Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2024.059047

ARTICLE

A Robust Security Detection Strategy for Next Generation IoT Networks

Hafida Assmi[1], Azidine Guezzaz[1], Said Benkirane[1], Mourade Azrour[2,*], Said Jabbour[3], Nisreen Innab[4] and Abdulatif Alabdulatif[5]

[1]Technology Higher School Essaouira, Cadi Ayyad University, Essaouira, 44000, Morocco

[2]IMIA Laboratory, MSIA Team, Faculty of Sciences and Techniques, Moulay Ismail University of Meknes, Errachidia, 50050, Morocco

[3]CRIL-CNRS, Artois University, Lens, 62300, France

[4]Department of Computer Science and Information Systems, College of Applied Sciences, AlMaarefa University, Diriyah, Riyadh, 13713, Saudi Arabia

[5]Department of Computer Science, College of Computer, Qassim University, Buraydah, 52571, Saudi Arabia

*Corresponding Author: Mourade Azrour. Email: mo.azrour@umi.ac.ma

Received: 27 September 2024 Accepted: 03 December 2024 Published: 03 January 2025

ABSTRACT

Internet of Things (IoT) refers to the infrastructures that connect smart devices to the Internet, operating autonomously. This connectivity makes it possible to harvest vast quantities of data, creating new opportunities for the emergence of unprecedented knowledge. To ensure IoT securit, various approaches have been implemented, such as authentication, encoding, as well as devices to guarantee data integrity and availability. Among these approaches, Intrusion Detection Systems (IDS) is an actual security solution, whose performance can be enhanced by integrating various algorithms, including Machine Learning (ML) and Deep Learning (DL), enabling proactive and accurate detection of threats. This study proposes to optimize the performance of network IDS using an ensemble learning method based on a voting classification algorithm. By combining the strengths of three powerful algorithms, Random Forest (RF), K-Nearest Neighbors (KNN), and Support Vector Machine (SVM) to detect both normal behavior and different categories of attack. Our analysis focuses primarily on the NSL-KDD dataset, while also integrating the recent Edge-IIoT dataset, tailored to industrial IoT environments. Experimental results show significant enhancements on the Edge-IIoT and NSL-KDD datasets, reaching accuracy levels between 72% to 99%, with precision between 87% and 99%, while recall values and F1-scores are also between 72% and 99%, for both normal and attack detection. Despite the promising results of this study, it suffers from certain limitations, notably the use of specific datasets and the lack of evaluations in a variety of environments. Future work could include applying this model to various datasets and evaluating more advanced ensemble strategies, with the aim of further enhancing the effectiveness of IDS.

KEYWORDS

IoT security; intrusion detection; RF; KNN; SVM; EL; NSL-KDD; Edge-IIoT

1 Introduction

The Internet of Things (IoT) is seen as an indispensable pillar in sectors like smart cities, transport, and healthcare [1]. However, the evolution of the IoT and mobile Internet has highlighted the limits of the centralized cloud, particularly in terms of latency and efficiency [2]. To remedy this, edge computing technologies such as fog computing and cloudlets have been introduced to optimize latency and network performance [3]. Security means preserving the integrity, availability, and confidentiality of data, through strategies such as access control, information encryption, and strict system configuration [4]. However, in the face of increasingly complex environments and the challenges they pose, these approaches have a number of limitations. This study concentrates on the analysis of network intrusion detection systems (NIDS), abusing innovative artificial intelligence techniques such as ML and DL algorithms, to detect abnormal or suspicious behavior, and aims to improve the performance of NIDS by adopting an ensemble learning method, specifically the voting classifier algorithm. This approach combines the strengths of three powerful algorithms, Random Forest (RF), K-Nearest Neighbors (KNN), and Support Vector Machine (SVM) to identify normal activities and various types of attack within the NSL-KDD dataset, while also applying to a recent dataset, Edge-IIoT, adapted to industrial IoT environments. Two contributions were validated:

- Firstly, we employ the Ensemble Learning (EL) method to maximize and enhance the proposed model's performance, while minimizing the learning time to maximize the system's overall efficiency.
- Secondly, to develop effective detection systems, a classification model is designed following three main steps: data preprocessing, training and building the model, and evaluating the model.

The rest of this paper is organized as following. The Section 2 provides background and review some related work on IDS approaches incorporating ML, DL and EL algorithms. The Section 3 presents and elaborates the proposed new framework. The Section 4 highlights the experimental evaluation results obtained with the model. The paper then concludes with suggestions for future research directions.

2 Background and Related works

This section presents general overviews and a critique of selected recent work on IDSs that integrate both ML and DL algorithms with the aim of improving the security of the IoT.

2.1 Background

Mobile Edge Computing or Fog Computing, however, offers the most auspicious solution for interoperability among various heterogeneous devices. Edge networks and Fog devices provide a larger scalability and flexibility. Furthermore, they are practical and easy to improve upon [3]. Various researchers have suggested different architectures for the IoT, the three layers represent the basic architecture most commonly used in IoT. However, it is still inadequate for emerging IoT applications [4]. The five-layer architecture shown in Fig. 1, is a conceptual structure that splits the IoT system into five separate functional layers to facilitate the design, deployment, and management of IoT systems.

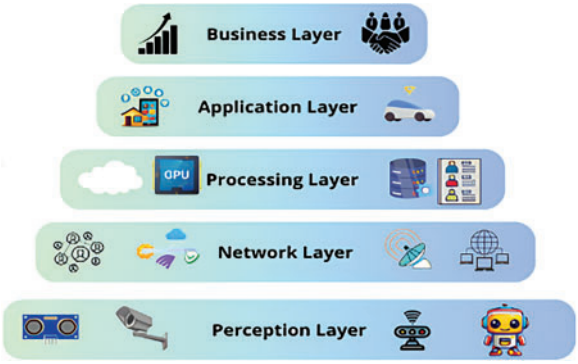

Figure 1: Five layers IoT architectures

This architecture includes the perception layer, which encompasses the end devices and digital applications, such as actuators, sensors, machines and other equipment, responsible for collecting and analyzing data. Here, raw data are acquired before being transmitted to the subsequent layer for pre-treatment. Next, the network layer, or transport layer, ensures data transmission between the various layers of the IoT architecture. This layer ensures connectivity between network nodes, using various wireless communication technologies, such as cellular networks, Wi-Fi and Bluetooth. The processing layer, located at the intermediate level, has the role of processing, storing and analyzing data from the transport layer [4]. It can also run basic analysis algorithms to extract relevant information from the raw data. The application layer, or Graphical User Interface (GUI) layer, represents the element with which the user interacts. It uses the data processed by the previous levels to make decisions, trigger actions, provide information to users and generate reports. The business layer converts data into actionable information and oversees IoT services, including security, device management, identity management, data and policy management. It ensures that data processing is carried out securely, in line with the policies established by the organization or IoT service provider.

Therefore, security is primarily about ensuring data integrity, availability, and confidentiality through various approaches, including authorization, secure storing of data, auditing, managing systems, and setting up configurations [5]. However, system complexity and other challenges make the application of these methods difficult and give rise to numerous problems. To enhance the security of IoT networks, IDS have been developed.

Our research focuses on NIDS, which are placed at the edge of the network infrastructure and analyze a real-time copy of the traffic. It is essential to improve these NIDS by integrating advanced artificial intelligence technologies, such as ML and DL. Identifying suspicious or abnormal activities is crucial for ensuring that IoT environments are secure and protected against a variety of attacks, including Keylogging Denial of Service (DoS), Distributed Denial of Service (DDoS) and Service Scanning. To enhance IDS performance, we have combined deep learning, ML and EL approaches. However, despite these advances, several challenges remain, notably real-time detection, unbalanced classes, high-dimensionality, optimization of massive data volumes and temporal performance constraints [6].

2.2 Related Works

In 2018, Benaddi et al. [7] have developed an approach called Principal Component Analysis-fuzzy (PCA-fuzzy) clustering-KNN, which integrates principal component analysis (PCA) and fuzzy

clustering with KNN-based feature selection techniques. This method was implemented on the NSL-KDD dataset to identify DoS, Probe, U2R and R2L attacks. At the same time, Resende et al. [8] have conducted an in-depth study of key concepts related to IDS, attack types, modeling, frequently used approaches and classifications. Their study focused on approaches based on RF, taking into account the specificities of these models. Fei et al. [9] analyze ML techniques applied to data flow analysis in cyber-physical systems, addressing several perspectives, including advice on integrating ML methods into Cloud and Fog architectures. In addition, Meidan et al. [10] introduce an innovative anomaly detection approach for the IoT, called N-BaIoT. This approach captures snapshots of network behavior and uses deep autoencoders to detect anomalies in network traffic generated by compromised IoT devices.

