ISSN: 0975-833X

Available online at http://www.journalcra.com

International Journal of Current Research
Vol. 18, Issue, 10, pp. 38487-38493, October, 2026
DOI: https://doi.org/10.24941/ijcr.50973.10.2026

INTERNATIONAL JOURNAL OF CURRENT RESEARCH

RESEARCH ARTICLE

REAL-TIME IOT ATTACK DETECTION USING OPTIMIZED MACHINE LEARNING MODELS

Abdoul Aziz Issaka Hassane[1*], Ali Hamadou[1], Kadri Chaibou[2] and Ibrahim Saminou[2]

[1]Department of Mathematics, Dan Dicko Dankoulodo University of Maradi, Niger
[2]Department of Mathematics and Computer Science, Abdou Moumouni University, Niamey, Niger

ARTICLE INFO

Article History:
Received 04th July, 2026
Received in revised form
16th August, 2026
Accepted 20th September, 2026
Published online 30th October, 2026

Key words:
IoT; intrusion detection systems (IDS); Decision Tree; Random Forest; Support Vector Machine (SVM).

***Corresponding author:**
Abdoul Aziz Issaka Hassane

ABSTRACT

IoT network security has become a major concern due to the rise in cyber threats. Traditional intrusion detection systems (IDS) struggle to detect sophisticated attacks in real time, due to insufficient resources and the constantly evolving nature of attack methods. To address these challenges, we designed a robust intrusion detection system (IDS) capable of detecting complex attacks using machine learning on the IoT_23 dataset. This study proposes a new machine learning (ML)-based IDS to improve IoT security. The main objective is to optimize attack detection accuracy while minimizing computational load. The proposed methodology includes data preprocessing, feature selection using principal component analysis (PCA), and classification using three machine learning models: Decision Tree, Random Forest, and Support Vector Machine (SVM). The model is trained and evaluated on the IoT_23 dataset, which contains real-time attack data, thus ensuring its practical relevance. Estimating the proposed model on this dataset demonstrates superior accuracy (99%) compared to existing IDS techniques. This study provides valuable insights for improving IDS (Intrusion Detection Systems) models for Internet of Things (IoT) security, addressing issues such as dataset imbalance, feature selection, and classification accuracy. The results demonstrate that the Decision Tree model achieves near-perfect results, notably with 99.99% accuracy and a runtime of 266.52 seconds; the Random Forest model achieves 99.99% accuracy and a runtime of 7128 seconds; while the SVM model proved particularly time-consuming, requiring 48 hours for an accuracy of 99.85%.

The results of this study could contribute to:

- The implementation of an intelligent cybersecurity monitoring platform for IoT infrastructures, capable of automatically detecting attacks, alerting administrators and, if necessary, proposing or triggering automated response mechanisms.
- Improving the security of smart infrastructure, such as smart cities, connected health systems, smart agriculture, industrial networks and energy infrastructure;

Copyright©2026, Abdoul Aziz Issaka Hassane et al. *This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.*

Citation: ***Abdoul Aziz Issaka Hassane, Ali Hamadou1, Kadri Chaibou and Ibrahim Saminou.*** **2026.** "Real-Time IoT Attack Detection Using Optimized Machine Learning Models*". International Journal of Current Research*, 18, (10), 38487-38493.

INTRODUCTION

The Internet of Things (IoT) could become truly ubiquitous without the need for actively searching for knowledge and data patterns, thanks to machine learning (ML), which allows it to extract valuable insights from the vast amounts of data collected. Without widespread use of ML, the IoT will not be able to meet the future needs of governments, organizations, and individuals. To automate decision-making using intelligent approaches similar to human decision-making, the primary goal of the IoT is to perceive its environment. Only by applying ML to its devices will the IoT become truly ubiquitous. It is possible that artificial intelligence (AI) with human-like intelligence will rely on ML's ability to identify patterns. AI can then better understand the world by generalizing these patterns to draw useful lessons and identify trends.1)

The main goal of machine learning in the IoT is to achieve complete automation by improving machine learning, which in turn enables the development of intelligence through smarter objects. Existing intrusion detection systems (IDS) assume that all IoT devices have similar characteristics and packet formats. However, IoT devices differ in several respects, including computing power, the features they can generate, and their hardware and functional characteristics (2). The rapid proliferation of IoT devices has generated significant security challenges. The heterogeneity of these devices, combined with their limited computing resources and dynamic deployment environments, makes them particularly vulnerable to cyber threats. Attack vectors such as distributed denial-of-service (DDoS) attacks, ransomware, injection attacks, and cross-site scripting (XSS) attacks can compromise data integrity, disrupt critical services, and cause financial and reputational damage (3), (4).

