in this experiment; broader hyperparameter tuning or different hardware may change the observed rankings and timing results.

CONCLUSION

This study compared eight feature selection strategies, five feature-set sizes, and six machine learning classifiers for IoT botnet detection, producing 240 controlled configurations. RFE-Hybrid Lasso achieved the highest average feature-selection performance, with 98.94% accuracy and a 98.09% F1-score. It exceeded standalone Lasso, Chi-Square, and Random Projection by 1.09, 1.32, and 2.76 percentage points in average accuracy, respectively. The best individual configuration combined RFE-Hybrid Lasso, 15 features, and Decision Tree, achieving 99.63% accuracy, 98.87% recall, 99.41% precision, and a 99.14% F1-score.

Decision Tree was the strongest classifier overall, achieving 99.18% average accuracy with a training time of 2.1 minutes and an inference latency of 0.18 ms. The 10-feature RFE-Hybrid Lasso and Decision Tree configuration achieved 99.42% accuracy, providing a smaller alternative with a limited decrease in predictive performance. The results show that RFE-based hybrid selection can improve feature quality over single statistical methods and that Decision Tree provides the best performance–cost balance among the evaluated classifiers. Further work should validate these findings on additional IoT datasets and actual edge hardware.

Declarations

Data Availability

The N-BaIoT dataset used in this study is publicly available through the Kaggle data repository [25].

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

Fong Zi Khang conducted the experiment, analysed the results, and prepared the initial manuscript. Mohd. Faizal Abdollah and Warusia Yassin supervised the research, reviewed the methodology, and revised the manuscript. All authors approved the final manuscript.

ACKNOWLEDGEMENT

The authors would like to express their sincere gratitude to Universiti Teknikal Malaysia Melaka (UTeM) for the invaluable support, resources, and facilities provided throughout this research. The university's continuous assistance, including access to computing infrastructure, laboratory facilities, and technical resources, was instrumental in enabling the successful completion the research.

REFERENCES

1. A. Marzano, David Alexander and Osvaldo Fonseca. (2018). The Evolution of Bashlite and Mirai IoT Botnets. IEEE Symposium on Computers and Communications (ISCC). 813–818, doi: 10.1109/iscc.2018.8538636.
2. C. Kolias, G. Kambourakis, A. Stavrou, and J. Voas. (2017). DDoS in the IoT: Mirai and Other Botnets. Computer, 50(7). doi: 10.1109/mc.2017.201.
3. E. Bertino and N. Islam. (2017). Botnets and Internet of Things Security. Computer, 50(2). doi: 10.1109/mc.2017.62.

