From Risk to Protection: Smart Threat Detection in IoT Using Machine Learning

RAFAEL SEBASTIÃO GALDINO[1]
VITOR LUCAS SILVA SANTOS[1]
ANDRE DE LIMA SALGADO[1]
RENATA LOPES ROSA[1]
DEMOSTENES Z. RODRIGUEZ[1]

[1] Federal University of Lavras, Lavras, Brazil
[1]rafael.galdino@estudante.ufla.br
[1]vitorlucassilvasantos@gmail.com
[1]andre.salgado@ufla.br
[1]renata.rosa@ufla.br
[1]demostenes.zegarra@ufla.br

Abstract. The proliferation of Internet of Things (IoT) devices has expanded the attack surface of modern networks, demanding intrusion detection systems (IDS) that combine high detection capability with low computational cost. This work develops and evaluates an IDS for IoT-oriented environments based on classical machine learning algorithms. The NSL-KDD dataset is used to build a preprocessing pipeline with normalization and one-hot encoding, followed by the training of supervised models (Decision Tree and Random Forest) and unsupervised models (K-means and DBSCAN). The supervised models are assessed using accuracy and weighted F1-score, while the clustering methods are evaluated with Adjusted Rand Index, Normalized Mutual Information, and silhouette coefficient. Experimental results show that both supervised models achieve near-perfect classification performance, with weighted F1-scores around 0.997, with Random Forest slightly outperforming Decision Tree, especially on rare attack classes. The unsupervised methods capture part of the data structure but exhibit only moderate agreement with the normal/attack separation, indicating a complementary role in anomaly detection. The best supervised model (Random Forest) is deployed and validated on a Raspberry Pi Desktop, where it maintains high accuracy, mean inference latency on the order of 10^{-2} ms per sample, and moderate resource usage, demonstrating the feasibility of running IDS functionalities on edge devices in IoT scenarios.

Keywords: machine learning, smart threat detection, IoT attacks.

(Received November 10th, 2025 / Accepted November 25th, 2025)

1 Introduction

The rapid evolution of the Internet of Things (IoT) has transformed domestic, industrial, and urban environments into interconnected intelligent ecosystems capable of supporting advanced services in smart grids, healthcare, intelligent transportation, multimedia systems, and public safety. When combined with 5G and emerging 6G technologies, IoT enables ultra-reliable, low-latency communication and massive device connectivity, supporting automation, resilience, and intelligent decision-making in critical infrastructures [2, 4, 18]. These advances have strengthened essential domains such as smart energy management, traffic automation, human behavior monitoring, multimedia ex-

perience optimization, and digital service quality evaluation [7, 8, 20, 22, 27].

However, these benefits come with significant cybersecurity risks. The exponential proliferation of IoT devices has dramatically expanded the network attack surface, exposing infrastructures to sophisticated cyber threats such as distributed denial-of-service (DDoS) attacks, botnet recruitment, unauthorized device hijacking, firmware exploitation, and data manipulation [13, 14, 18, 24]. Many IoT devices operate under severe computational restrictions and often lack native security mechanisms, standardized encryption, and timely firmware maintenance, making them highly attractive targets for cybercriminals. Moreover, IoT ecosystems are highly heterogeneous and dynamic, integrating mobile, embedded, sensor-based, and industrial devices that continuously exchange data across different network layers [2, 12, 25]. This complexity complicates threat detection, anomaly identification, and resilience assurance.

Intrusion Detection Systems (IDS) play a crucial role in mitigating cyber risks by identifying malicious activities and preventing breaches before they compromise system integrity, availability, or user privacy. However, traditional signature-based IDS struggle to detect emerging or unknown attacks and suffer under highly dynamic IoT traffic behaviors. Anomaly-based IDS offer better adaptability, but their effectiveness depends on intelligent learning capabilities and computational feasibility [5, 14, 19]. Additionally, IoT security must address deployment constraints imposed by edge environments, where latency, processing overhead, and power consumption need to be carefully balanced [6, 24, 26].

In this context, Machine Learning (ML) has emerged as a powerful enabler for intelligent intrusion detection. Supervised ML techniques offer robust classification capabilities, while unsupervised and deep learning approaches facilitate anomaly detection and zero-day threat identification. ML has already demonstrated success across several real-world intelligent systems, including traffic monitoring, healthcare analytics, multimedia quality estimation, renewable energy systems, and behavioral modeling [2, 8, 17, 21, 23]. Furthermore, AI-driven frameworks enable context awareness, predictive network intelligence, and proactive defense strategies, improving resilience of IoT infrastructures [4, 6, 12, 13, 18, 19]. Nevertheless, challenges persist, including dataset imbalance, model generalization, computational overhead [15, 28, 29], and real-time execution feasibility in constrained devices.