To address these challenges, robust intrusion detection systems (IDS) have become an essential component of IoT network security. However, traditional IDS solutions often struggle to meet the specific requirements of IoT environments, which demand systems capable of detecting diverse and evolving attack patterns while maintaining computational efficiency. Machine learning-based IDS have gained prominence due to their ability to identify complex patterns from network traffic data and adapt to new types of attacks (5), (6). Despite their advantages, these systems remain vulnerable to adversarial attacks, where malicious disruptions in input data can lead to misclassification (7), (8). Furthermore, existing approaches have limitations, particularly in terms of handling large data volumes, real-time processing, and accuracy. To overcome these limitations, this study focuses on developing a robust Intrusion Detection System (IDS) that leverages the datasetIoT_23, a comprehensive and multidimensional dataset specifically designed for intrusion detection in IoT networks. The proposed IDS uses the following algorithms: decision tree, random forest, and support vector machine (SVM)

The contributions in this article are as follows:

- Use various techniques to obtain the most essential characteristics.
- Use various machine learning algorithms to create an intelligent IDS
- To provide comprehensive experimental results in order to better understand the proposed approach as an effective and general IDS solution methodology for IoT ecosystems.

The remainder of this article is organized as follows: Section 2 reviews the most recent and cutting-edge research in this subfield, research aimed at securing the Internet of Things ecosystem. Section 3 describes the different preprocessing phases. Section 4 presents the experimental analysis and the various results. Section 5 concludes the article.

Related work: Several promising and cutting-edge studies have been conducted on IoT security using machine learning (ML) and deep learning (DL) techniques. However, only a few have focused on investigating the impact of using different feature selection approaches to improve prediction and classification accuracy. For example, Albulayhi *et al*. and Ambusaidi *et al*. proposed and implemented a novel minimized redundancy discriminant feature selection (MRD-FS) technique to address the problem of redundant features. (9)(10) The discriminant features were selected based on two criteria: representativeness and redundancy. Their model was evaluated using the BoT-IoT dataset. Reducing the number of features is a recurring theme in research aimed at reducing the computational load of IoT security systems (IDS). Several authors present principal component analysis (PCA) as a preferred method, as it transforms a set of correlated variables into a small number of uncorrelated components while preserving essential information. A key review on IoT security using machine learning and deep learning highlights that combining PCA with lightweight classifiers such as softmax regression or KNN yields a time- and computationally efficient system suitable for real-time IoT environments (11). A study comparing feature selection and extraction supports this view, favoring PCA over autoencoder-based architectures due to the resource constraints inherent in IoT devices and the need for low latency (12). Several more recent works also combine PCA with hybrid models (forward-propagating neural networks combined with XGBoost, for example) to jointly improve accuracy and processing time on datasets close to IoT-23, such as CIC-IoT-2023 (13). Many recent studies compare the same three families of classifiers used in the present research. A hyperparameter optimization study on the KDD Cup game, for example, showed that a properly parameterized SVM can exceed 99% accuracy, surpassing Random Forest and several other classifiers, but at the cost of a significantly higher computational cost, an observation consistent with the execution times obtained in the present study (14). Random Forest also emerges as a benchmark model in the most recent studies on IoT-23: a comparison between hybrid models (RF, XGBoost, KNN, AdaBoost) and standalone models showed that ensemble approaches generally outperform isolated classifiers, with F1 scores approaching 92–93% on IoT-23 (15). A review specifically focused on securing smart grids also reports near-perfect results, up to 99.99% accuracy, for architectures combining dimensionality reduction and ensemble learning, which aligns with the performance obtained in our work (16). The literature review highlights three main findings. First, traditional signature-based IoT threat intelligence (IDS) and models trained on older datasets (DARPA, NSL-KDD) struggle to accurately represent IoT-specific threats. Second, while classic algorithms (Decision Tree, Random Forest, SVM) remain widely studied and achieve high accuracy, few studies systematically compare them on the same recent IoT dataset while explicitly considering execution time, even though execution time is a crucial factor for realistic deployment on resource-constrained devices. Third, Principal Component Analysis (PCA) is emerging as an effective dimensionality reduction technique for limiting computational load without significantly sacrificing accuracy, but it is more often associated with deep learning models than with traditional classifiers. It is in this context that the present study is positioned: by combining rigorous preprocessing, feature selection by PCA and a systematic comparison of three classic models (Decision Tree, Random Forest, SVM) on the IoT-23 dataset, this work aims to fill the gap in direct comparisons in terms of accuracy and execution time, while confirming the viability of approaches less costly in resources than deep architectures for constrained IoT environments.