In 2019, Zeng et al. [11] provided a trivial DL-based scheme to classify encapsulated traffic and detect intrusions. In addition, Guezzaz et al. [12] introduced an IDS framework for monitoring network traffic, based on Pcapsocks and a traffic classification scheme based on MLP. This approach was used to detect and classify occurrences as either normal or unusual. In addition, Chaabouni et al. [13] explored ML algorithms used in NIDSs specifically designed for IoT policies, while highlighting the particular challenges these NIDSs face in IoT positions.

In 2020, Chaabouni et al. [14] have developed a ML-based IDS, dubbed OneM2M, to ensure IoT security. Test results reveal a detection rate of around 93.80%, a precision of 92.95%, FPR of 1.53%, an accuracy (ACC) of 92.32%, and a CPU learning time of 9280 ms. Simultaneously, Tang et al. [15] have designed a real-time detection method for SQL injection attacks in HTTP traffic. Their method is supported by the use of different ANN models, such as long-term memory networks (LSTM) and multilayer perceptrons (MLP). At the same time, Wazirali et al. [16] suggested a method for strengthening an IDS by adjusting KNN hyperparameters using five-fold cross-validation. Hussain et al. [17] have extended the NIDS concept by addressing several security issues in the IoT. Their study evaluates the application of ML and DL techniques to solve problems. In addition, Thakker et al. [18] examined the various IDS datasets used for evaluating intrusion detection models. Their study includes an overview of ML and DL techniques applied to IDS, as well as an analysis of the CIC-IDS-2017 and CSE-CIC-IDS-2018 datasets. They also explored current advances in IDS datasets, which can serve as a reference for different research communities to use new IDS datasets to develop effective ML- and DL-based intrusion detection systems.

In 2021, Gu et al. [19] have developed an intrusion detection system based on an SVM model, incorporating a NB feature transformation applied to the original data. This approach generates high-quality data, underlining the crucial importance of data quality in optimizing the performance of IDS. The model demonstrates exceptional accuracy on several datasets, reaching 99.35% on NSL-KDD, 98.58% on KYOTO 2006+, 98.92% on CICIDS 2017, and 93.75% on UNSW-NB15. Concurrently, Jin et al. [20] have exploited Gradient Boosting Machine (GBM), Extreme Gradient Boosting (XGB) and LightGBM techniques to enhance the performance of an IDS featuring a CNN, suitable for binary and multi-class clustering tasks. Experimental outcomes demonstrate that the it guarantees a minimum detection rate of 99.7%, attesting to its remarkable efficiency. Concomitantly, Guezzaz et al. [21] designed a decision tree (DT)-based NIDS model based on the CICIDS 2017 and NSL-KDD datasets. They then compared the performance of their model with that of other approaches using the same datasets. This proposal achieved an average accuracy of 98.8% on CICIDS 2017 and 99.42% on NSL-KDD. Simultaneously, Debicha et al. [22] studied the impact of adversarial attacks on intrusion detection based on deep learning. In addition, by evaluating adversarial training efficiency as a defense strategy, they showed that adversarial examples, when sufficiently distorted, can mislead the detector. However, this approach enhances the robustness of intrusion detection through the integration of

adversarial training. Furthermore, Peterson et al. [23] offer a detailed analysis of one of the most recent datasets, Bot-IoT, with around 73 million instances (Big Data). Their aim is to provide researchers with an in-depth understanding of Bot-IoT, outlining its key features and highlighting potential challenges to consider. In addition, they looked at data cleansing procedures, followed by an overview of applications for this dataset in published research. Shafiq et al. [24] have taken up the challenge of efficient feature selection to accurately detect malicious traffic in IoT networks. Their solution, named CorrAUC, is based on an innovative wrapper method for accurately sorting and selecting features relevant to the chosen ML, employing the area under the curve (AUC) as a performance measure. They subsequently incorporated TOPSIS and Shannon entropy in a bidirectional framework for validating the chosen patterns, with the aim of identifying the malicious content in IoT networks. This approach was experienced on the Bot-IoT database with four different machine learning algorithms. Experimental results revealed average performances in excess of 96%. Leevy et al. [25] presented a simplified method for Bot-IoT, reducing the number of features used to just 3 of the 29 available. They opted for a simple learning algorithm, such as a decision tree classifier, to ensure accurate classification. The results show that the predictive models achieve mean scores of over 0.99 for the area under the receiver operating characteristic curve (AUC ROC), as well as for the area under the recall and precision curve (AUPRC).

In 2022, Mohy-eddine et al. [26] have employed the wustl-iiot-2021 and BoT-IoT datasets for expanding IDS by using EL, specifically for IoT edge computing. Their methodology incorporates the application of Pearson correlation for feature selection and the use of isolation forest for outlier elimination. Experimental results revealed that this model performed remarkably well, with success rates (ACC) of 99.99% and 99.12%, AUC scores of 92.48% and 99.3%, and Matthew correlation coefficients (MCC) of 92.17% and 93.96% on the BoT-IoT and wustl-iiot-2021 datasets. Besides, Yang et al. [27] carried out an extensive literature review, studying 119 major papers in the field of anomaly-based intrusion detection. Their research covered various aspects, including data preprocessing, evaluation criteria, attack detection techniques, and other key elements. Furthermore, Sengan et al. [28] have proposed DAR-ML, an innovative solution for detecting DoS attacks in healthcare data. The results obtained show that this approach achieves 98.19% accuracy, ensuring high reliability with a remarkably low false alarm rate. At the same time, Liu et al. [29] developed an intrusion detection system to solve the challenges encountered in WSNs by combining the KNN algorithm and arithmetic optimization (AOA). This system, evaluated on the WSN-DS dataset, achieved 99% accuracy. The application of the PL-AOA algorithm for the detection of DoS attacks showed a significant improvement of 10% over the use of the KNN algorithm alone. Asif et al. [30] introduced the MR-IMID model, based on MapReduce, offering reliable management of large datasets using basic hardware infrastructures. the results show an accuracy of 97.7% in the learning phase and 95.7% in the validation phase. Simultaneously, Fu et al. [31] have developed a traffic IDS using the NSL-KDD dataset. Their method combines an attention mechanism with a bidirectional long-term memory network (Bi-LSTM), creating a deep learning (DL) model for network intrusion detection (DLNID). To overcome the problem of data imbalance, they incorporated adaptive synthetic sampling (ADASYN). The results indicated that their model outperformed the comparison methods, with an accuracy of 90.73% and an F1-score of 89.65%. Saba et al. [32] proposed an IDS based on CNN, using the BoT-IoT and NID datasets. Their model showed significant improvements in accuracy, reaching 92.85% and 99.51%, respectively. In addition, Roy et al. [33] have designed a ML-based IDS to efficiently identify anomalies and cyber-attacks, using the NSL-KDD and CICIDS2017 datasets. This proposition makes it possible to target the features most essential for intrusion detection. Thanks to several optimizations, such

as multicollinearity elimination, sampling and dimensionality reduction, the model achieves a high detection rate while reducing false alarms.

In 2023, Mohy-eddine et al. [4] have designed an IDS based on the KNN algorithm, incorporating feature selection via an elective approach that combines PCA, statistical tests and a genetically based algorithm. In order to optimize model performance on unbalanced target datasets, such as BoT-IoT, they adopted the Matthews Correlation Coefficient (MCC), reaching a score of 97%, an accuracy (ACC) of 99.99%, and a processing time of 102 s for the five selected features. In addition, Ennaji et al. [34] have designed an innovative IDS, i-2NIDS, based on ML. By exploiting the NSL-KDD dataset, this system can distinguish normal activity and detect various kinds of attack, including DDoS/DoS, Probing, R2L and U2R. The results of the experiments confirmed the efficiency of the model, with an accuracy rate of around 99% in tests. Simultaneously, He et al. [35] provide an in-depth analysis of DL-based IDS and explore in detail black-box and white-box adversarial attacks against deep neural networks (DNNs), examining their relevance in the context of NIDS. Gaurav et al. [36] review the body of research devoted to malware detection, covering both dynamic and static detection methods, as well as hybrid approaches and techniques that improve detection efficiency. Harini et al. [37] have proposed a three-layer intrusion detection method for attack identification and prediction. The first layer employs a weighted-DNN, the second associates a CNN with a LSTM, and the third layer integrates the XGBoost algorithm. To deal with the imbalance of minority attack classes, an adaptive synthetic oversampling algorithm (ADASYN) is used to generate additional samples. Evaluations carried out on the NSL-KDD, CIDDS 001 and CICIDS-2017 datasets show outstanding performance, with an accuracy rate of 97.94% for NSL-KDD, 97.9% for CIDDS 001 and 98.3% for CICIDS-2017. Concomitantly, AI Lail et al. [38] have developed a NIDS that exploits machine learning to effectively identify modern attacks. Their approach, using the random forest model, surpassing previous models, achieving a detection rate for modern network attacks of up to 97%. In addition, to address the challenges identified, Song et al. [39] proposed an innovative method combining temporal convolutional network (TCN). This approach is based on a synchronized bidirectional recurrent unit (BiGRU) and a self-attention device. Using such an approach, an outstanding accuracy of 97.83% was achieved on the CSE-CIC-IDS2018 public dataset. Vitorino et al. [40] propose a systematization of knowledge (SoK) to synthesize and summarize adversarial learning approaches, while addressing open questions related to their application in the field of NIDS. This synthesis also highlights the essential characteristics that a contradictory example must possess to be considered realistic. Zhang et al. [41] propose a new classification system for detecting intrusions, combining advanced techniques in feature engineering and optimization of models. Their approach is based on advanced feature engineering, incorporating methods such as redundancy minimization and relevance maximization (mRMR), correlation optimization via mutual information, and the use of synthetic minority oversampling (SMOTE) to process network data. Additionally, Yao et al. [42] propose an innovative and optimized NIDS, taking advantage of a GRU bidirectional autoencoder and the Soft Voting method to efficiently detect unknown attacks, including zero-day attacks. The performance evaluation, carried out on the WSN-DS, KDD CUP99 and UNSW-NB15 datasets, shows recognition rates of 97.91%, 98.23% and 98.92% respectively. Louk et al. [43] have evaluated an innovative ensemble model, called Dual-IDS, using the NSL-KDD, UNSW-NB15 and HIKARI-2021 datasets. Such approach cleverly merges two well-established ensemble techniques: bagging and gradient-enhanced decision tree (GBDT). The results show that this combination is a particularly effective solution for anomaly-based intrusion detection.