Motivated by these challenges, this work proposes the development and evaluation of a machine learning-based Intrusion Detection System tailored for IoT environments. The primary objective is to analyze the trade-off between detection accuracy and computational performance while supporting execution in edge-computing environments. The study explores preprocessing, feature engineering, model selection, energy/resource efficiency, and practical deployment feasibility using a Raspberry Pi platform, advancing secure and resilient IoT-driven infrastructures.

The main contributions of this work include: (i) the design and implementation of a lightweight ML-based IDS framework for IoT networks; (ii) comparative evaluation of supervised and unsupervised ML techniques using the NSL-KDD dataset; (iii) performance assessment considering latency, execution cost, and memory constraints; and (iv) validation of real deployment feasibility in an edge environment.

The remainder of this paper is organized as follows. Section 2 presents the theoretical background and related work. Section 3 explains the methodology. Section 4 presents the experimental evaluation and edge validation. Section 5 discusses results. Section 6 addresses threats to validity. Section 7 concludes the paper and highlights future directions.

2 Theoretical Background and Related Work

The Internet of Things (IoT) is composed of a vast ecosystem of interconnected devices—such as sensors, actuators, cameras, household appliances, wearable gadgets, and industrial controllers—capable of continuously capturing, processing, and exchanging data. This pervasive connectivity enhances automation and decision-making efficiency across domains such as smart cities, smart homes, healthcare, multimedia systems, and Industry 4.0 environments [2, 22, 27]. However, IoT connectivity also significantly increases the network attack surface, since each device may serve as a potential entry point for malicious activities. Common threats include DDoS attacks, firmware exploitation, device hijacking, privacy breaches, and botnet formation [13, 14, 18], demanding robust security architectures and proactive detection mechanisms.

2.1 Intrusion Detection Systems in IoT

Intrusion Detection Systems (IDS) aim to identify suspicious or malicious activity in networks and hosts, complementing mechanisms such as encryption, firewalls, authentication controls, and access management policies. IDS approaches are typically classified as signature-based or anomaly-based. Signature-based

IDS compare network traffic with known attack patterns, achieving high performance for previously cataloged threats but struggling against evolving cyberattacks. Conversely, anomaly-based IDS construct models of normal system behavior and identify deviations, which is particularly suitable for IoT environments characterized by heterogeneity, traffic unpredictability, and continuous evolution [14, 19, 24].

IDS design in IoT settings introduces additional constraints such as limited memory, restricted processing capability, reduced battery capacity, protocol heterogeneity, latency constraints, and the need to avoid cascading failures. Therefore, modern research emphasizes lightweight, scalable, and edge-capable solutions capable of reducing dependency on centralized cloud processing while ensuring resilience and rapid response [9, 10, 16]. Edge computing combined with intelligent analytics has demonstrated strong potential to support local detection while maintaining real-time performance [6, 12, 24].

2.2 Machine Learning Applied to IoT-Based IDS

Machine Learning (ML) techniques have emerged as powerful enablers for intrusion detection in IoT ecosystems. Supervised learning techniques have proven highly efficient in classifying malicious traffic, while unsupervised and semi-supervised approaches enable anomaly detection and zero-day threat identification [2, 13,18]. Decision Tree and Random Forest models stand out due to their interpretability, robustness, and moderate computational cost, which is advantageous for IoT [16]. Meanwhile, unsupervised approaches such as K-means and DBSCAN help identify novel attack types and rare traffic behaviors.

More recently, deep learning architectures—including recurrent networks, convolutional models, autoencoders, and federated learning approaches—have strengthened distributed intelligence and collaborative defense capabilities in IoT, transportation systems, multimedia networks, and healthcare [1, 3, 8, 14, 19]. While deep models may increase detection accuracy, they also introduce execution costs and energy overhead challenges, requiring careful balance to ensure adoption in constrained edge environments [6, 24, 26].

2.3 Datasets and the Role of NSL-KDD

The evaluation of ML-based IDS solutions depends on representative datasets. Among widely used benchmarks are KDD Cup 1999, NSL-KDD, CICIDS datasets, and emerging IoT-specific datasets. NSL-KDD evolved to overcome redundancy and imbalance limitations of its predecessor while providing structured attack labeling, supporting reliable comparative experimentation [16]. Although it does not fully represent modern IoT traffic conditions, it remains highly relevant due to scalability compatibility with constrained hardware and suitability for algorithmic benchmarking.