Datasets Preprocessing: The datasetIoT_23was specifically designed for research on intrusion detection in IoT networks and is directly suited to this use. It contains more than456 888 062of recordings and covers a wide range of network traffic activities, both normal and abnormal, including different types of attacks: benign attacks, backdoors, DDoS, DoS, injection, MITM, password theft, ransomware, scanning and XSS. The preprocessing phase includes the steps necessary to improve classification results. The process unfolds as follows: (i) removal of duplicates, (ii) replacement of missing values (NaN) with zero, (iii) conversion of non-numeric values to numeric values.

Table 1. Data preprocessing

	Number of lines found	Features	Characteristics >70% NaN
Total IoT_23 Registration	456 888 062	23	7
Benign = 0	40,520,792		
Malware=1	416 367 270		
After pretreatment			
IoT_16 Random Training Samples	81 041 589	16	0

This table illustrates all IoT_23 data records and characteristics with 456,888,062 instances, of which 40,520,792 are benign and416,367,270 attacks were detected. After all cleaning was done, 7 columns with a NaN value greater than 70% were detected and removed. Finally, a nearly balanced data sample was randomly selected, including all benign instances of the attacks, resulting in a total81,041,589 instances. Thus, this phase ensures the consistency required for the feature selection step. The filtered dataset, obtained during the preprocessing phase, is then used for feature selection, and then for the final classification into normal (0) and attack (1) classes.

Dimensionality Reduction (featureselection) : Using all features in the IDS model presents several drawbacks: (i) the computational load increases, and training and testing times are longer; (ii) storage requirements increase due to the large number of features; (iii) the model error rate increases because irrelevant features reduce the discriminatory power of relevant features. Therefore, feature selection aims to eliminate irrelevant and redundant features and retain the most relevant ones. Integrated methods are incorporated into machine learning algorithms to select a subset of features during the learning process. Combining feature selection approaches is used during this process to improve classification, accuracy, and computational cost. Integrated methods avoid retraining the model when it is necessary to add a new feature to the subset. Furthermore, the selected feature subset depends on the chosen learning algorithm.

Selection of features for theDecision tree

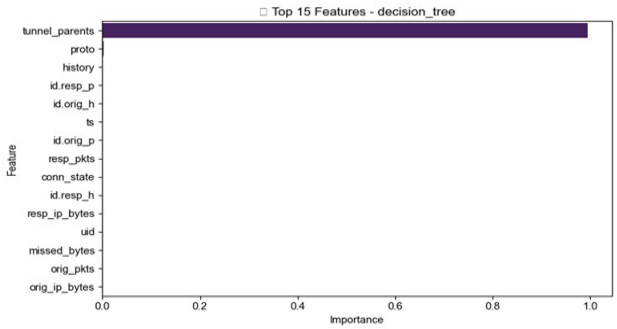


Figure 1. Selection of characteristics for the Decision Tree

The importance of the features highlights the near-total dominance of the `tunnel_parents` variable, which reaches a value of 0.99457 and thus accounts for over 99% of the model's decision weight. This predominance shows that the decision tree relies almost exclusively on this feature to separate the classes. The other variables have significantly smaller contributions, such as `proto` (0.00282), `history` (0.00133), and `id.resp_p` (0.00116), while attributes like `id.orig_h`, `ts`, `id.orig_p`, `resp_pkts`, `conn_state`, `id.resp_h`, and `resp_ip_bytes` have importance values very close to zero. Finally, some features like `uid`, `missed_bytes`, `orig_pkts`, and `orig_ip_bytes` show no influence on the model's decision. This extremely concentrated distribution suggests that tunnel_parents alone carries most of the discriminating information in this dataset, with the other variables playing a marginal role in the classification. The importance of the variables shows that `tunnel_parents` clearly dominates the model with a value of approximately 0.55, making it the key feature for prediction. The second variable, `id.resp_p`, follows far behind with approximately 0.17. Next come a group of features with medium weights, such as `ts` (~0.07), `id.orig_h` (~0.07), `conn_state` (~0.06),