4. R. Doshi, N. Apthorpe, and N. Feamster. (2108). Machine Learning DDoS Detection for Consumer Internet of Things Devices. IEEE Symposium on Security and Privacy Workshops
5. Y. Dhote, S. Agrawal, and A. J. Deen. (2015). A Survey on Feature Selection Techniques for Internet Traffic Classification. International Conference on Computational Intelligence and Communication Networks.
6. A. Guerra-Manzanares, H. Bahşi, and S. Nõmm. (2019). Hybrid Feature Selection Models for Machine Learning Based Botnet Detection in IoT Networks. International Conference on Cyberworlds.
7. R. Doriguzzi-Corin, S. Millar, S. Scott-Hayward, J. Martínez-del-Rincón, and D. Siracusa. (2020). Lucid: A Practical, Lightweight Deep Learning Solution for DDoS Attack Detection," IEEE Transactions on Network and Service Management. 17(2), 876–889. doi: 10.1109/tnsm.2020.2971776.
8. H. Blockeel, L. Devos, B. Frénay, G. Nanfack, and S. Nijssen. (2023). Decision trees: from efficient prediction to responsible AI. Frontiers in Artificial Intelligence. (6), 124553–1124553. doi: 10.3389/frai.2023.1124553.
9. C. Cortes and V. Vapnik. (1995). Support-vector networks. Machine Learning, 20(3), 273–297. doi: 10.1007/bf00994018.
10. G.-X. Yuan, C.-H. Ho, and C. Lin. (2012). Recent Advances of Large-Scale Linear Classification. Proceedings of the IEEE. 100(9), 2584–2603. doi: 10.1109/jproc.2012.2188013.
11. I. H. Sarker, A. S. M. Kayes, and P. Watters. (2019). Effectiveness Analysis of Machine Learning Classification Models for Predicting Personalized Context-aware Smartphone Usage. Journal Of Big Data. 6(1). doi: 10.1186/s40537-019-0219-y.
12. M. Injadat, A. Moubayed, and A. Shami. (2020). Detecting Botnet Attacks in IoT environments: An Optimized Machine Learning Approach," in Proc. 32nd International Conference on Microelectronics.
13. M. Alqahtani, H. Mathkour, and M. M. B. Ismail. (2020). IoT Botnet Attack Detection Based on Optimized Extreme Gradient Boosting and Feature Selection. Sensors. 20(21), 6336–6336, .
14. S. Susanto, D. Stiawan, M. A. S. Arifin, Mohd. Y. Idris, and R. Budiarto. (2020). IoT Botnet Malware Classification Using Weka Tool and Scikit-learn Machine Learning. 7th International Conference on Electrical Engineering, Computer Sciences and Informatics.
15. F. Abbasi, M. Naderan, and S. E. Alavi. (2021). Anomaly Detection In Internet Of Things Using Feature Selection And Classification Based On Logistic Regression And Artificial Neural Network On N-Baiot Dataset. 5th International Conference on Internet of Things and Applications (IoT)
16. H. Bahşi, S. Nõmm, and F. B. L. Torre. (2018). Dimensionality Reduction for Machine Learning Based IoT Botnet Detection. International Conference on Control, Automation, Robotics and Vision
17. S. Susanto, D. Stiawan, M. A. S. Arifin, J. Rejito, Mohd. Y. Idris, and R. Budiarto. (2021). A Dimensionality Reduction Approach for Machine Learning Based IoT Botnet Detection. 8th International Conference on Electrical Engineering, Computer Science and Informatics.
18. M. G. Desai, Y. Shi, and K. Suo, "A Hybrid Approach for IoT Botnet Attack Detection. (2021). 2021 IEEE 12th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)
19. S. Rabhi, T. Abbes, and F. Zarai. (2023). Transfer learning-based Mirai botnet detection in IoT networks. International Conference on Innovations in Intelligent Systems and Applications (INISTA).
20. R. Muthukrishnan and R. Rohini. (2016). LASSO as a feature selection method for weather forecasting. International Conference on Communication and Electronics Systems.
21. H. L. And, H. Liu, and R. Setiono. (1995). Chi2: Feature Selection and Discretization of Numeric Attributes. 7th IEEE International Conference on Tools with Artificial Intelligence
22. W. B. Johnson and J. Lindenstrauss. (1984). Extensions of Lipschitz mappings into a Hilbert Space. Contemporary mathematics - American Mathematical Society. American Mathematical Society, pp. 189–206,
23. H. Wang, B. Gu, and S. Qu. (2019). A novel hybrid feature selection method for improved high-dimensional data classification. Neurocomputing, vol. 338.

24. PedregosaFabian, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel and Bertrand Thirion,(2012). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research.
25. Alghamdi, A., & Keshta, I. (2026). Blockchain consensus mechanisms and enhancement techniques for federated learning-based intrusion detection systems in IoT smart homes. Journal of Reliable and Secure Computing, 2(1), 1–26.
26. Punitha, P., Dinesh Kumar, V. K., & Lakshmana Kumar, R. (2025). Advancing IoT security with an innovative machine learning paradigm for botnet attack detection. EAI Endorsed Transactions on Internet of Things, 11(1).