In 2024, Saied et al. [44] have written a comprehensive review of recent advances in artificial intelligence applied to intrusion detection in IoT domain. They made a careful selection of articles, classifying them according to the artificial intelligence algorithms used to enhance IoTsecurity devices.

Akhiat et al. [45] used KDDCup-99 network dataset to evaluate the performance of IDS-EFS tool, designed to identify the optimal subset of features for attack detection, by comparing it with other increasingly popular feature selection methods. The results demonstrated a considerable improvement in performance, with accuracy, precision and recall rates approaching 99.9%. Furthermore, Wang et al. [46] propose a robust framework called Learn-IDS, which effectively adjusts to the challenges posed by the datasets used in training deep learning models. Experimental results reveal that this platform improves detection accuracy and dynamically adapts to emerging threats in complex scenarios. Biswas et al. [47] proposed a high-performance, machine-learning-based, real-time IDS framework that incorporates hybrid feature selection techniques. They also carried out an in-depth analysis on five public datasets, including CICIDS2017, NSL-KDD, UNSW-NB15, ISCX-IDS2012 and KDD Cup99. The findings demonstrate that the IDS presented achieves remarkable results, 99.98% accuracy in detecting malicious traffic. Simultaneously, Paya et al. [48] introduced Apollon, an innovative defense system designed to protect IDSs against adversarial machine learning (AML) attacks. To evaluate its effectiveness, they used three datasets: CSE-CIC-DS-2018, CIC-DDoS 2019 and CIC-IDS-2017, and trained several classifiers (MLP, RF, DT, NB and LR) to compare them with the proposed solution. The results underline the robustness of the Apollon system against AML attacks, as well as its ability to effectively detect these attacks while preserving optimal performance on traditional network traffic. In this context, Table 1 presents a comparison of different approaches to intrusion detection systems (IDS) aimed at enhancing the security of IoT environments.

Table 1: Comparing IDS approaches offering secure IoT

Papers	Year	Algorithms	Methods	Accuracy	Data
Benaddi et al. [7]	2018	KNN	PCA-fuzzy clustering-KNN technique	94.23% (DoS) 69.87% (R2L) 80.09% (U2L) 78.86% (Probe)	NSL-KDD
Zeng et al. [11]	2019	DL	Deep-full-range (DFR)	99.85% 99.41%	ISCX VPN-non VPN ISCX 2012 IDS
Chaabouni et al. [14]	2020	Decision tree J48	–	92.32%	OneM2Mdata
Tang et al. [15]	2020	LSTM, MLP	–	99.67%	ISP
Wazirali et al. [16]	2020	KNN in supervised KNN insemi -supervised	5-fold cross validation	99.10% 98.49%	NSL-KDD
Jin et al. [20]	2021	CNN	GBM, XGB, LightGBM	99.7%	BoT-IoT, IoT Network Intrusion MQTT-IoT-IDS2020 IoT-23

(Continued)

Table 1 (continued)

Papers	Year	Algorithms	Methods	Accuracy	Data
Gu et al. [19]	2021	SVM	NB feature	99.35% 93.35% 98.58% 98.92%	NSL-KDD UNSW-NB15 KYOTO 2006+ CICIDS 2017
Guezzaz et al. [21]	2021	DT	–	99.42% 98.8%	NSL-KDD CICIDS 2017
Douiba et al. [1]	2022	GB, DT	CatBoost	99.81% 99.98% 100% 100%	NSL-KDD IoT-23 BoT-IoT Edge-IIoT
Hazman et al. [5]	2022	AdaBoost	Feature selection methods: Boruta, mutual information, correlation	99.98% 99.99% 100%	IoT-23 BoT-IoT Edge-IIoT
Mohy-eddine et al. [26]	2022	RF	Pearson's correlation	99.12% 99.99%	Wustl-iioT-2021 BoT-IoT
Mohy-eddine et al. [4]	2023	KNN	Feature selection: PCA, GA, univariate statistical	99.99%	BoT-IoT
Ennaji et al. [34]	2023	RF, KNN, LR and MLP	5-fold cross validation	99.97% (normal activities) 99.79% (DDoS/DoS) 99.72% (Probe) 99.96% (R2L) 99.98% (U2R)	NSL-KDD

Although a great deal of research has led to the development of intrusion models based on machine learning methods, several gaps still remain. Some research, such as that by Chaabouni et al. [13], focuses mainly on isolated algorithms, often overlooking the potential benefits of ensemble learning. In addition, much research is limited to specific datasets, such as NSL-KDD, without exploring the generalization of models to other datasets as Edge-IIoT. Our model, by integrating RF, KNN and SVM algorithms into a voting classifier, overcomes these limitations by providing a robust and adaptable solution that enhances intrusion detection on both NSL-KDD and Edge-IIoT datasets.

3 Proposed Model

The purpose of the present section is to present in detail the design principles of the new model and describe the algorithms employed in its construction.

3.1 Outline of the Suggested Model

This section presents the procedures used to develop an enhanced model, aimed at boosting detection efficiency, accuracy and time to completion. An overview of model design is provided in Fig. 2.

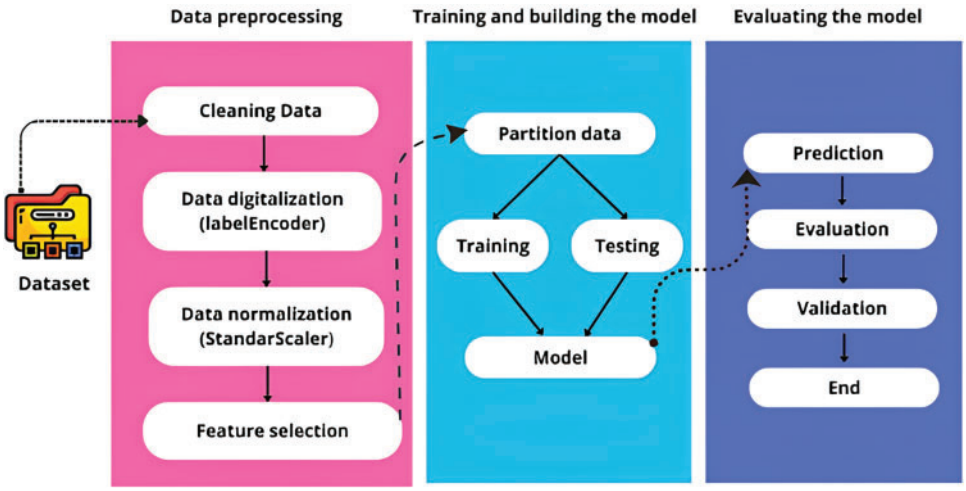

Figure 2: Architecture of our IDS approach for IoT security

Our optimized modeling approach is structured into three main steps:

- Data pre-processing: pre-processing is applied to the dataset as a whole, to eliminate unused data (NaN, etc.) and duplicates. This process includes feature selection, which involves the extraction of sub-sets of characteristics, retaining the most significant and pertinent ones, as well as eliminating others that are viewed as noisy influences. The objective is to achieve a supervised classification of records from the NSL-KDD and Edge-IIoT datasets, dividing them into two categories (Normal and Attack), while optimizing the cost and time required for learning.
- Training and building the model: the aim of this step is to reassemble the training and test datasets, and to develop a classification model using the data processed by the pre-processing subsystem.
- Evaluating the model: to assess the efficiency of the proposed model, it is necessary to calculate several performance measures, such as accuracy, precision, recall, and F1-score, from the confusion matrix.