2.4 Summary of Related Work and Research Gap

Recent works demonstrate promising IDS solutions based on supervised ML approaches, although challenges persist regarding real-time adaptability, heterogeneity handling, and deployment feasibility [9]. Studies comparing ML strategies show that supervised methods excel when labeled data is available [11], whereas unsupervised and deep-learning approaches are widely explored for anomaly detection and emerging threats [1]. Additionally, several solutions leverage intelligent analytics in smart grids, intelligent transportation, healthcare, multimedia quality, and cyberdefense to build trustworthy IoT ecosystems [2, 4, 7, 22, 25].

Overall, literature converges on the need for IDS approaches that simultaneously guarantee high detection capability, computational efficiency, and suitability for edge deployment. This study addresses this gap by evaluating supervised (Decision Tree and Random Forest) and unsupervised (K-means and DBSCAN) approaches using NSL-KDD, while also analyzing execution complexity on Raspberry Pi. This dual perspective contributes to advancing both theoretical understanding and practical feasibility of ML-driven IDS in IoT environments [10, 16].

3 Methodology

This section describes the methodology adopted for the development and evaluation of the proposed intrusion detection system (IDS). First, it presents the NSL-KDD dataset used as the experimental basis, as well as the preprocessing pipeline responsible for normalization, encoding, and organization of the instances into training, validation, and test sets. Next, the machine learning algorithms are detailed, covering both supervised models (Decision Tree and Random Forest) and unsupervised methods (K-means and DBSCAN), along with the performance metrics used for evaluation. Finally, the implementation of the IDS in an edge environment using the Raspberry Pi Desktop operating system is described, together with the procedure for measuring execution time, latency, and computational resource usage, which supports the analysis of the practical feasibility

of the proposed approach.

3.1 Data Collection

The dataset used in this study is the **NSL-KDD** set, widely employed in research on intrusion detection in networked and IoT environments. This dataset was developed as an improved version of KDD Cup 1999, reducing redundancies and mitigating class imbalance, which makes model training more reliable and decreases the risk of overfitting.

Recent studies reinforce the suitability of NSL-KDD for experimentation in Intrusion Detection Systems (IDS) targeting IoT and general network security, highlighting its more balanced distribution between normal and malicious samples and the presence of clearly defined labels for different attack types. Works such as [1, 9, 11] have used this dataset to evaluate machine learning algorithms applied to network and IoT security, evidencing its relevance and consolidation in the literature.

The dataset contains records classified as *normal* or *malicious*, which allows the application of both supervised algorithms and unsupervised strategies for anomaly detection. In addition, its moderate size favors execution on platforms with limited computational resources, such as embedded devices and edge solutions, aligning with this work's objective of developing a system that is viable in real IoT environments.

3.2 Data Preprocessing

Data preprocessing is a critical step to ensure input quality before feeding the machine learning models. In this work, the entire pipeline was implemented in Python 3.9 using the pandas, NumPy, and scikit-learn libraries, in a Raspberry Pi Desktop environment (Debian-based Linux).

The experiments were carried out on a general-purpose computer configured to approximate the computational constraints of an edge or IoT gateway device, with the following characteristics:

- **Operating system:** Raspberry Pi Desktop (Linux)
- **Interpreter:** Python 3.9
- **Main libraries:** pandas, NumPy, scikit-learn, joblib
- **Hardware:** Intel(R) Core(TM) i5-7200U CPU @ 2.50GHz (2.71 GHz), 12 GB RAM, 100 GB disk storage.

This environment was chosen to approximate the execution conditions of an IoT gateway or edge device, where CPU, memory, and storage are limited and preprocessing cost becomes an important factor.

The preprocessing flow applied to the NSL-KDD dataset followed these steps:

- **Data integration and cleaning:** The original NSL-KDD files (KDDTrain+ and KDDTest+) were merged and converted into a single CSV file (NSL_KDD_preprocessed.csv). During this stage, duplicate records and rows with missing or inconsistent values were removed, ensuring a consistent dataset for model training.
- **Processing of numerical attributes:** Numerical attributes were standardized by normalization/scaling to a suitable range using scikit-learn preprocessing functions. This procedure helps model convergence and reduces the impact of scale differences among variables.
- **Encoding of categorical variables:** Categorical attributes in NSL-KDD (such as protocol, service, and flag) were transformed into numerical representations through encoding techniques, allowing the machine learning algorithms to process them. Strategies such as one-hot encoding were employed to avoid introducing artificial ordering among categories.
- **Separation of features and labels:** After these transformations, the dataset was organized into two components: (i) a feature matrix containing only normalized and encoded numeric attributes; and (ii) a label vector representing the class of each connection (normal or attack types).
- **Generation of the final experimental set:** The outcome of this pipeline was a preprocessed file (NSL_KDD_preprocessed.csv) used in both the supervised experiments (Decision Tree and Random Forest) and the unsupervised experiments (K-means and DBSCAN). This file incorporates all cleaning, normalization, and encoding steps, reducing data preparation overhead in subsequent executions.