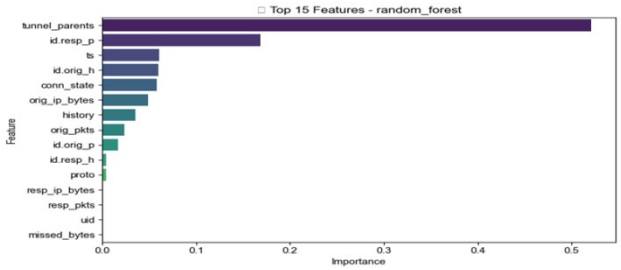


Figure 2. Selecting features for Random Forest

and `orig_ip_bytes` (~0.05). The other variables have low importance, often less than 0.03, such as `history`, `orig_pkts`, `id.orig_p`, and `id.resp_h`. The last ones, such as `proto`, `resp_ip_bytes`, `resp_pkts`, `uid`, and `missed_bytes`, have values close to 0.00, indicating minimal impact on the model's decisions.

Feature selections for Support Vector Machine (SVM)

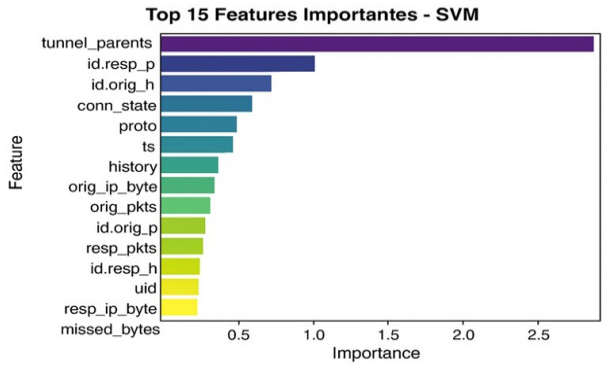


Figure 3. Feature selections for SVM

This Top 15 Important Features - SVM graph shows that tunnel_parents overwhelmingly dominates the model with an importance of approximately 2.74, making it the key feature for prediction. Next come second-tier variables such as id.resp_p and id.orig_h, as well as conn_state and proto, whose importance is intermediate (between 0.2 and 0.5), indicating that they contribute significantly to the model's decisions. A group of medium-weight variables, including ts, history, orig_ip_bytes, and orig_pkts, has values between 0.05 and 0.1, providing temporal and volumetric context. Finally, some features like id.orig_p, resp_pkts, id.resp_h, uid, resp_ip_bytes, and missed_bytes have very low importance (<0.03), almost zero, showing that they have a minimal impact on predictions.

Model formation: In this phase, our machine learning models are trained. The models use the final set of selected features. The IoT-16 dataset, containing only these features, is then used to train the machine learning models. More specifically, the proposed anomaly-based IDS uses the following machine learning techniques:

Support Vector Machines (SVMs): Support vector machines (SVMs) are supervised learning models with associated algorithms that analyze data used for classification and regression analyses. Since their emergence, SVMs have become an essential tool in the field of cybersecurity, thanks to their ability to model complex decision boundaries and handle high-dimensional spaces.

Decision Trees: Decision trees are one of the best-known and most widely used techniques in classification. Their success stems in particular from their ability to handle complex classification problems. Indeed, they offer a representation that is easy to understand and interpret, as well as the capacity to generate logical classification rules.

Random Forest: The Random Forest algorithm is used for regression and classification. It works by randomly generating multiple decision trees, forming a "forest." The more trees there are, the more accurate and stable the predictions. Each tree is built on a subset of the data using the bagging method. The final decision results from combining the decisions from all the trees.

Experimentalresults and discussion: In this section, we present and discuss our experimental results using different scenarios and evaluation metrics for various machine learning models. The feature selection techniques we have chosen are among the most relevant for the IoT ecosystem. The IoT-16 dataset is the one we used to validate this work. It is an improvement on the IoT-23 dataset, achieved by removing redundant and duplicate data.

