An Optimized Multiclass Machine Learning Approach for Detecting Advanced Intrusions in IoT Systems

Mostafa Ibrahim Labib
Department of Computer Science, Higher Future Institute for Specialized Technological Studies, Egypt
mostafa.elkhalil@fa-hists.edu.eg (corresponding author)

Mohamed Salah Mohamed
Department of Computer Science, Faculty of Computer and Information, Suez University, Egypt
mohamed.smah@fci.suezuni.edu.eg

Amira Ibrahim El-Desokey
Department of Basic Sciences, Higher Future Institute for Specialized Technological Studies, Egypt
amira.eldesouky@fa-hists.edu.eg

Fatma Harby Mohamed
Department of Computer Science, Higher Future Institute for Specialized Technological Studies, Egypt | College of Communication Techniques Engineering, Al-Farahidi University, Baghdad, Iraq
fatma.mohamed@fa-hists.edu.eg

Received: 27 October 2025 | Revised: 29 November 2025 | Accepted: 7 December 2025

 DOI: https://doi.org/10.48084/etasr.15800

ABSTRACT

Intrusion Detection Systems (IDS) play a vital role in securing Internet of Things (IoT) networks against cyber-attacks. Previous work used a binary classification approach to detect only Denial-of-Service (DoS) attacks. This paper presents an improved multiclass IDS that identifies multiple attack types. The proposed approach uses simulated IoTID20-based traffic datasets to classify Normal, Denial-of-Service, Mirai botnet, Man-in-the-Middle (MITM), and Scan activity attacks. The proposed approach uses a combination of feature selection methods, including correlation-based filtering and a genetic algorithm, followed by machine learning classifiers such as Decision Tree (DT), Random Forest (RF), K-Nearest Neighbors (KNN), and Support Vector Machine (SVM). Experimental results show a significant improvement in classification performance. The RF classifier with Genetic Algorithm (GA)-based feature selection achieved the highest accuracy (96.5%), followed closely by DT and SVM. Therefore, based on our theoretical and experimental comparisons, the proposed approach could be a practical step toward deploying more robust, realistic IDS models in IoT environments.

Keywords-intrusion detection system; internet of things; denial-of-service attacks; man-in-the-middle attacks; scan activities attacks; decision tree; random forest; k-nearest neighbors; support vector machine

I. INTRODUCTION

The IoT is rapidly expanding into homes, healthcare, industry, and smart cities, but its connectivity also creates new security vulnerabilities. Traditional protection is often insufficient because IoT devices have limited computational power and use a wide variety of communication protocols. An IDS is a security mechanism that monitors network or system activity to detect malicious behavior, policy violations, or unauthorized access. IoT environments pose significant challenges for traditional IDS approaches. Consequently, there is growing interest in intelligent IDS models that deliver real-time, adaptive protection. Recent studies have used machine learning to build intelligent IDSs that analyze network traffic, extract features, and classify activity as benign or malicious [1]. A typical IDS architecture comprises four main components: data collection, feature extraction, detection engine, and decision-making. The data collection module captures network traffic or system logs. The feature extraction module transforms raw data into structured attributes that represent connection behavior, including packet size, duration, rate, and flow statistics. The detection engine then analyzes these features using detection techniques (signature, anomaly,

or hybrid). Finally, the decision-making component classifies activity as either normal or malicious and triggers appropriate responses [1], as shown in Figure 1.

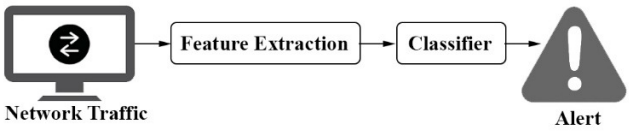

Fig. 1. The block diagram of the IDS system

Machine learning enhances IDS performance, particularly in anomaly detection, by using supervised algorithms trained on labeled network data to detect complex attack patterns.