In the NSL-KDD Test dataset, each instance is labeled to identify its class. Each instance is thus classified as normal or abnormal. To assess the performance of machine learning models, we employ cross-validation with 10 iterations. We hypothesize that the model should generalize to new data when trained on labeled data. However, validation or confirmation is required to guarantee the accuracy of its predictions. Cross-validation enables us to check whether this hypothesis is valid or not so that we can then choose the right machine-learning algorithm to perform a specific task.

3.2 Details of the Designed Model

In the present section, we examine how supervised machine learning can be employed to develop the most effective IDS.

3.2.1 Random Forest (RF)

Random Forest, a supervised learning algorithm widely adopted, is used for both classification and regression tasks. It is built on the construction of decision trees from various samples, followed by a majority decision for the classification or a mean decision concerning the results of the regression.

Each tree has a fragmented view of the problem thanks to a double random draw shown in Eq. (1) [49]:

$$RF = tree\ bagging + feature\ sampling \tag{1}$$

- Tree bagging: a random draw with replacement from the database rows (the observations).
- Feature sampling: a random draw on the database columns (the variables).

The final decision is based on the voting majority score for classifying, and the mean score for regressing.

3.2.2 Support Vector Machine

SVM stands out as a particularly useful supervised learning algorithm for both classification and regression tasks. Nevertheless, it is mainly used to explain classification difficulties.

In binary classification, the SVM's aim is to find the hyperplane that most effectively divides the examples in the two classes. This hyperplane is defined by Eq. (2) [50]:

$$\omega^{t} x + b = 0 \tag{2}$$

where ω is the weight vector usual to the hyperplane and b is the bias term.

When data are linearly separable, the optimum hyperplane will be the one maximizing the margin, defined as the distance to the nearest examples in each class. SVM optimization can be expressed as a convex optimization problem. For linearly separable data, the optimization problem shown in Eq. (3):

$$min_{\omega,b} \frac{1}{2} \|\omega\|^{2} \tag{3}$$

Under the constraints escribed in Eq. (4):

$$y_i \left(\omega^{T} x_i + b \geq 1, \forall i\right) \tag{4}$$

when the input data are non-linearly separated, SVM uses a kernel to project them into a higherdimensional space, where they can be separated by a hyperplane. The kernel can be linear, polynomial or Gaussian (RBF).

3.2.3 K-Nearest Neighbors

KNN is a distance-based ML algorithm utilized for regression and classification problems. It calculates the distance between an unlabeled point and its nearest neighbors, then identifies items with the smallest distances to define the category of the indefinite variable [35].

The KNN algorithm uses Euclidean distance to identify nearest neighbors. To find the Euclidean distance between two points x and y [51], we apply the Eq. (5):

$$d(x, y) = \sqrt{\sum_{i=1}^{N} x_i^2 - y_i^2} \tag{5}$$

with N denoting the number of features such as, $x = \{x_1, x_2, x_3, \ldots, x_N\}$ and $y = \{y_1, y_2, y_3, \ldots, y_N\}$.

3.2.4 Voting Classifier

The principle of the algorithmic voting classifier is based on combining the predictions of several machine learning models, with the aim of obtaining an optimized result. This approach involves using several models, each with its own strengths and weaknesses, to compensate for their biases and improve overall prediction accuracy. There are two types of voting classifiers: In the hard-vote classifier framework, each model contributes an equal vote to the predicted class, and the final class is determined by a majority vote as shown in Eq. (6).

$$Final\ Class = Majority\ class\ among\ \{M_1(x), M_2(x), \ldots, M_N(x)\} \tag{6}$$

when N is the number of classification models.

In the soft-vote classifier, models weight their votes according to their confidence in the prediction. Thus, the votes of the most confident models have more influence on the final decision, as described in in Eq. (7).

$$Final\ Class = \arg max_j \sum_{i=1}^{N} \omega_i P_i(j|x) \tag{7}$$

where ω_i denotes the weight assigned to the model M_i, and $P_i(j|x)$ presents the probability that the model attributes to the sample x to belong to class j.

4 Experimental Evaluation and Results

The purpose of this section is to provide a comprehensive analysis of both datasets selected and the results obtained from the model developed.

4.1 Datasets with Features Selection

NSL-KDD datasets are subdivided into three categories: numericals, nominals and binaries. Attributes in binary form occupy positions 7, 12, 14, 15, 21 and 22, while positions 2, 3 and 4 correspond to nominal attributes. All other attributes are numeric. In addition, in the Edge-IIoT dataset, various attack types are listed, including Normal, DDoS_UDP, DDoS_ICMP, Ransomware, DDoS_HTTP, SQL_injection, and others, as shown in Fig. 3. Furthermore, Fig. 4 illustrates the distribution of attacks within the dataset.

Fig. 3 displays the frequency of different types of attack detected in an Edge-IIoT environment. The horizontal axis indicates the attack label (0 for normal connections and 1 for attacks), while the vertical axis signifies the number of occurrences of each type. Attacks are classified by type, including methods such as MITM, Ransomware, SQL Injection, DoS under multiple protocols (TCP, UDP, ICMP), and others. The color and length of the bars illustrate the diversity and frequency of each type of attack, making it easy to visualize the prevalence of certain threats within the infrastructure.

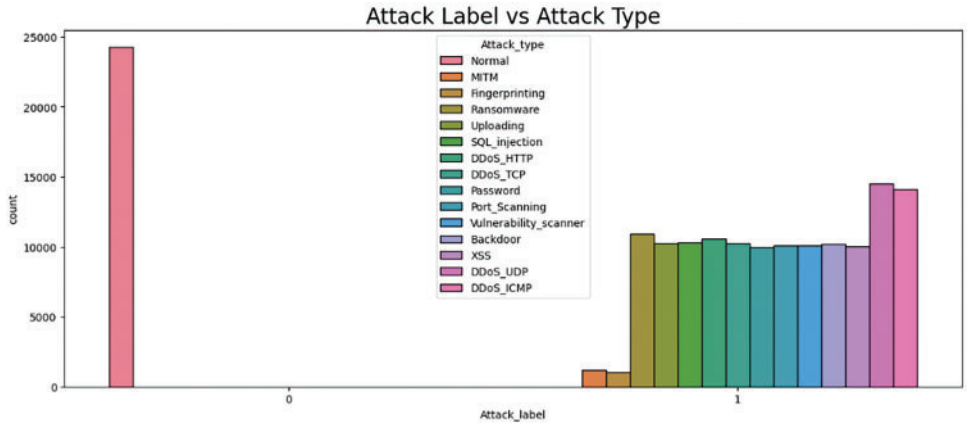

Figure 3: Attack label *vs.* attack type on Edge-IIoT set

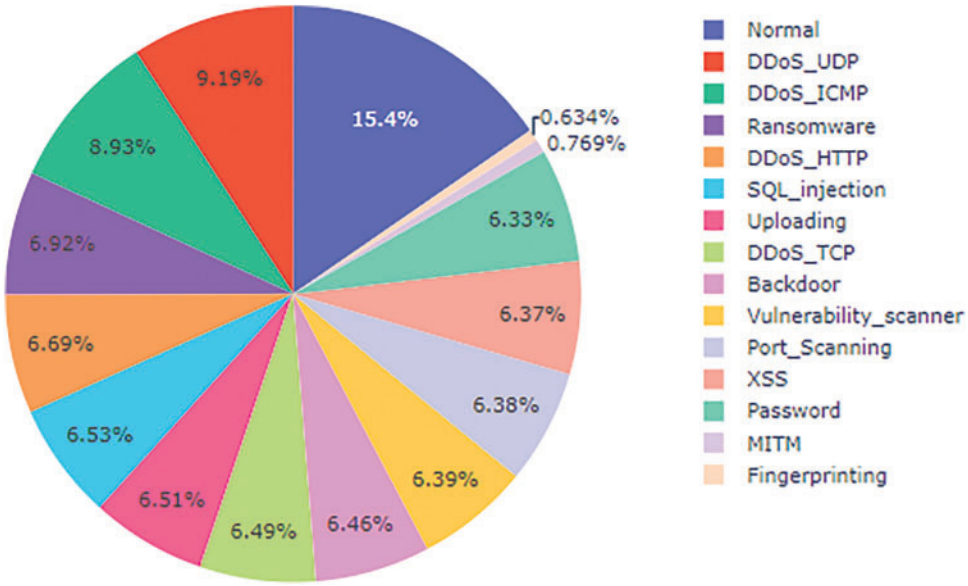

Figure 4: Distribution of attack type on Edge-IIoTset

Fig. 4 shows the percentage distribution of different traffic types in an Edge-IIoT environment. Each segment represents a specific type of traffic or attack, with a legend detailing category including Normal, DDoS (via UDP, ICMP, HTTP, TCP), Ransomware, SQL Injection, Port Scanning, Backdoor, XSS, MITM, and others. The largest proportion is occupied by "Normal" traffic (15.4%), while attacks are distributed with varying proportions, illustrating the diversity and frequency of threats in the IoT environment.