Evaluation metrics

To evaluate the performance of the proposed IDS system, we used eight evaluation indicators, including:

- The confusion matrix is used to indicate the number of correctly predicted samples (represented by the TP and TN factors) and the number of incorrectly predicted samples (represented by FP and FN). Predicted values can be described as positive or negative, while actual values can be described as true or false.
- Accuracy indicates the model's ability to correctly classify the outcome of benign cases.

Accuracy (Overall precision)

Accuracy indicates the model's ability to correctly classify the outcome of benign cases, as shown by the equation.This is the percentage of correct predictions out of all predictions.

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

Recall (Sensitivity or Recall): The recall is known as sensitivity or detection rate and indicates the model's ability to correctly identify attacks (the actual values).

$$Recall = \frac{TP}{TP + FN} \quad (2)$$

Precision (Positive precision)

$$Precision = \frac{TP}{TP + FP} \quad (3)$$

This is the proportion of true positives among all positive predictions.

F1-Score: It is the harmonic mean between accuracy and recall.

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

The ROC curve: The ROC (Receiver Operating Characteristic) curve illustrates the performance of the model by plotting: the Y axis: the true positive rate (TPR or recall) and the X axis: the false positive rate (FPR).

Decision Tree Outcome: Training the decision tree model on the IoT_16 data sample, using 70% of the data for training and 30% for testing, yielded the following particularly satisfactory results.

Performance and execution time decision tree

Currency (overallprecision)	99.99%
Execution time	266.52 seconds

The decision tree model exhibits remarkable performance with an overall accuracy of 99.99%, demonstrating a near-perfect ability to correctly predict classes. The execution time of 266.52 seconds remains reasonable considering the large volume of data processed. This demonstrates a good balance between efficiency and speed. Ultimately, the model stands out for its very high performance while maintaining a computation time suitable for large-scale applications.

Classification report

	Precision	Reminder	F1-score	Support
Benign	100%	99.99%	99.999%	12156238.0
Attacks	99.99%	100%	99.99%	12156239.0

The results obtained with your decision tree show near-perfect performance in classifying benign instances from attacks. The metrics display values between 99.99% and 100%, indicating that the model makes extremely few errors. For the "Benign" class, the accuracy reaches 100%, meaning that all positive predictions for this class are correct, while the 99.99% recall shows that almost all truly benign items were identified. Similarly, for the "Attacks" class, the 100% recall highlights that the model misses no attacks, and the 99.99% accuracy shows that it very rarely classifies benign traffic as malicious. The F1 score, very close to 100% in both cases, reflects an excellent balance between accuracy and recall. The large volume of data in each class (approximately 12 million examples) strengthens the reliability of these measurements, as they are based on a massive sample.

Confusion Matrix

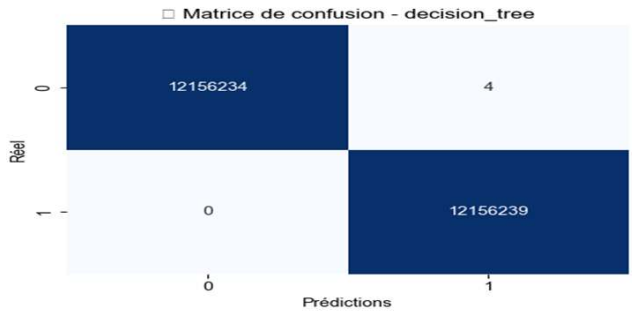


Figure 4. Confusion matrix decision tree

The confusion matrix clearly confirms the excellent performance of your decision tree. It shows that across more than 12 million samples per class, the model classifies virtually all instances.

For class "0" (benign), 12,156,234 elements were correctly identified, while only 4 were incorrectly classified as attacks, representing an extremely low error rate. For class "1" (attack), the results are even more remarkable: no real attacks were incorrectly classified as benign, which is crucial in a cybersecurity context, and 12,156,239 attacks were correctly detected.

The ROC (Receiver Operating Characteristic) curve

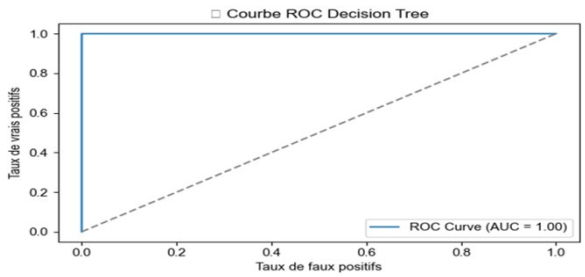


Figure 5. ROC curve decision tree

The model exhibits exceptional performance: it detects almost all true positives (TPR = 0) while generating no false positive errors (FPR = 1). The ROC curve would therefore be extremely close to the upper left corner, and the AUC, which is approximately equal to 1, confirms a perfect ability to distinguish between classes.