These preprocessing decisions are fundamental to improving the accuracy, stability, and efficiency of the intrusion detection models in IoT networks, especially when executed on platforms with limited computational resources, such as the Raspberry Pi environment adopted in this study.

3.3 Evaluation of Supervised Methods

The supervised IDS training process is summarized in Algorithm 1.

Algorithm 1 Training of the supervised IDS model

NSL_KDD_preprocessed.csv Best model saved as ids_iot_rf_pi.joblib Load dataset D Split D into X (features) and y (labels) Apply preprocessing Split data into train / validation / test Define models: Decision Tree and Random Forest $F1_{best} \leftarrow -\infty$, $M_{best} \leftarrow \varnothing$ each model m Train m Evaluate on validation and test Compute weighted F1-score $F1(m) > F1_{best}$ $F1_{best} \leftarrow F1(m)$ $M_{best} \leftarrow m$ Save M_{best} M_{best}

The intrusion detection flow in the edge environment, executed on Raspberry Pi Desktop, is presented in Algorithm 2.

Algorithm 2 Intrusion detection in an edge environment (Raspberry Pi)

Trained model M from `ids_iot_rf_pi.joblib`, traffic file `trafego_simulado.csv` Intrusion predictions and edge performance metrics Load the model M from `ids_iot_rf_pi.joblib` Load the file `trafego_simulado.csv` into dataset T Split T into input features X_{edge} and ground-truth labels y_{edge} (when available) Measure initial CPU and memory usage of the device Start timing Obtain predictions: $\hat{y}_{\text{edge}} \leftarrow M(X_{\text{edge}})$ Stop timing and compute total inference time Compute average latency per sample: $\text{latency} = \dfrac{\text{total time}}{|X_{\text{edge}}|}$ Measure final CPU and memory usage y_{edge} is available Compute accuracy, precision, recall, $F1$-score, and confusion matrix between y_{edge} and $\hat{y}_{\text{edge}}$ Log: – total time and average inference latency – CPU and memory variation – classification metrics (when applicable) $\hat{y}_{\text{edge}}$ and recorded metrics

From a complexity standpoint, the algorithms used in this work have well-known costs in the literature. Considering n instances and d attributes, training a Decision Tree typically has complexity on the order of $O(n\,d\log n)$, assuming efficient partitioning of the data, whereas prediction with a trained tree depends only on its height, i.e., $O(h)$ per sample, with h generally proportional to $\log n$. For Random Forest, which aggregates T trees, training cost grows to approximately $O(T\,n\,d\log n)$, and inference cost becomes $O(T\,h)$, which explains the higher training time observed, although per-sample latency remains low and compatible with edge scenarios. For unsupervised methods, K-means presents cost $O(I\,k\,n\,d)$, where k is the number of clusters and I the number of iterations until convergence, while DBSCAN, when spatial indexing structures are available, can reach an average complexity close to $O(n\log n)$, with worst case $O(n^2)$ due to neighborhood searches. In all cases, the choice of a 5,000-instance subset for unsupervised experiments and the moderate size of NSL-KDD help keep these costs within practical limits, consistent with execution times and latencies measured in the Raspberry Pi Desktop environment.

3.4 Evaluation of Unsupervised Methods

In addition to the supervised models, this work evaluates the behavior of unsupervised algorithms for intrusion detection, focusing on their ability to separate normal and malicious traffic without directly using labels during training. For this purpose, a sample of 5,000 NSL-KDD instances was used, to which the same preprocessing pipeline adopted for the supervised models was applied. Then, the problem was converted into a binary setting, where connections labeled as `normal` were mapped to class 0, and all others, representing attacks, were mapped to class 1.

The evaluation process of K-means and DBSCAN is summarized in Algorithm **??**. First, K-means is executed with two clusters, producing labels that are compared with the binary normal/attack label using the adjusted Rand index (ARI) and normalized mutual information (NMI), in addition to the silhouette coefficient, which quantifies cohesion and separation of the clusters. Next, DBSCAN is applied with empirically defined neighborhood and density parameters, identifying dense regions and marking isolated instances as noise. For DBSCAN, the proportion of points labeled as noise is computed, and the silhouette coefficient is calculated only for instances belonging to valid clusters. These metrics enable a consistent comparison of the potential of K-means and DBSCAN as anomaly detection components in an IDS designed for IoT environments.