II. LITERATURE REVIEW

Recent research on IDS for IoT environments examines various feature selection methods and machine learning and deep learning approaches. Authors in [2] introduced a hybrid feature selection method that combines GA with Logistic Regression (LR) to improve IDS performance on the KDD99 and UNSW-NB15 datasets. Their approach selected highly discriminative features, resulting in improved detection rates and reduced false positives, and demonstrated the value of optimization techniques in intrusion detection.

Authors in [3] conducted a comprehensive study of the use of genetic algorithms for feature optimization in IDS. Their work showed that applying genetic feature selection before model training significantly improved performance, particularly in resource-constrained IoT scenarios. In [4], it was found that applying sophisticated feature selection methods, such as Principal Component Analysis (PCA) and Singular Value Decomposition (SVD), significantly improved the discriminative performance of the detection algorithms. Using these advanced feature selection techniques in conjunction with strong machine learning classifiers was essential to improving the detection system's accuracy and reliability. In [5], the authors conducted a comparative performance evaluation of seven machine learning classification algorithms—namely RF, AdaBoost (AB), Gradient Boosted Machine (GBM), Extreme Gradient Boosting (XGB), Extremely Randomized Trees (ETS), Classification and Regression Trees (CART), and Multi-Layer Perceptron (MLP)—and found that the CART and XGB classifiers offer the most favorable balance among key performance metrics and response time. In [6], the authors focused on detecting DoS attacks in IoT-based innovative environments using an SVM classifier. Although the model achieved good results against DoS traffic, it struggled to detect other attack categories and lacked feature selection, limiting its adaptability to real-world scenarios. In [7], the authors assessed the NSL-KDD dataset using an Artificial Neural Network (ANN) and reported satisfactory performance for both intrusion detection and attack-type classification. In [8], a deep learning intrusion detection system based on CNNs was developed to analyze IoT device behavior and detect unauthorized activity with high precision. However, its high computational requirements limit its practicality on low-power IoT devices. In [9], the authors developed a lightweight anomaly detection system that uses flow-based features and statistical analysis to identify Mirai-like botnet activity at low computational cost, but it lacked adaptability to new or unseen attack types. The authors in [10] developed Pulse, an adaptive IDS that dynamically adjusts detection thresholds based on real-time behavioral analysis of IoT devices to detect abnormal IoT traffic. It performs well in live environments but requires ongoing monitoring and tuning to maintain accuracy.

Authors in [11] emphasized the importance of lightweight, multiclass intrusion detection systems for IoT environments to ensure practical deployment, demonstrating that a system implemented on a Raspberry Pi achieves significantly higher detection speed with minimal impact on detection accuracy. In [12], the reliability of the BoT-IoT dataset for forensic purposes was evaluated by comparing a range of statistical and machine learning methods against benchmark datasets. In [13], a simulation-based anomaly detection framework for IoT was proposed, achieving over 99% accuracy in simulations. However, its lack of validation on public IoT datasets raises concerns about reproducibility. In [14], a lightweight IDS was proposed that uses multiple ML classifiers (DT, RF, KNN, SVM) on a binary-transformed version of the IoTID20 dataset. With Correlation-based Feature Selection (CFS) and GA feature selection, the system achieved near-perfect accuracy with low computational overhead.

A CNN-LSTM hybrid model successfully captured temporal network behavior and achieved strong detection accuracy across various attack types, but its high computational demands still make it impractical for lightweight IoT devices [15]. Based on the recent studies discussed in the previous sections, it can be concluded that the reviewed studies primarily focused on IoT network environments, using either a single or multiple ML classifiers to identify threats. Most systems were designed to detect multiple types of attacks, though some targeted specific threats, such as DoS and DDoS. Commonly used datasets included UNSW-NB15, NSL-KDD, and KDD99. Despite their reliability, these are now considered somewhat outdated. Several studies have created custom datasets or used real-time IoT datasets such as IoTID20 and BoT-IoT [14]. Feature selection was frequently used to improve model performance, employing methods such as GA, LA, or RF.