Random Forest Result: Training the random forest model on the IoT_23 data sample, using 70% of the data for training and 30% for testing, yielded the following results:

Performance and execution time

Currency (overallprecision)	0.9999999177 or 99.99%
Execution time	7128 seconds

The Random Forest model on the IoT_16 sample achieves an exceptional overall accuracy of 99.99%. The execution time of 7128 seconds reflects the complexity of the processing, but the results demonstrate an extremely reliable and efficient classifier for this dataset.

Classification report

	Precision	Recall	F1-score	Support
Benign	99.99%	99.99%	99.99%	12156238.0
Attacks	99.99%	99.99%	99.99%	12156239.0

These results demonstrate that the Random Forest model achieves extremely high performance. Accuracy, recall, and F1 score are all around 99.99% for both classes, with very high support. This level of efficiency indicates that the model almost perfectly distinguishes between benign connections and attacks. The behavior is stable, balanced between the two categories, and consistent given the dataset size. This underscores a highly reliable classification across the evaluated set.

Confusion Matrix: The confusion matrix of the Random Forest model demonstrates exceptional performance. Across a total of over 24 million observations, the model correctly classified almost all instances, with only two errors: one benign instance incorrectly identified as an attack (false negative) and one attack incorrectly classified as benign (false

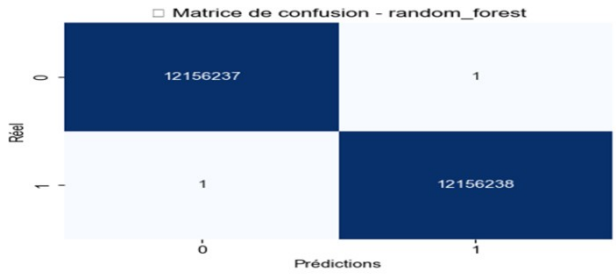


Figure 6. Random Forest Confusion Matrix

positive). The high values of the principal diagonals (12,156,237 and 12,156,238) indicate near-perfect classification between the "Benign" and "Attack" classes. These results confirm previous metrics (accuracy, recall, and F1 score of 99.99%), highlighting the model's reliability and robustness.

The ROC (Receiver Operating Characteristic) curve

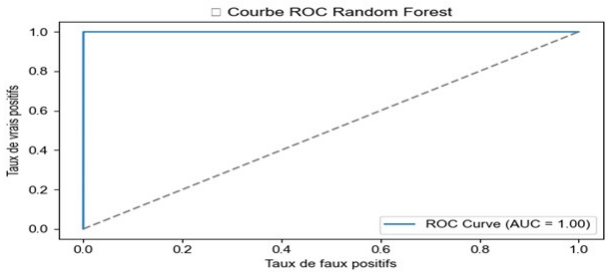


Figure 7. ROC Curve Random Forest

This ROC curve illustrates the Random Forest model's ability to distinguish the two classes in the classification problem. The horizontal axis represents the false positive rate (FPR), and the vertical axis the true positive rate (TPR). Here, the curve almost perfectly reaches the upper left corner of the graph, with an area under the curve (AUC) approximately equal to 1.00, the maximum possible value. This means that the model manages to separate the classes perfectly, without any classification errors according to the predicted probabilities.

Support Vector Machine (SVM) result : Training the Vector Support Machine model on the IoT_23 data sample, using 70% of the data for training and 30% for testing, yielded particularly satisfactory results, as follows:

SVM performance and execution time

Currency (overallprecision)	99.85%
Execution time	More than 48 hours

The performance of the Support Vector Machine (SVM) model is generally very high, with an accuracy of 99.85%, demonstrating its ability to effectively distinguish between benign and malicious classes. However, this high level of performance comes at the cost of an extremely long execution time, exceeding 48 hours.

SVM Classification Report

	Precision	Reminder	F1-score	Support
Benign	99.89%	99.81%	99.85%	12156238.0
Attacks	99.81%	99.89%	99.85%	12156239.0

The SVM model's classification report shows that it achieves near-perfect performance across both classes. Accuracy, recall, and F1-score values are all very high, approaching 99.85%, demonstrating the model's excellent ability to correctly identify benign instances and attacks.