All source code used in this study, including preprocessing scripts, supervised model training, and unsupervised experiments, is available in an anonymous repository at: `https://github.com/artigosanonimos/From-Risk-to-Protection`.

3.5 Feature Selection

Feature selection aims to identify the most relevant variables for the intrusion detection task, reducing dimensionality and improving model performance. The following techniques are considered:

- **Statistical methods:** Correlation analysis and significance tests to identify attributes with higher discriminative power.
- **Automatic selection algorithms:** Use of methods such as Recursive Feature Elimination (RFE) to select near-optimal subsets of features.

Careful feature selection is fundamental to the development of efficient and effective models, particularly in resource-constrained IoT and edge environments.

3.6 Model Training and Evaluation

Several machine learning algorithms are explored in this work, including:

- **Supervised:** Decision Tree, Random Forest.
- **Unsupervised:** K-means, DBSCAN.

Model evaluation is carried out using metrics such as accuracy, precision, recall, F1-score, and false positive rate, with particular focus on adapting the models to the resource limitations of IoT devices and edge gateways. For unsupervised approaches, clustering-based metrics such as ARI, NMI, silhouette coefficient, and noise proportion are used to quantify their suitability for anomaly detection.

3.7 Implementation and Validation

The implementation of the intrusion detection system is performed on hardware platforms with limited resources, such as Raspberry Pi, to emulate realistic operating conditions in IoT networks. System validation is conducted at two levels:

- **Technical validation:** Performance tests focusing on execution time, resource usage (CPU and memory), and energy consumption where applicable.
- **Practical validation:** Deployment in controlled environments to assess effectiveness in detecting intrusions or anomalous traffic in near real time.

This multi-level validation strategy ensures that the proposed system is not only accurate in terms of threat detection, but also operationally feasible for deployment in real-world IoT and edge scenarios.

4 Results and Discussion

This section presents the results obtained from the experiments carried out with supervised machine learning algorithms (Decision Tree and Random Forest) and unsupervised algorithms (K-means and DBSCAN), as well as the validation of the selected model in an edge environment based on Raspberry Pi Desktop.

4.1 Performance of Supervised Models

The supervised experiments were conducted using the NSL-KDD dataset, as described in Section 3. The dataset was split into 60% for training, 20% for validation, and 20% for testing, using stratified sampling to preserve the class distribution.

The Decision Tree required approximately 2.17 seconds of training time. On the test set, the model achieved overall accuracy close to 100% and a **weighted F1-score of about 0.9970**, indicating excellent performance in classifying the main traffic categories, especially `normal` and denial-of-service attacks (`neptune`, `smurf`). The average inference latency, measured on a subset of up to 1000 test samples, was approximately 0.0082 ms per instance, demonstrating very high computational efficiency.

The Random Forest, configured with 50 trees and maximum depth of 15, required approximately 5.22 seconds for training. In return, it achieved slightly superior performance, with a weighted F1-score close to 0.9973 and also near-perfect accuracy. The average inference latency was approximately 0.0194 ms per instance, which remains compatible with deployment in resource-constrained environments.

In both models, the majority classes (`normal`, `neptune`, `ipsweep`, `satan`, `smurf`, `portsweep`) presented precision and recall values very close to 1.0. For some extremely rare classes with only one or two occurrences in the test set, undefined or zero metrics appeared, generating warning messages from the evaluation library. This behavior results from the intrinsic imbalance of the NSL-KDD dataset and does not compromise overall system performance.

Considering the weighted F1-score and stability on minority classes, Random Forest was selected as the final supervised model and serialized as a complete pipeline (preprocessing + classifier) in the file `ids_iot_rf_pi.joblib`.

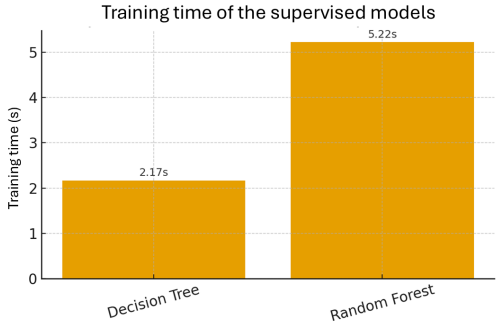

Figure 1: Training time of the supervised machine learning models (Decision Tree and Random Forest) evaluated on the NSL-KDD dataset.