III. METHODOLOGY

The effectiveness of an anomaly-based IDS depends on the quality of its detection engine and feature representation. The system follows a structured pipeline that includes data preprocessing, feature selection, model training, and evaluation, as shown in Figure 2.

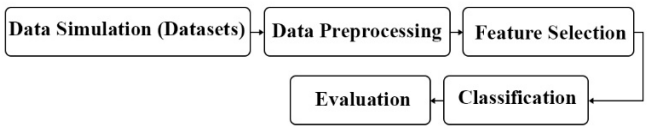

Fig. 2. The proposed approach workflow

A. Dataset and Simulation

This study uses the IoTID20 dataset, which reflects modern IoT network communication by combining IoT devices and their interconnections. It extracts features from PCAP files and outputs them in csv format [16]. The IoTID20 dataset simulates realistic network traffic across five attack categories: normal activity, DoS attacks, Mirai botnet behavior, Man-in-the-Middle (MITM) attacks, and port scanning attacks. It contains 8,000 records and 30 features, of which 20 were selected using feature selection methods.

B. Data Preprocessing

Data preprocessing was performed to clean and normalize the input data using the following steps:

- Removing null values (if present) to ensure model consistency.
- Scaling all feature values to standardize them using StandardScaler to ensure a uniform distribution across dimensions.
- Encoding Class labels to convert them from string format to integer values for classification.
- Excluding noisy or irrelevant features (e.g., flow identifiers such as IP addresses) to avoid bias and reduce generalizability.

C. Feature Selection

Two methods were examined for feature selection, which is used to reduce dimensionality and select the most informative attributes:

1. CFS Selection: The CFS algorithm defines an optimal feature subset as one in which features are strongly correlated with the class but weakly correlated with each other. It is used as a filter method to assess correlations between inputs and outputs. It removes irrelevant or redundant data to reduce computation time and improve detection accuracy. It supports various attribute types, such as binary, date, nominal, empty, and unary values [17].
2. GA Selection: The GA is an evolution-based optimization method that selects optimal feature subsets through iterative generations. It begins by forming a population of feature subsets, evaluating them with a predictive model, and selecting winners through tournaments. New generations are created via crossover and mutation, and after several iterations, the best-performing subset is chosen as the optimal feature set [11].

Both methods were examined, and their results were compared to determine the optimal performance; the 20 most important features were then retained for classification.

D. Classification Algorithms

This study uses multiclass supervised machine learning classifiers (DT, RF, SVM, and KNN). These classifiers were trained and compared to identify the best model, producing class predictions and evaluation metrics. All models used the same training and testing splits and were evaluated under consistent conditions. Hyperparameters were selected empirically through manual grid-based testing and performance comparisons. The operation-based concept of each classifier is shown in Table I.

TABLE I. THE ML CLASSIFIER OPERATION-BASED

Classifier	Operation
DT [6]	Rule-based model offering fast decision paths
RF [2,18]	Ensemble of decision trees with majority voting
SVM [13, 19]	Hyperplane-based classifier for high-dimensional spaces
KNN [20]	Instance-based learning relying on distance metrics

Each model was trained on 70% of the data and tested on the remaining 30%. Stratified cross-validation was used to train each model on the preprocessed dataset. Class imbalance was addressed with sample weighting, and hyperparameters such as tree depth, the SVM kernel, and the number of neighbors were manually tuned.

E. Evaluation

The proposed approach was evaluated using performance measures such as Accuracy (1), Precision (2), Recall (3), and F1 score (4) [4]. A confusion matrix is used to describe a classifier's performance on a set of observations with known actual values. The confusion matrix is computed using these four parameters:

- TP: Both the actual and predicted values are positive.
- TN: Both the actual and predicted values are negative.
- FP: The actual value is negative, but the model predicts it as positive.
- FN: The actual value is positive, but the model predicts it as negative.