Edge-IIoT and NSL-KDD datasets first undergo pre-processing, including attribute type conversion, according to the following steps:

4.1.1 Pre-Processing for NSL-KDD Dataset

- Digitization (conversion of symbolic data into digital form): As mentioned earlier, the NSL-KDD database consists of 3 nominal attributes ("protocol_type", "service" and "flag"). Given that container network models only accept numerical attributes. There are several conversion methods, including the LabelEncoder method, which we used to digitize the nominal-type attributes of the NSL-KDD database.
  LabelEncoder is a utility class that transforms labels into numerical values, limiting them to a range from 0 to n_classes-1.
- Normalization: The results obtained after digitization are very diverse and cover a wide range of values. Some attributes have high values (such as src_bytes, dst_bytes, etc.), while others have low values (such as serror_rate, same_srvrate, etc.). This disparity can adversely affect the effectiveness of the intrusion detection model, particularly in terms of profitability. To overcome this difficulty and ensure the model's effectiveness, it is essential to adjust or denormalize the database values. In our case, the data in the two databases are normalized to the interval [0,1] using a transfer function named StandardScaler.
- Cross-validation is carried out with a constant number of 10 folds. With this configuration, each model is trained on 90% of the data, while the remaining 10% is used for testing.

In view of the size of the NSL-KDD dataset, which consists of a total of 41 features and 125,973 items for learning, and 22,544 for tests, it would be too complex to use all these features to build a classification model. This approach could significantly affect the performances of the training algorithm, especially in terms of running time and utilization of resources of the system. In addition, it is unnecessary to use all NSL-KDD attributes for the IDS to be able to classify TCP/IP connections and detect attacks. Some attributes are more important than others in this context. Hence our use of a feature selection step, namely recursive feature elimination.

4.1.2 Pre-Processing for Edge-IIoT Dataset

To make the data suitable for use in our machine learning models, we subjected it to a pre-processing process. Our aim was to optimize model accuracy and efficiency. To achieve this, we applied various pre-processing methods to Edge-IIoT datasets.

- Drop features: we have eliminated columns composed entirely of NaN values or containing only specific values, such as zeros or ones. Before this deletion, our dataset contained 63 columns and 157,800 rows. After this operation, we're left with 33 columns, all valid, while retaining the 157,800 rows.
- Encoding (Digitization): we adopted a coding approach using LabelEncoder to convert categorical variables into numerical values, making them usable in our analysis. More specifically, we transformed the "Attack_type" column into two categories: 1 for normal attacks and 0 for all other forms of attack.
- Class balancing: imbalanced datasets are defined by an irregular distribution of classes or labels, resulting in a significantly different number of instances belonging to each class. This can result in one class being over- or under-represented in the dataset. We adopt the Random Under Sampling method, eliminating examples from the majority class in order to balance the classes. However, this approach may result in the loss of information essential to the model's performance.

- Normalization: we scaled the columns to normalize them to the same amplitude, a necessary step when the ranges of values are disparate. For this, we used the StandardScaler method, which ensures consistent scaling of values.

An experimental assessment of our method has been carried out on systems equipped with IntelTMCoreTMi5 multicore processors, 12 GM RAM and 64-bit operating system. Simulations were run on Colab with Python 3, using 107 GB disk space and 12.7 GB RAM. We also employed several libraries, including Scikit-learn, Pandas, NumPy, Time and Seaborn, for implementation and results analysis.

4.2 Experimental Outcomes and Analysis

This section is split into three parts: it begins with the results obtained with the NSL-KDD dataset, followed by those from the Edge-IIoT dataset, before concluding with a comparison between these two datasets.

The confusion matrix, shown in Table 2, is used to assess the effectiveness of the intrusion detection model. By comparing model predictions with actual (labeled) data, it provides a measure of the model's accuracy.

Table 2: Confusion matrix

		Predicted	
		Normal	Attack
Actual	Normal	True_Negative (TN)	False_Positive (FP)
	Attack	False_Negative (FN)	True_Positive (TP)

Where

- True positive (TP): The test correctly detects an (-).
- False positive (FP): The test detects (+) activity as an (-).
- True negative (TN): The test correctly detects (+) activity.
- False negative (FN): The test detects an (-) as (+) activity.

To assess IDS performance, we compute various measures based on the parameters presented in Table 2.

Accuracy (Acc) represents the percentage of correct guesses completed by the model in relative to the total of calculations made, is calculated by the Eq. (8):

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{8}$$

Recall (R) evaluates the percentage of true positives, indicating the capacity of the model to correctly identify positive cases among all true positives.

$$R = \frac{TP}{FN + TP} \tag{9}$$

In Eq. (10), precision (P) represents the probability that a prediction for a given category is accurate.

$$P = \frac{TP}{TP + FP} \tag{10}$$

In Eq. (11), the F1-score (F1) expresses the harmonic mean F, which merges the two measures of recall and precision into a single number, with a range of values between 0 and 1.

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{11}$$

Referring to the confusion matrix (Table 2), it is apparent that the model correctly identified DoS/DDoS and Probe attacks, as illustrated in Figs. 5 and 6, with significant TN and TP values.

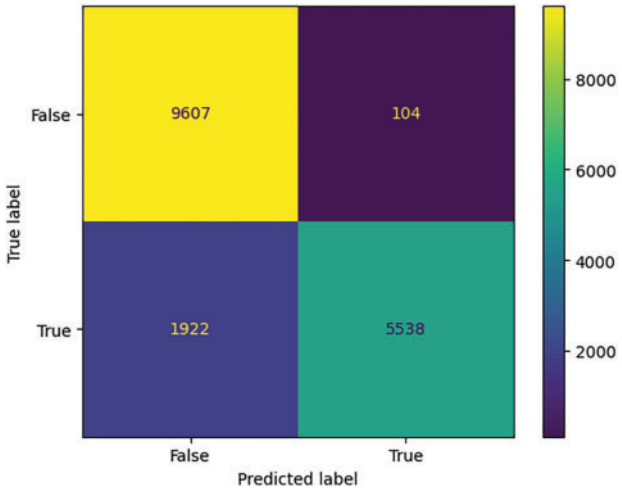

Figure 5: Confusion matrix for DoS attack using ensemble learning

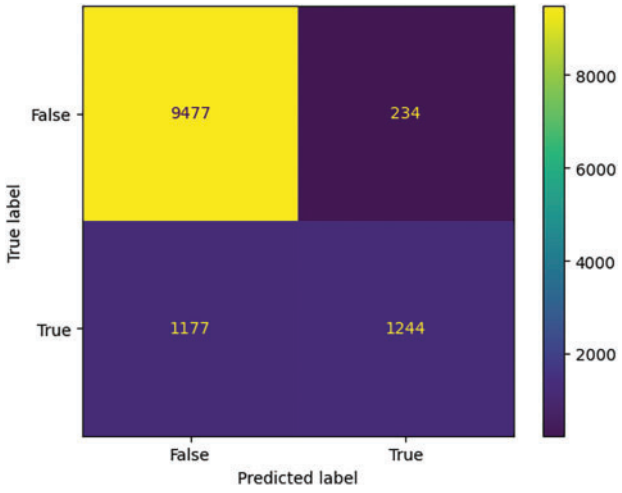

Figure 6: Confusion matrix for Probe attack using ensemble learning

On the other hand, Figs. 7 and 8 show that the model had difficulty identifying the R2L and U2R attacks, with TP values of 1 and 0, respectively.