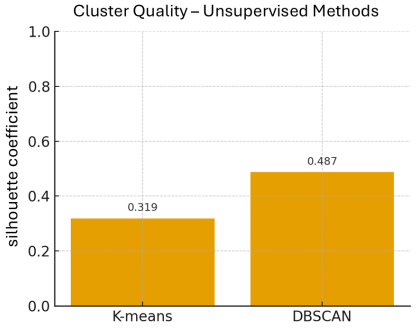

Figure 2: Silhouette coefficient obtained by the unsupervised methods K-means and DBSCAN applied to a 5000-instance sample of the NSL-KDD dataset.

4.2 Performance of Unsupervised Models

For the unsupervised experiments, a sample of 5000 NSL-KDD instances was used, preserving all 41 original attributes, with the labels kept only for posterior evaluation. For analysis, the problem was converted into a binary scenario in which label `normal` was mapped to 0, and all other attack classes were mapped to 1.

Following the procedure described in Algorithm **??**, K-means achieved an Adjusted Rand Index (ARI) of approximately 0.39, a Normalized Mutual Information (NMI) of 0.43, and a silhouette coefficient of about 0.32. DBSCAN, in turn, achieved a silhouette coefficient of approximately 0.49, while labeling about 43% of the instances as noise.

Using `eps = 0.5` and `min_samples = 10`, DBSCAN classified approximately 2157 instances (around 43% of the sample) as noise (`label = -1`), interpreted as potential anomalies. For instances assigned to valid clusters, the silhouette coefficient reached approximately 0.49, higher than that obtained with K-means. This indicates that DBSCAN forms more cohesive and better-separated clusters; however, the high proportion of noise suggests an aggressive anomaly detection behavior.

4.3 Edge Deployment Validation

To evaluate the feasibility of deployment in an edge environment, the selected Random Forest model was executed on a Raspberry Pi Desktop platform. A simulated traffic file (`trafego_simulado.csv`) containing 2000 randomly sampled and preprocessed NSL-KDD instances was used, including both normal traffic and different attack categories.

The model classified all 2000 samples in approximately **0.035 seconds**, resulting in an average latency of about 0.0175 ms per instance. Memory consumption fluctuated between **119 MB and 121 MB** during inference, while CPU usage briefly reached 100%, returning to normal shortly thereafter.

From a detection perspective, the model maintained full accuracy over the evaluated samples, with precision, recall, and F1-score values close to 1.0 for the main attack categories. Slight degradation was observed in underrepresented classes, such as `warezclient`, but without any meaningful impact on global detection performance.

Overall, the results confirm the suitability of machine learning-based IDS solutions for IoT environments, especially when supported by well-designed preprocessing and algorithm selection. Random Forest demonstrated superior robustness relative to Decision Tree, which is particularly relevant in IoT contexts, where false negatives may generate critical consequences. The unsupervised models, in turn, proved adequate primarily as complementary mechanisms for anomaly awareness rather than as standalone detection cores.

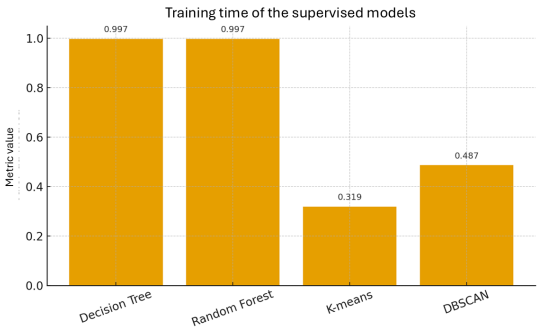

Figure 3: Performance comparison between supervised algorithms (weighted F1-score) and unsupervised algorithms (silhouette coefficient) applied to the NSL-KDD dataset.