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+ TN+FP+FN}} * 100 \quad (1)$$

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

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

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

Accuracy is the proportion of correctly classified cases among all cases. Precision is the proportion of detected intrusions that are correct, with higher values indicating fewer false alarms. Recall indicates the proportion of real intrusions correctly identified. The F1 score combines precision and recall to assess overall performance, balancing false positives and false negatives. After training, the classifiers' predictions are evaluated against the ground-truth labels. Performance metrics are then calculated and visualized, with confusion matrices illustrating the errors for each class.

IV. RESULTS AND DISCUSSION

The primary objective of our experiments is to compare and evaluate the four machine learning classifiers (DT, RF, SVM,

and KNN) after selecting features using the two feature selection techniques described earlier. Each model was tested on the same simulated multiclass IoT traffic dataset with five distinct classes. The experiments were conducted using a simulated dataset modeled after IoTID20 and run in a Python-based environment on standard computing hardware. The following specifications were used: Windows 11, Intel® Core™ i7-12700K CPU @ 3.60GHz, 16 GB of RAM, and Python 3.14. Python libraries such as Scikit-learn, NumPy/Pandas, Matplotlib/Seaborn, Scikit-Genetic/DEAP (optional GA simulation), and Jupyter Notebook were used for visualization and experiments.

A. Decision Tree Results

The DT model was configured using the Gini impurity criterion, with a maximum depth of 8, a minimum of 2 samples per split, and 1 sample per leaf. The depth was limited to 8 to reduce overfitting while preserving an appropriate level of model complexity. It provides a balance between speed and interpretability. It successfully classifies most traffic types, though it struggles to distinguish between similar attack patterns, such as Mirai and Scan, as shown in Figure 3.

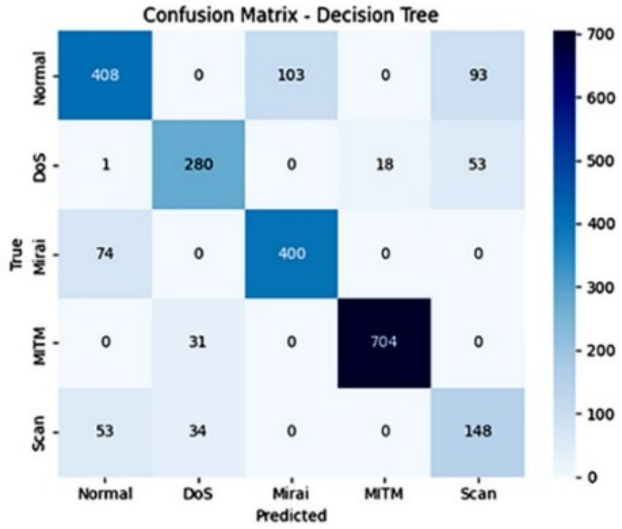

Fig. 3. Confusion matrix of the DT model

B. Random Forest Results

The RF model was configured with 150 trees, a maximum depth of 12, and the Gini criterion for splitting, with bootstrap sampling enabled and a random state of 42. Increasing the number of trees improved the model's stability without adding significant computational overhead. It achieved the highest overall accuracy and F1-score across all classes, showing strong robustness to noise and imbalance, with few misclassifications, mainly in the Scan class, as illustrated in Figure 4.

C. Support Vector Machine Results

The SVM model used an RBF kernel, enabling nonlinear decision boundaries and achieving the best performance after parameter tuning. The final configuration used a regularization parameter (C) of 5, a kernel coefficient (gamma) of 0.05, and balanced class weights. It performs nearly as well as Random Forest, effectively distinguishing Normal, DoS, and MITM traffic but showing some confusion between Scan and Mirai attacks, as shown in Figure 5.