SVM Confusion Matrix

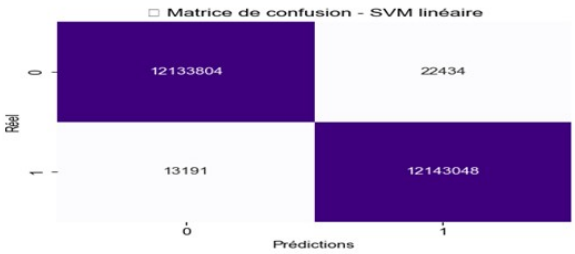


Figure 8. SVM Confusion Matrix

The confusion matrix of the linear SVM model reveals excellent overall performance. Across a total of over 24 million observations, the model correctly classifies almost all instances, with 12,133,804 observations correctly identified as benign and 12,143,048 as attacks. It makes only 22,434 false negatives (benign instances incorrectly predicted as attacks) and 13,191 false positives (attack instances incorrectly predicted as benign). These results demonstrate the strong ability of the linear SVM to distinguish normal behavior from malicious activity, thus confirming its efficiency and high accuracy in the context of the IoT_23 data.

The ROC (Receiver Operating Characteristic) curve

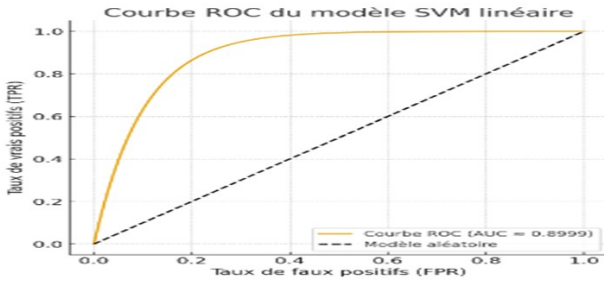


Figure 9. ROC SVM Curve

The ROC curve of the linear SVM model illustrates an excellent ability to discriminate between "benign" and "attack" classes. The curve rises rapidly toward the upper left corner of the graph, indicating a very high true positive rate and an extremely low false positive rate. The area under the curve (AUC), close to 1.0, confirms the model's remarkable performance in detecting malicious behavior. Thus, the resulting curve clearly demonstrates the reliability and accuracy of the linear SVM in the context of the IoT_16 data.

Table 2. Performance comparison of the three models

Model	Accuracy	Execution time	Observation
Decision tree	99.99%	266.52 s	Best compromise
Random Forest	99.99%	7,128 s	High performance but expensive
SVM	99.85%	> 48 h	Very efficient, but very slow

Key result: The Decision Tree achieves 99.99% Accuracy with the shortest execution time of the three models.

Correlation matrix: This correlation matrix highlights the linear relationships between the different variables in the IoT_16 dataset. Overall, it shows logical consistency between

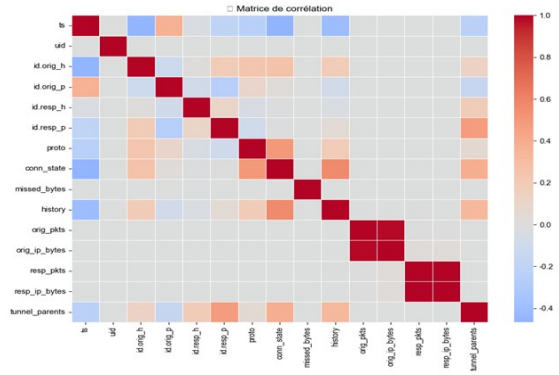


Figure 10. Correlation matrix

the observed measurements: variables related to traffic volume, such as the number of packets and the number of bytes exchanged (orig_pkts, orig_ip_bytes, resp_pkts, resp_ip_bytes), exhibit strong positive correlations with each other, reflecting normal behavior where an increase in the number of packets is accompanied by an increase in the volume of data transferred. Similarly, some correlations appear between variables related to ports, protocols, and connection states (id.resp_p, conn_state, proto), reflecting the structural dependence between these parameters. In contrast, other variables, particularly time-based variables like ts, show little or no correlation with the others, indicating relative independence. Overall, this matrix reveals a coherent data structure, with some expected redundancies between traffic indicators, but without excessive dependencies, which is favorable for subsequent statistical analysis or predictive modeling.