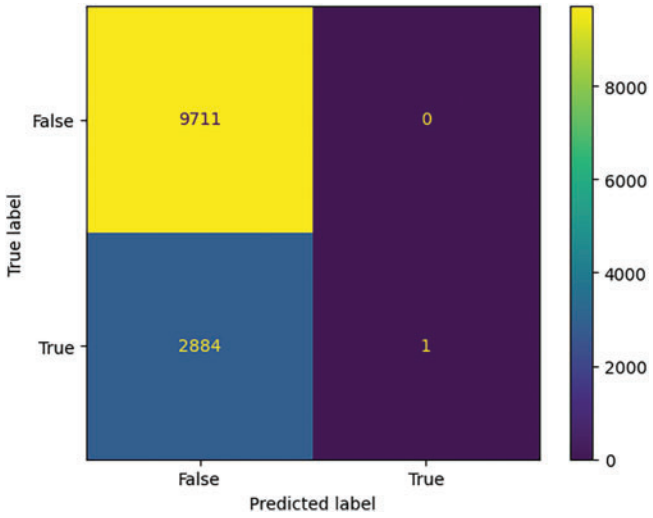

Figure 7: Confusion matrix for R2L attack using ensemble learning

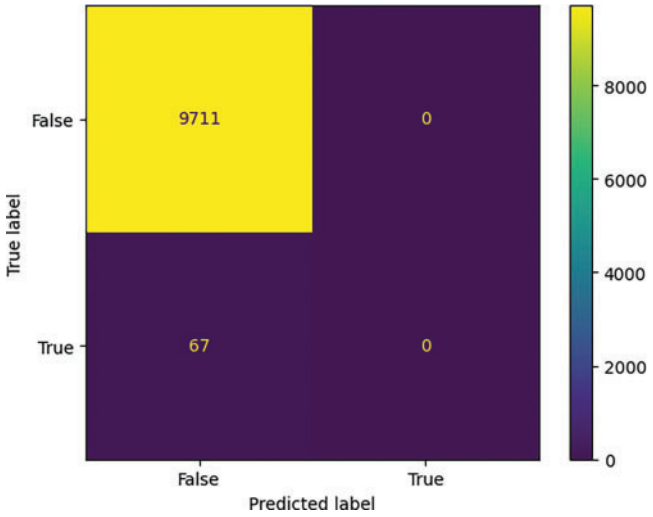

Figure 8: Confusion matrix for U2R attack using ensemble learning

By analyzing the performance of the algorithms, Table 3 presents the conditions and assumptions that influence their efficiency. Subsequently, the analysis of execution times and maximum memory usage for each algorithm, as shown in Tables 4–8, enables us to evaluate their scalability along these two dimensions. Algorithms such as Random Forest (RF) demonstrate good scalability in terms of execution time, although they may encounter limitations concerning memory usage. In contrast, the SVM and Voting Classifier algorithms, despite their low memory consumption, require longer execution times. KNN, however, stands out for its speed, but its calculation method may constrain it due to the size of the data.

Table 3: Assumption and Condition for our algorithms

Algorithm	Assumption/Condition
RF	Assumption: The performance of the RF algorithm stays satisfactory provided that the number of trees does not surpass a certain limit. Beyond this threshold, there is a significant increase in memory consumption and execution time.
SVM	Condition: SVM scalability is affected by kernel size and number of features, which requires data pre-processing (like dimension reduction).
KNN	Assumption: KNN is scalable as long as data dimensionality remains low. However, with high-dimensional data, performance can deteriorate due to the so-called "curse of dimensionality".
Voting_classifier	Condition: The scalability of the Voting Classifier is closely linked to the number of classifiers it contains. Indeed, integrating too many classifiers can lead to a considerable increase in execution time, which could limit the efficiency of the method.

Table 4: Results in % of proposed models for detecting DoS/DDos attacks using multi-class classification

Algorithms	Accuracy	Precision	Recall	F1-score	Time (s)	Memory (MiB)
RF	99.82	99.88	99.62	99.78	9.32	1480.83
KNN	99.71	99.68	99.67	99.68	15.13	0.00
SVM	99.38	99.10	99.45	99.23	75.28	0.09

Table 5: Results in % of proposed models for detecting Probe attacks using multi-class classification

Algorithms	Accuracy	Precision	Recall	F1-score	Time (s)	Memory (MiB)
RF	99.65	99.63	99.27	99.59	6.51	1481.34
KNN	99.07	98.60	98.50	98.55	8.17	0.00
SVM	98.45	96.90	98.36	97.61	31.39	0.01

Table 6: Results in % of proposed models for detecting R2L attacks using multi-class classification

Algorithms	Accuracy	Precision	Recall	F1-score	Time (s)	Memory (MiB)
RF	98.06	97.21	96.84	96.99	7.08	1482.21
KNN	96.74	95.32	95.49	95.40	7.59	0.00
SVM	96.79	94.85	96.26	95.53	115.05	0.04

Table 7: Results in % of proposed models for detecting U2R attacks using multi-class classification

Algorithms	Accuracy	Precision	Recall	F1-score	Time (s)	Memory (MiB)
RF	99.75	97.23	85.81	88.55	5.36	1481.74
KNN	99.70	93.14	85.07	87.83	4.57	0.00
SVM	99.63	91.05	82.90	84.87	7.57	0.03

Table 8: Results of performance evaluation for voting classifier

Attack	Accuracy	Precision	Recall	F1-score	Time (s)	Memory (MiB)
DoS	0.9978	0.9987	0.9970	0.9975	101.01	0.27
Probe	0.9925	0.9974	0.9890	0.9885	52.97	0.46
R2L	0.9727	0.9580	0.9631	0.9604	139.89	0.21
U2R	0.9974	0.9487	0.8662	0.8764	18.54	0.26

The data presented in Tables 4–7 show that the RF algorithm clearly outperformed the rest of the models throughout the testing dataset. Indeed, it achieved a maximum accuracy score of 99.82%, a precision level of 99.88%, a recall value of 99.62%, and an F1-score of 99.78% in the detection of DoS/DDoS attacks. What's more, when it comes to detecting probe attacks, the RF model outperforms other models on all performance indicators. It achieves 99.65% accuracy, 99.63% precision, 99.27% recall, and an F1-score of 99.59%. What's more, when it comes to detecting U2R attacks, the RF model boasts an impressive accuracy of 99.75%. However, the performance of the three models used—KNN, RF, and SVM—in terms of precision (97.23%, 91.05%, 93.14%, respectively), recall (85.81%, 82.09%, 85.07%, respectively) and F1-score (88.55%, 84.87%, 87.83%, respectively) is rather poor. On the other hand, in the case of R2L attacks, the models used offer relatively modest performance compared with other types of attack: RF has an accuracy of 98.06%, while KNN and SVM have accuracies of 96.79%. These results highlight the difficulties encountered by our model in detecting U2R and R2L attacks.

To optimize performance, EL is employed by aggregating the predictions of the Random Forest, K-Neighbors, and Support Vector Machine algorithms to achieve optimal results, as illustrated in Table 8. The ensemble approach consists in merging the different results of different machine learning models to get an optimum overall performance. In this way, the aim is to combine several machines, with their own strengths and weaknesses, to compensate for each other's biases and achieve a more reliable prediction. For our purpose, we are using a Voting Classifier, a method similar to the Bagging Classifier.

To evaluate IDS performances with Edge-IIoT dataset, various metrics are calculated using RF, KNN and SVM algorithms, as shown in Figs. 9–12, The RF algorithm showed a low FP and FN rate, at 0.88% and 5.76%, respectively, as well as optimal performance in terms of TP. In contrast, the SVM algorithm recorded a high FN rate, reaching 26.33%, suggesting that the classifier struggled to classify attacks correctly. On the other hand, the KNN algorithm achieved similar results to the Voting Classifier, with TP rates of 59.73% and 59.84%, respectively.

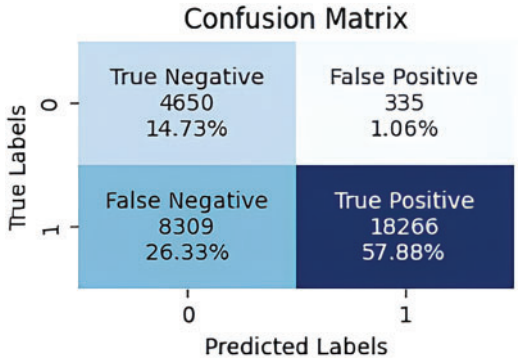

Figure 9: Confusion matrix for SVM classifier on Edge-IIoT dataset

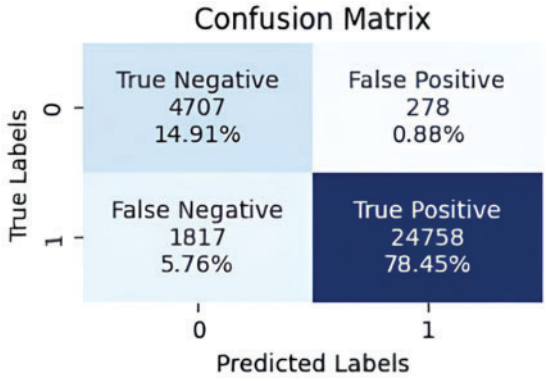

Figure 10: Confusion matrix for RF classifier on Edge-IIoT dataset

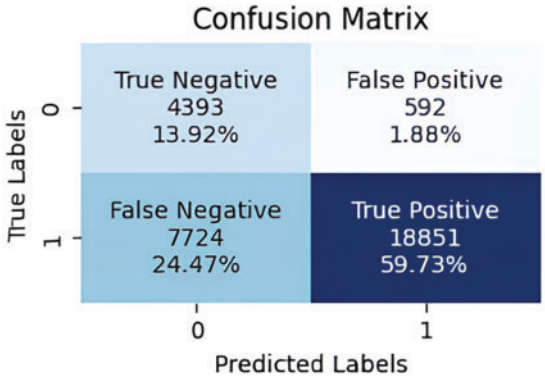

Figure 11: Confusion matrix for KNN classifier on Edge-IIoT dataset

In order to evaluate the performance of our model on the Edge-IIoT dataset, we chose the optimal parameters, as revealed in Table 9, using the GridSearchCV method to optimize the hyperparameters. This process aims to maximize the efficiency of the algorithm by identifying the optimal values for each key parameter. Fig. 13 demonstrates that the optimized parameters delivered an average cross-validation precision of 94.66%, with a standard deviation of 1.87%, and an accuracy of 93.36%, testifying to the stability of the RF algorithm.