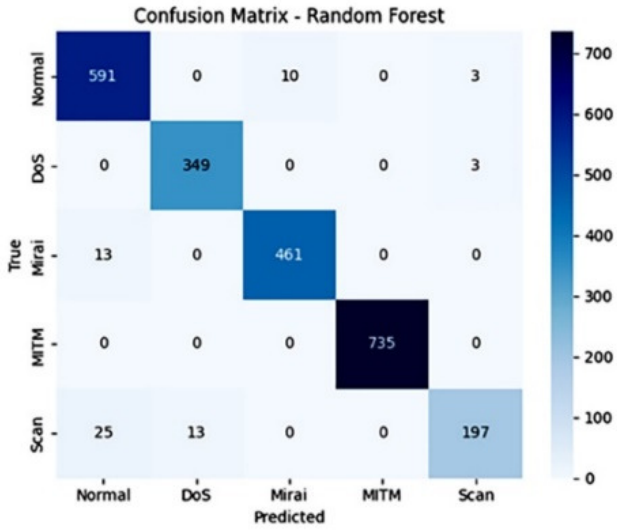

Fig. 4. The confusion matrix of the RF model

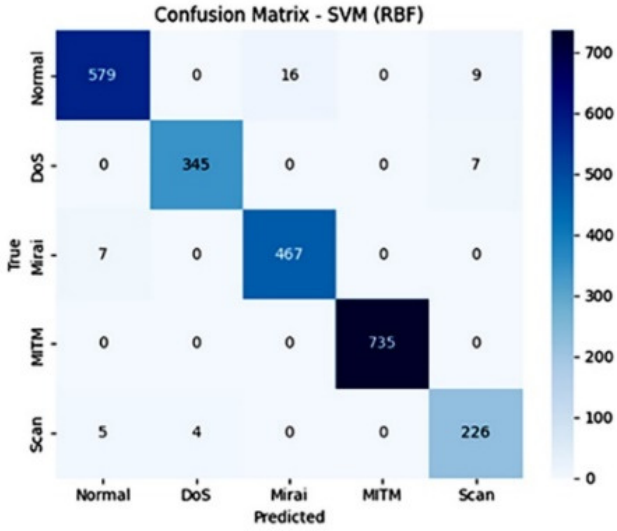

Fig. 5. The confusion matrix of the SVM model

D. K-Nearest Neighbors Results

The KNN model was configured with 5 neighbors, identified during initial tuning as offering the best bias–variance balance, using Euclidean distance as the distance metric and uniform weighting across all neighbors. The KNN model performs well overall but is more affected by class imbalance and overlapping features, leading to lower precision and recall due to its sensitivity to local data patterns, as illustrated in Figure 6.

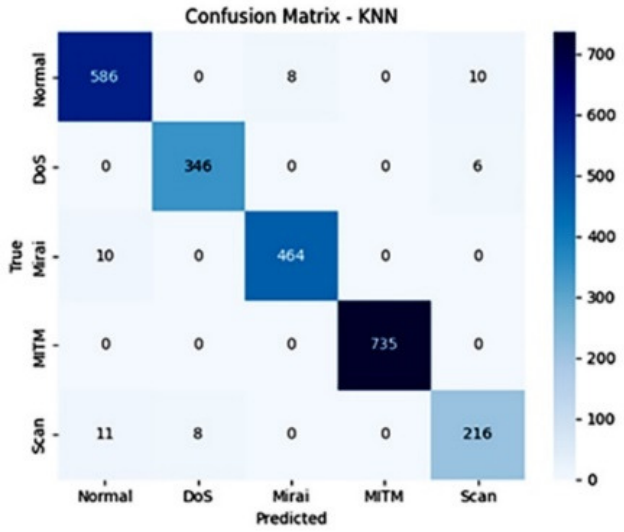

Fig. 6. The confusion matrix of the KNN model

E. Comparative Analysis and Results

The average performance metrics for each classifier model using the top 20 features selected by GA-based feature selection are summarized in Table II. The RF model achieved the highest accuracy, precision, recall, and F1-score among the evaluated models, as illustrated in Figure 7.