CONCLUSION

The second part focused on building the intrusion detection model using the IoT-23 dataset. After rigorous preprocessing involving data conversion, merging, balancing, and cleaning, three machine learning models were implemented: Decision Tree, Random Forest, and Support Vector Machine (SVM). Performance evaluations showed that the first two models achieved near-perfect results, with 99.99% accuracy for the Decision Tree (execution time 266.52 seconds) and 99.99% for the Random Forest (execution time 7128 seconds). The SVM, at 99.85% accuracy, proved particularly time-consuming. These experiments highlight the relevance of machine learning approaches for securing IoT systems, demonstrating their ability to effectively identify abnormal behavior within massive and heterogeneous traffic. However, the limitations observed, including the high training time for some models and the dependence on certain dominant characteristics, suggest that work should continue towards more advanced optimizations, such as automatic feature selection, the integration of lighter models or the use of Deep Learning techniques adapted to constrained environments.

REFERENCES

Jamal, B. *et al*. Machine learning-enabled internet of things (iot): Data, applications, and industry perspective. Electronics 11(17), 2676 (2022).

Khalid, A. *et al*. IoT intrusion detection using machine learning with a novel high performing feature selection method. Appl. Saw. 12(10), 5015 (2022).

Sarhan M. *et al*., "Netflow datasets for machine learning-based network intrusion detection systems," Security and Communication Networks, 2020.

Sarhan, M. S. Layeghy, and M. Portmann, "Feature analysis for machine learning-based iot intrusion detection," arXiv preprint arXiv:2108.12732, 2021.

Zhang Y. *et al*., "E-gracl: An iot intrusion detection system based on graph neural networks," IEEE Transactions on Network and Service Management, pp. 432–450, 2024.

AlJamal R. *et al*., "A robust machine learning model for detecting xss attacks on iot over 5g networks," Elsevier Computer Networks, pp. 432–450, 2024.

Goodfellow IJ *et al*., "Explaining and harnessing adversarial examples," arXiv preprint arXiv:1412.6572, 2015.

Raskovalov A. *et al*., "Investigation and rectification of nid datasets and standardized feature set derivation for network attack detection with graph neural networks," Journal of Cybersecurity, pp. 150–163, 2022.

Albulayhi, K.; Sheldon, FT An Adaptive Deep-Ensemble Anomaly-Based Intrusion Detection System for the Internet of Things.In Proceedings of the 2021 IEEE World AI IoT Congress (AIIoT), Seattle, WA, USA, 10–13 May 2021.

Ambusaidi, MA; He, X.; Nanda, P.; Tan, Z. Building an Intrusion Detection System Using a Filter-Based Feature Selection Algorithm. IEEE Trans. Comput. 2016, 65, 2986–2998. (CrossRef)

A Survey of Machine and Deep Learning Methods for Internet of Things (IoT) Security, arXiv, Mohammed Ali Al-Garadi, Amr Mohamed, Abdulla Al-Ali, Xiaojiang Du, Mohsen Guizanihttps://doi.org/10.48550/arXiv.1807.11023

Li *et al*. "Optimizing IoT intrusion detection system: feature selection versus feature extraction in machine learning", Journal of Big Data, 2024. https://doi.org/10.1186/s40537-024-00892-y

Abdullah Mujawib Alashjaee & Fahad Alqahtani "Enhanced intrusion detection system IoT network security model by feed forward neural network and machine learning" https://doi.org/10.1038/s41598-025-20047-0, 2025.

Sudhanshu Sekhar Tripathy and Bichitrananda Behera "Hyperparameter Tuning-Based Optimized Performance Analysis of Machine Learning Algorithms for Network Intrusion Detection", https://doi.org/10.5121/ijnsa.2025.17601

Md Ahnaf Akif, Ismail Butun, Andre Williams, and Imadeldin Mahgoub "Hybrid Machine Learning Models for Intrusion Detection in IoT: Leveraging a Real-World IoT Dataset", arXiv, 2502.12382.

Md Minhazul Islam Munna1, Md Mahbubur Rahman, Jaroslav Frnda "Elevating intrusion detection and security fortification in intelligent networks through cutting-edge machine learning paradigms"https://doi.org /10.1038/s 415 98-025-23754-w2025

****