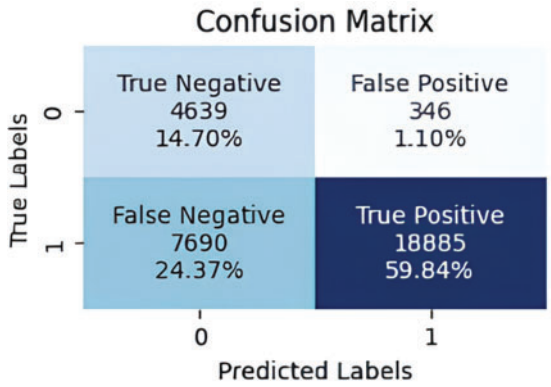

Figure 12: Confusion matrix for Voting classifier on Edge-IIoT dataset

Table 9: Parameter selection and optimization for our algorithms

Algorithm	Parameters tested	Best parameters	Method
RF	n_estimators = [100, 200, 300] max_depth = [5, 10] random_state = 42 cross_validation = 10	n_estimators = 100 max_depth = 5 random_state = 42 cross_validation = 10	GridSearchCV
SVM	C = [0.1, 1, 10] Kernel = ['rbf', 'linear']	C = 10 Kernel = 'rbf'	GridSearchCV
KNN	n_neighbors = [5, 10, 15, 20, 30] weights= ['uniform', 'distance']	n_neighbors = 10 weights = 'uniform'	GridSearchCV

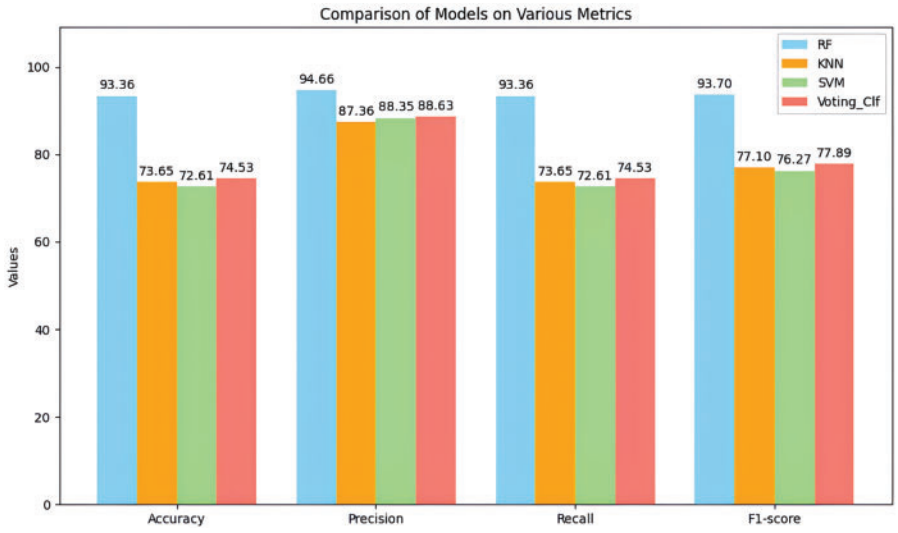

Figure 13: Comparison of performance measurements in the Edge-IIoT dataset

For the optimized SVM model, the accuracy obtained on the test set is 72.61%, with a precision of 88.35%, a recall of 72.61%, and an F1-score of 76.27%. In the case of the KNN model, after adjustment of the hyperparameters, performance on the test set translates into an accuracy of 73.65%, a precision of 87.36%, and an F1-score of 77.1%. These results highlight the crucial impact of hyperparameter optimization in improving KNN model performance for attack detection.

Lastly, the performance of the Voting Classifier indicates that it is possible to optimize results by combining the advantages of the different models, achieving a specificity of 93.06% and an overall improvement in performance.

5 Conclusion

We have developed a new IDS based on supervised classification algorithms from ML, including RF, SVM, and KNN. We have evaluated this system based on two datasets dedicated to intrusion detection: NSL-KDD and Edge-IIoT. The results indicate that our model succeeded in effectively detecting the different types of attack, with accuracy values of 99% for the NSL-KDD dataset and 93% for Edge-IIoT. This performance confirms the robustness of our approach. In our future work, we will further explore the possibilities of improving data security by exploiting the federated learning method.

Acknowledgement: None.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: The authors confirm contribution to the paper as follows: study conception and design: Hafida Assmi, Azidine Guezzaz; data collection: Hafida Assmi, Said Benkirane; analysis and interpretation of results: Nisreen Innab, Abdulatif Alabdulatif; draft manuscript preparation: Mourade Azrour, Said Jabbour. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The dataset used to support the finding of this study is available at: https://ieee-dataport.org/documents/edge-iiotset-new-comprehensive-realistic-cyber-security-dataset-iot-and-iiot-applications and http://nsl.cs.unb.ca/NSL-KDD/ (accessed on 10 Jun 2024).

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

[1] M. Douiba, S. Benkirane, A. Guezzaz, and M. Azrour, "An improved anomaly detection model for IoT security using decision tree and gradient boosting," *J. Supercomput.*, vol. 79, no. 3, pp. 3392–3411, Feb. 2023. doi: 10.1007/s11227-022-04783-y.

[2] Y. Ai, M. Peng, and K. Zhang, "Edge computing technologies for Internet of Things: A primer," *Digit. Commun. Netw.*, vol. 4, no. 2, pp. 77–86, 2018. doi: 10.1016/j.dcan.2017.07.001.

[3] U. Y. Khan and T. R. Soomro, "Applications of IoT: Mobile edge computing perspectives," in *2018 12th Int. Conf. Mathe., Actu. Sci., Comput. Sci. Stat. (MACS)*, Nov. 2018, pp. 1–7. doi: 10.1109/MACS.2018.8628388.

[4] M. Mohy-eddine, A. Guezzaz, S. Benkirane, and M. Azrour, "An intrusion detection model using election-based feature selection and K-NN," *Microprocess. Microsyst.*, vol. 4, Oct. 2023, Art. no. 104966. doi: 10.1016/j.micpro.2023.104966.

[5] C. Hazman, A. Guezzaz, S. Benkirane, and M. Azrour, "lIDS-SIoEL: Intrusion detection framework for IoT-based smart environments security using ensemble learning," *Clust. Comput.*, vol. 26, no. 6, pp. 4069–4083, Dec. 2023. doi: 10.1007/s10586-022-03810-0.

[6] M. Azrour, J. Mabrouki, A. Guezzaz, and Y. Farhaoui, "New enhanced authentication protocol for Internet of Things," *Big Data Min. Anal.*, vol. 4, no. 1, pp. 1–9, Mar. 2021. doi: 10.26599/BDMA.2020.9020010.

[7] H. Benaddi, K. Ibrahimi, and A. Benslimane, "Improving the intrusion detection system for NSL-KDD dataset based on PCA-fuzzy clustering-KNN," in *2018 6th Int. Conf. Wirel. Netw. Mobile Commun. (WINCOM)*, Oct. 2018, pp. 1–6. doi: 10.1109/WINCOM.2018.8629718.

[8] P. A. A. Resende and A. C. Drummond, "A survey of random forest based methods for intrusion detection systems," *ACM Comput. Surv.*, vol. 51, no. 3, pp. 48:1–48:36, mai 2018. doi: 10.1145/3178582.

[9] X. Fei *et al.*, "CPS data streams analytics based on machine learning for Cloud and Fog Computing: A survey," *Future Gener. Comput. Syst.*, vol. 90, pp. 435–450, Jan. 2019. doi: 10.1016/j.future.2018.06.042.

[10] Y. Meidan *et al.*, "N-BaIoT—Network-based detection of IoT botnet attacks using deep autoencoders," *IEEE Pervasive Comput.*, vol. 17, no. 3, pp. 12–22, Jul. 2018. doi: 10.1109/MPRV.2018.03367731.