TABLE II. CLASSIFIER PERFORMANCE SUMMARY (GA-FEATURE SELECTION)

Classifier	Accuracy	Precision	Recall	F1-Score
DT	0.943	0.944	0.941	0.942
RF	**0.965**	**0.966**	**0.963**	**0.964**
SVM	0.956	0.958	0.951	0.954
KNN	0.928	0.930	0.926	0.927

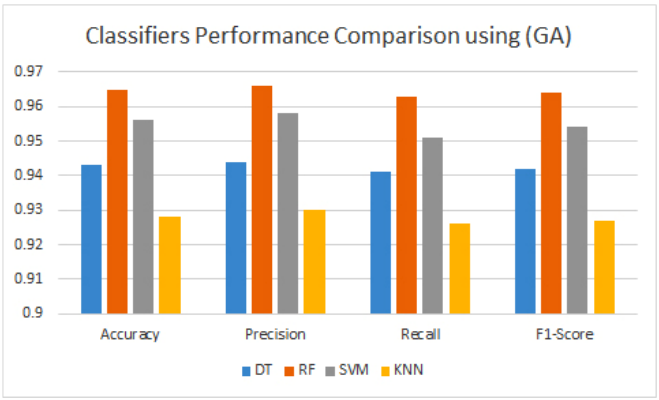

Fig. 7. Classifiers performance comparison using GA

Similarly, the performance metrics for the CFS technique are reported in Table III. The RF model still achieved the highest accuracy, precision, recall, and F1-score among the evaluated models, although its performance was slightly lower than that of the GA–based feature selection, as illustrated in Figure 8. A comparison of CFS and GA-inspired feature selection showed that both techniques improved model performance relative to using the full feature set, with GA-based selection achieving slightly higher classification accuracy, particularly for the Random Forest and SVM classifiers, as shown in Table IV. Overall, GA-based selection yielded a more discriminative feature subset, thereby enhancing the models' ability to detect subtle attack patterns.

TABLE III. CLASSIFIER PERFORMANCE SUMMARY USING CORRELATION-BASED FEATURE SELECTION (CFS)

Classifier	Accuracy	Precision	Recall	F1-Score
DT	0.928	0.929	0.925	0.926
RF	**0.953**	**0.955**	**0.950**	**0.952**
SVM	0.944	0.946	0.941	0.943
KNN	0.914	0.916	0.910	0.912

Based on the confusion matrices for each classifier, RF and SVM achieved high accuracy across all classes, while DT and KNN showed minimal confusion between DoS and Mirai attacks. The Scan class had the lowest precision due to feature overlap with MITM, highlighting the strength of ensemble and kernel-based methods in multiclass detection. The experimental results confirm that traditional ML classifiers are effective for multiclass intrusion detection in IoT environments, with Random Forest offering the best balance of accuracy, interpretability, and speed. The multiclass approach enables precise attack identification, enhancing response actions. Furthermore, lightweight feature selection methods such as CFS and GA reduce complexity and training time, enabling practical edge deployment in smart home and IoT environments.

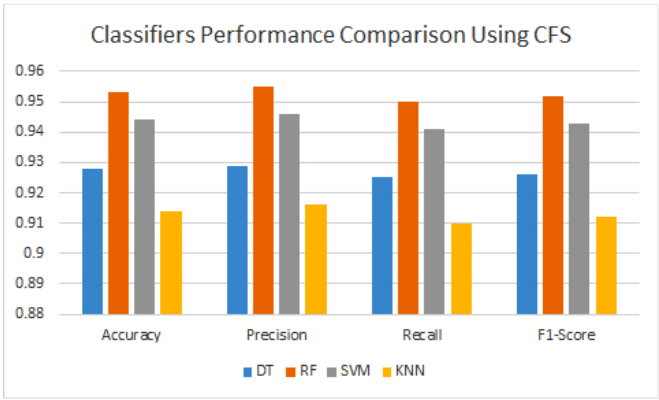

Fig. 8. Classifiers performance comparison using CFS

TABLE IV. PERFORMANCE COMPARISON BETWEEN GA AND CFS USING RF

Classifier	Accuracy	Precision	Recall	F1-Score
GA	0.965	0.966	0.963	0.964
CFS	0.953	0.955	0.950	0.952