5 Conclusion

This work investigated the use of supervised machine learning algorithms (Decision Tree and Random Forest) and unsupervised algorithms (K-means and DBSCAN) in the construction of an Intrusion Detection System (IDS) for Internet of Things environments, with emphasis on computational complexity and execution feasibility in edge scenarios based on Raspberry Pi Desktop. The adopted methodology comprised preprocessing of the NSL-KDD dataset, construction of pipelines with normalization and attribute encoding, stratified partitioning into training, validation, and test sets, and systematic evaluation of the supervised models. The results showed that both Decision Tree and Random Forest achieved very high predictive accuracy, with weighted F1 values close to 0.997 on the test set. Random Forest presented slightly superior performance, particularly in minority attack classes, and was therefore selected as the final model and deployed in the Raspberry Pi Desktop environment. From a computational standpoint, training times on the order of seconds and inference latencies on the order of hundredths of a millisecond per instance indicate that the proposed solution is compatible with edge deployment. The validation in Raspberry Pi Desktop confirmed this feasibility, with stable accuracy, reduced latency, and moderate memory consumption when classifying 2000 simulated traffic samples. The unsupervised methods demonstrated more modest capacity to separate normal and malicious traffic. K-means presented moderate agreement with the binary ground truth, while DBSCAN generated more cohesive clusters, albeit at the cost of labeling a large fraction of samples as noise. These findings reinforce the idea that unsupervised learning should be used as a complementary anomaly-monitoring module rather than replacing supervised IDS models. Overall, the study demonstrates that it is possible to reconcile high predictive capability with acceptable computational cost in IoT security systems. As future work, we intend to evaluate the approach using more recent IoT-native datasets, apply class balancing and synthetic data generation techniques to improve rare attack detection, compare performance with lightweight deep learning models, and integrate unsupervised anomaly detection modules to enhance resilience against emerging cyber threats in real time.

References

[1] Alsharif, N. A., Mishra, S., and Alshehri, M. Ids in iot using machine learning and blockchain. *Engineering, Technology & Applied Science Research*, 13(4):11197–11203, 2023.

[2] Barbosa, R., Ogobuchi, O. D., Joy, O. O., Saadi, M., Rosa, R. L., Al Otaibi, S., and Rodríguez, D. Z. Iot based real-time traffic monitoring system using images sensors by sparse deep learning algorithm. *Computer Communications*, 210:321–330, 2023.

[3] Bhavsar, M. H., Bekele, Y. B., Roy, K., Kelly, J. C., and Limbrick, D. Fl-ids: Federated learning-based intrusion detection system using edge devices for transportation iot. *IEEe Access*, 12:52215–52226, 2024.

[4] Carrillo, D., Kalalas, C., Raussi, P., Michalopoulos, D. S., Rodríguez, D. Z., Kokkoniemi-Tarkkanen, H., Ahola, K., Nardelli, P. H., Fraidenraich, G., and Popovski, P. Boosting 5g on smart grid communication: A smart ran slicing approach. *IEEE Wireless Communications*, 30(5):170–178, 2022.

[5] Carvalho Barbosa, R., Shoaib Ayub, M., Lopes Rosa, R., Zegarra Rodríguez, D., and Wuttisittikulkij, L. Lightweight pvidnet: A priority vehicles detection network model based on deep learning for intelligent traffic lights. *Sensors*, 20(21):6218, 2020.

[6] de Sousa, A. L., OKey, O. D., Rosa, R. L., Saadi, M., and Rodriguez, D. Z. Unified approach to video-based ai inference tasks in augmented reality systems assisted by mobile edge computing. pages 1–5, 2023.

[7] dos Santos, M. R., Batista, A. P., Rosa, R. L., Saadi, M., Melgarejo, D. C., and Rodríguez, D. Z. Asqm: Audio streaming quality metric

based on network impairments and user preferences. *IEEE Transactions on Consumer Electronics*, 69(3):408–420, 2023.

[8] Fonseca, D., da Silva, K. C. N., Rosa, R. L., and Rodríguez, D. Z. Monitoring and classification of emotions in elderly people. In *2019 International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, pages 1–6. IEEE, 2019.

[9] Haji, S. H. and Ameen, S. Y. Attack and anomaly detection in iot networks using machine learning techniques: A review. *Asian journal of research in computer science*, 9(2):30–46, 2021.

[10] Kaushik, S., Bhardwaj, A., Almogren, A., Bharany, S., Altameem, A., Rehman, A. U., Hussen, S., and Hamam, H. Robust machine learning based intrusion detection system using simple statistical techniques in feature selection. *Scientific Reports*, 15(1):3970, Feb 1 2025.

[11] Malathi, C. and Padmaja, I. N. Identification of cyber attacks using machine learning in smart iot networks. *Materials Today: Proceedings*, 80:2518–2523, 2023.

[12] Matthew, U. O., Rosa, R. L., Kazaure, J. S., Adesina, O. J., Oluwatimilehin, O. A., Oforgu, C. M., Asuni, O., Okafor, N. U., and Rodriguez, D. Z. Software-defined networks in iot ecosystems for renewable energy resource management. pages 1–5, 2024.

[13] Okey, O. D., Maidin, S. S., Adasme, P., Lopes Rosa, R., Saadi, M., Carrillo Melgarejo, D., and Zegarra Rodríguez, D. Boostedenml: Efficient technique for detecting cyberattacks in iot systems using boosted ensemble machine learning. *Sensors*, 22(19):7409, 2022.