V. CONCLUSION AND FUTURE WORK

This study presented a lightweight, multiclass supervised machine learning-based Intrusion Detection System (IDS) for Internet of Things (IoT) networks, expanding the original binary DoS detection model to identify both known and

unknown threats and multiple attack types—DoS, Mirai, MITM, and Scan—along with normal traffic, providing a more comprehensive solution. The proposed approach was tested on a simulated IoTID20-based dataset using four ML algorithms—Decision Tree (DT), Random Forest (RF), Support Vector Machine (SVM), and K-Nearest Neighbors (KNN). Two feature selection techniques, Correlation-based Feature Selection (CFS) and Genetic Algorithm (GA), were used to optimize model performance and reduce feature dimensionality. The experimental results showed that the RF classifier with GA-based feature selection achieved the highest accuracy (96.5%), closely followed by DT and SVM. The confusion matrix analysis confirmed that the approach accurately differentiated all five traffic classes with strong precision and recall, even when class behaviors overlapped. The proposed approach can be used in future work to incorporate real-world IoT datasets, deploy the system on a live IoT testbed to validate its robustness, adapt the system to streaming data to enable real-time intrusion response, and evaluate the model's efficiency on embedded systems (e.g., Raspberry Pi or NVIDIA Jetson). Exploring hybrid models (e.g., stacking or CNN-LSTM) may further improve performance, especially for zero-day attacks.

REFERENCES

[1] N. A. Alsharif, S. Mishra, and M. Alshehri, "IDS in IoT using Machine Learning and Blockchain," *Engineering, Technology & Applied Science Research*, vol. 13, no. 4, pp. 11197–11203, Aug. 2023, https://doi.org/10.48084/etasr.5992.

[2] C. Khammassi and S. Krichen, "A GA-LR wrapper approach for feature selection in network intrusion detection," *Computers & Security*, vol. 70, pp. 255–277, Sept. 2017, https://doi.org/10.1016/j.cose.2017.06.005.

[3] Z. Halim *et al.*, "An effective genetic algorithm-based feature selection method for intrusion detection systems," *Computers & Security*, vol. 110, Nov. 2021, Art. no. 102448, https://doi.org/10.1016/j.cose.2021.102448.

[4] R. A. Al Hasan and E. K. Hamza, "An Improved Intrusion Detection System Using Machine Learning with Singular Value Decomposition and Principal Component Analysis," *International Journal of Intelligent Engineering and Systems*, vol. 16, no. 4, pp. 25–38, Apr. 2023, https://doi.org/10.22266/ijies2023.0831.03.

[5] A. Verma and V. Ranga, "Machine Learning Based Intrusion Detection Systems for IoT Applications," *Wireless Personal Communications*, vol. 111, no. 4, pp. 2287–2310, Apr. 2020, https://doi.org/10.1007/s11277-019-06986-8.

[6] P. Kasinathan, C. Pastrone, M. A. Spirito, and M. Vinkovits, "Denial-of-Service detection in 6LoWPAN based Internet of Things," in *2013 IEEE 9th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, July 2013, pp. 600–607, https://doi.org/10.1109/WiMOB.2013.6673419.

[7] B. Ingre and A. Yadav, "Performance analysis of NSL-KDD dataset using ANN," in *2015 International Conference on Signal Processing and Communication Engineering Systems*, Jan. 2015, pp. 92–96, https://doi.org/10.1109/SPACES.2015.7058223.

[8] Y. Meidan *et al.*, "Detection of Unauthorized IoT Devices Using Machine Learning Techniques." arXiv, Sept. 14, 2017, https://doi.org/10.48550/arXiv.1709.04647.

[9] R. Doshi, N. Apthorpe, and N. Feamster, "Machine Learning DDoS Detection for Consumer Internet of Things Devices," in *2018 IEEE Security and Privacy Workshops (SPW)*, Feb. 2018, pp. 29–35, https://doi.org/10.1109/SPW.2018.00013.

[10] E. Anthi, L. Williams, and P. Burnap, "Pulse: An adaptive intrusion detection for the Internet of Things," in *Living in the Internet of Things: Cybersecurity of the IoT - 2018*, Mar. 2018, pp. 1–4, https://doi.org/10.1049/cp.2018.0035.

[11] Y. N. Soe, Y. Feng, P. I. Santosa, R. Hartanto, and K. Sakurai, "Implementing Lightweight IoT-IDS on Raspberry Pi Using Correlation-Based Feature Selection and Its Performance Evaluation," in *Advanced Information Networking and Applications*, Cham, 2020, pp. 458–469, https://doi.org/10.1007/978-3-030-15032-7_39.

[12] N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull, "Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: Bot-IoT dataset," *Future Generation Computer Systems*, vol. 100, pp. 779–796, Nov. 2019, https://doi.org/10.1016/j.future.2019.05.041.

[13] I. Mukherjee, N. K. Sahu, and S. K. Sahana, "Simulation and Modeling for Anomaly Detection in IoT Network Using Machine Learning," *International Journal of Wireless Information Networks*, vol. 30, no. 2, pp. 173–189, June 2023, https://doi.org/10.1007/s10776-021-00542-7.

[14] E. Altulaihan, M. A. Almaiah, and A. Aljughaiman, "Anomaly Detection IDS for Detecting DoS Attacks in IoT Networks Based on Machine Learning Algorithms," *Sensors*, vol. 24, no. 2, Jan. 2024, Art. no. 713, https://doi.org/10.3390/s24020713.

[15] E. F. Khairullah and N. Alsenani, "A Comprehensive Study of Deep Learning Models for Intrusion Detection in IoT Devices," *Engineering, Technology & Applied Science Research*, vol. 15, no. 2, pp. 21029–21036, Apr. 2025, https://doi.org/10.48084/etasr.9490.

[16] I. Ullah and Q. H. Mahmoud, "A Scheme for Generating a Dataset for Anomalous Activity Detection in IoT Networks," in *Advances in Artificial Intelligence*, Cham, 2020, pp. 508–520, https://doi.org/10.1007/978-3-030-47358-7_52.

[17] I. Alrashdi, A. Alqazzaz, E. Aloufi, R. Alharthi, M. Zohdy, and H. Ming, "AD-IoT: Anomaly Detection of IoT Cyberattacks in Smart City Using Machine Learning," in *2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)*, Jan. 2019, pp. 0305–0310, https://doi.org/10.1109/CCWC.2019.8666450.

[18] M. A. Alsheikh, S. Lin, D. Niyato, and H.-P. Tan, "Machine Learning in Wireless Sensor Networks: Algorithms, Strategies, and Applications," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 4, pp. 1996–2018, 2014, https://doi.org/10.1109/COMST.2014.2320099.

[19] H. Tyagi and R. Kumar, "Attack and Anomaly Detection in IoT Networks Using Supervised Machine Learning Approaches," *Revue d'Intelligence Artificielle*, vol. 35, no. 1, pp. 11–21, Feb. 2021, https://doi.org/10.18280/ria.350102.

[20] G. Thamilarasu and S. Chawla, "Towards Deep-Learning-Driven Intrusion Detection for the Internet of Things," *Sensors*, vol. 19, no. 9, Jan. 2019, Art. no. 1977, https://doi.org/10.3390/s19091977.