[14] Okey, O. D., Rodriguez, D. Z., and Kleinschmidt, J. H. Enhancing iot intrusion detection with federated learning-based cnn-gru and lstm-gru ensembles. In *2024 19th International Symposium on Wireless Communication Systems (ISWCS)*, pages 1–6. IEEE, 2024.

[15] Pinto, G. E., Rosa, R. L., and Rodriguez, D. Z. Applications for 5g networks. *INFOCOMP Journal of Computer Science*, 20(1), 2021.

[16] Rahman, M. M., Shakil, S. A., and Mustakim, M. R. A survey on intrusion detection system in iot networks. *Cyber Security and Applications*, 3:100082, 2025.

[17] Rezaie, V., Parnianifard, A., Zegarra Rodriguez, D., Mumtaz, S., and Wuttisittikulkij, L. Speech emotion recognition using anfis and pso-optimization with word2vec. *Research Square*, 2022.

[18] Ribeiro, D. A., Melgarejo, D. C., Saadi, M., Rosa, R. L., and Rodríguez, D. Z. A novel deep deterministic policy gradient model applied to intelligent transportation system security problems in 5g and 6g network scenarios. *Physical Communication*, 56:101938, 2023.

[19] Ribeiro, D. A., Silva, J. C., Lopes Rosa, R., Saadi, M., Mumtaz, S., Wuttisittikulkij, L., Zegarra Rodriguez, D., and Al Otaibi, S. Light field image quality enhancement by a lightweight deformable deep learning framework for intelligent transportation systems. *Electronics*, 10(10):1136, 2021.

[20] Rodríguez, D. Z. and Möller, S. Speech quality parametric model that considers wireless network characteristics. In *2019 Eleventh International Conference on Quality of Multimedia Experience (QoMEX)*, pages 1–6. IEEE, 2019.

[21] Rosa, R. L., Rodriguez, D. Z., and Bressan, G. Sentimeter-br: A social web analysis tool to discover consumers' sentiment. In *2013 IEEE 14th international conference on mobile data management*, volume 2, pages 122–124. IEEE, 2013.

[22] Silva, D. H., Maziero, E. G., Saadi, M., Rosa, R. L., Silva, J. C., Rodriguez, D. Z., and Igorevich, K. K. Big data analytics for critical information classification in online social networks using classifier chains. *Peer-to-Peer Networking and Applications*, 15(1):626–641, 2022.

[23] Silva, D. H., Rosa, R. L., and Rodriguez, D. Z. Sentimental analysis of soccer games messages from social networks using user's profiles. *INFOCOMP Journal of Computer Science*, 19(1), 2020.

[24] Teodoro, A. A., Gomes, O. S., Saadi, M., Silva, B. A., Rosa, R. L., and Rodríguez, D. Z. An fpga-based performance evaluation of artificial neural network architecture algorithm for iot. *Wireless Personal Communications*, 127(2):1085–1116, 2022.

[25] Teodoro, A. A., Silva, D. H., Rosa, R. L., Saadi, M., Wuttisittikulkij, L., Mumtaz, R. A., and Rodriguez, D. Z. A skin cancer classification approach using gan and roi-based attention mechanism. *Journal of Signal Processing Systems*, 95(2):211–224, 2023.

[26] Teodoro, A. A., Silva, D. H., Saadi, M., Okey, O. D., Rosa, R. L., Otaibi, S. A., and Rodríguez, D. Z. An analysis of image features extracted by cnns to design classification models for covid-19 and non-covid-19. *Journal of signal processing systems*, pages 1–13, 2023.

[27] Terra Vieira, S., Lopes Rosa, R., Zegarra Rodriguez, D., Arjona Ramírez, M., Saadi, M., and Wuttisittikulkij, L. Q-meter: Quality monitoring system for telecommunication services based on sentiment analysis using deep learning. *Sensors*, 21(5):1880, 2021.

[28] Ugochukwu, O. M., Rosa, R. L., Adenike, O. O., and Rodriguez, D. Z. Advancing cybersecurity use of sensitive data in electronic healthcare system: A review of privacy and regulations. *INFOCOMP Journal of Computer Science*, 23(2), 2024.

[29] Ugochukwu, O. M., Rosa, R. L., Adenike, O. O., and Rodriguez, D. Z. Distributed healthcare privacy protection in emerging cybersecurity use of sensitive data. *INFOCOMP Journal of Computer Science*, 23(2), 2024.