[11] Y. Zeng, H. Gu, W. Wei, and Y. Guo, "Deep-full-range: A deep learning based network encrypted traffic classification and intrusion detection framework," *IEEE Access*, vol. 7, pp. 45182–45190, 2019. doi: 10.1109/ACCESS.2019.2908225.

[12] A. Guezzaz, A. Asimi, Y. Asimi, Z. Tbatous, and Y. Sadqi, "A global intrusion detection system using PcapSockS sniffer and multilayer perceptron classifier," *Int. J. Netw. Secur.*, vol. 21, no. 3, pp. 438–450, May 2019. doi: 10.6633/IJNS.20190521(3).10.

[13] N. Chaabouni, M. Mosbah, A. Zemmari, C. Sauvignac, and P. Faruki, "Network intrusion detection for IoT security based on learning techniques," *IEEE Commun. Surv. Tutor.*, vol. 21, no. 3, pp. 2671–2701, 2019. doi: 10.1109/COMST.2019.2896380.

[14] N. Chaabouni, M. Mosbah, A. Zemmari, and C. Sauvignac, "A OneM2M intrusion detection and prevention system based on edge machine learning," in *NOMS 2020-2020 IEEE/IFIP Netw. Operat. Manag. Symp.*, Apr. 2020, pp. 1–7. doi: 10.1109/NOMS47738.2020.9110473.

[15] P. Tang, W. Qiu, Z. Huang, H. Lian, and G. Liu, "Detection of SQL injection based on artificial neural network," *Knowl.-Based Syst.*, vol. 190, Feb. 2020, Art. no. 105528. doi: 10.1016/j.knosys.2020.105528.

[16] R. Wazirali, "An improved intrusion detection system based on KNN hyperparameter tuning and cross-validation," *Arab J. Sci. Eng.*, vol. 45, no. 12, pp. 10859–10873, Dec. 2020. doi: 10.1007/s13369-020-04907-7.

[17] F. Hussain, R. Hussain, S. A. Hassan, and E. Hossain, "Machine learning in IoT security: Current solutions and future challenges," *IEEE Commun. Surv. Tutor.*, vol. 22, no. 3, pp. 1686–1721, 2020. doi: 10.1109/COMST.2020.2986444.

[18] A. Thakkar and R. Lohiya, "A review of the advancement in intrusion detection datasets," *Procedia Comput. Sci.*, vol. 167, no. 1–2, pp. 636–645, Jan. 2020. doi: 10.1016/j.procs.2020.03.330.

[19] J. Gu and S. Lu, "An effective intrusion detection approach using SVM with naïve Bayes feature embedding," *Comput. Secur.*, vol. 103, Apr. 2021, Art. no. 102158. doi: 10.1016/j.cose.2020.102158.

[20] D. Jin, Y. Lu, J. Qin, Z. Cheng, and Z. Mao, "SwiftIDS: Real-time intrusion detection system based on LightGBM and parallel intrusion detection mechanism," *Comput. Secur.*, vol. 97, Oct. 2020, Art. no. 101984. doi: 10.1016/j.cose.2020.101984.

[21] A. Guezzaz, S. Benkirane, M. Azrour, and S. Khurram, "A reliable network intrusion detection approach using decision tree with enhanced data quality," *Secur. Commun. Netw.*, vol. 2021, no. 1, 2021, Art no. 1230593. doi: 10.1155/2021/1230593.

[22] I. Debicha, T. Debatty, J. -M. Dricot, and W. Mees, "Adversarial training for deep learning-based intrusion detection systems," Apr. 20, 2021. doi: 10.48550/arXiv.2104.09852.

[23] J. M. Peterson, J. L. Leevy, and T. M. Khoshgoftaar, "A review and analysis of the Bot-IoT dataset," in *2021 IEEE Int. Conf. Serv.-Orient. Syst. Eng. (SOSE)*, Aug. 2021, pp. 20–27. doi: 10.1109/SOSE52839.2021.00007.

[24] M. Shafiq, Z. Tian, A. K. Bashir, X. Du, and M. Guizani, "CorrAUC: A malicious Bot-IoT traffic detection method in IoT network using machine-learning techniques," *IEEE Internet Things J.*, vol. 8, no. 5, pp. 3242–3254, Mar. 2021. doi: 10.1109/JIOT.2020.3002255.

[25] J. L. Leevy, J. Hancock, T. M. Khoshgoftaar, and J. M. Peterson, "An easy-to-classify approach for the Bot-IoT dataset," in *2021 IEEE Third Int. Conf. Cognit. Mach. Intell. (CogMI)*, Dec. 2021, pp. 172–179. doi: 10.1109/CogMI52975.2021.00031.

[26] M. Mohy-eddine, A. Guezzaz, S. Benkirane, and M. Azrour, "An effective intrusion detection approach based on ensemble learning for IIoT edge computing," *J. Comput. Virol. Hacking Tech.*, vol. 19, no. 4, pp. 469–481, Nov. 2023. doi: 10.1007/s11416-022-00456-9.

[27] Z. Yang *et al.*, "A systematic literature review of methods and datasets for anomaly-based network intrusion detection," *Comput. Secur.*, vol. 116, May 2022, Art. no. 102675. doi: 10.1016/j.cose.2022.102675.

[28] S. Sengan, O. I. Khalaf, V. S. P, D. K. Sharma, A. J. P. L and A. A. Hamad, "Secured and privacy-based IDS for healthcare systems on E-medical data using machine learning approach," *Int. J. Reliab. Qual. E-Healthc.*, vol. 11, no. 3, pp. 1–11, Jul. 2022. doi: 10.4018/IJRQEH.289175.

[29] G. Liu, H. Zhao, F. Fan, G. Liu, Q. Xu and S. Nazir, "An enhanced intrusion detection model based on improved kNN in WSNs," *Sensors*, vol. 22, no. 4, Jan. 2022, Art. no. 1407. doi: 10.3390/s22041407.

[30] M. Asif, S. Abbas, M. A. Khan, A. Fatima, M. A. Khan and S. -W. Lee, "MapReduce based intelligent model for intrusion detection using machine learning technique," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 34, no. 10, pp. 9723–9731, Nov. 2022. doi: 10.1016/j.jksuci.2021.12.008.

[31] Y. Fu, Y. Du, Z. Cao, Q. Li, and W. Xiang, "A deep learning model for network intrusion detection with imbalanced data," *Electronics*, vol. 11, no. 6, Jan. 2022, Art. no. 898. doi: 10.3390/electronics11060898.

[32] T. Saba, A. Rehman, T. Sadad, H. Kolivand, and S. A. Bahaj, "Anomaly-based intrusion detection system for IoT networks through deep learning model," *Comput. Electr. Eng.*, vol. 99, Apr. 2022, Art. no. 107810. doi: 10.1016/j.compeleceng.2022.107810.

[33] S. Roy, J. Li, B. -J. Choi, and Y. Bai, "A lightweight supervised intrusion detection mechanism for IoT networks," *Future Gener. Comput. Syst.*, vol. 127, pp. 276–285, Feb. 2022. doi: 10.1016/j.future.2021.09.027.

[34] S. Ennaji, N. E. Akkad, and K. Haddouch, "i-2NIDS novel intelligent intrusion detection approach for a strong network security," *Int. J. Inf. Secur. Priv.*, vol. 17, no. 1, pp. 1–17, Jan. 2023. doi: 10.4018/IJISP.

[35] K. He, D. D. Kim, and M. R. Asghar, "Adversarial machine learning for network intrusion detection systems: A comprehensive survey," *IEEE Commun. Surv. Tutor.*, vol. 25, no. 1, pp. 538–566, 2023. doi: 10.1109/COMST.2022.3233793.

[36] A. Gaurav, B. B. Gupta, and P. K. Panigrahi, "A comprehensive survey on machine learning approaches for malware detection in IoT-based enterprise information system," *Enterp. Inf. Syst.*, vol. 17, no. 3, Mar. 2023, Art. no. 2023764. doi: 10.1080/17517575.2021.2023764.

[37] R. Harini, N. Maheswari, S. Ganapathy, and M. Sivagami, "An effective technique for detecting minority attacks in NIDS using deep learning and sampling approach," *Alex Eng. J.*, vol. 78, no. 1, pp. 469–482, Sep. 2023. doi: 10.1016/j.aej.2023.07.063.

[38] M. Al Lail, A. Garcia, and S. Olivo, "Machine learning for network intrusion detection—A comparative study," *Fut. Int.*, vol. 15, no. 7, Jul. 2023, Art. no. 7. doi: 10.3390/fi15070243.

[39] Y. Song, N. Luktarhan, Z. Shi, and H. Wu, "TGA: A novel network intrusion detection method based on TCN, BiGRU and attention mechanism," *Electronics*, vol. 12, no. 13, Jan. 2023, Art. no. 13. doi: 10.3390/electronics12132849.

[40] J. Vitorino, I. Praça, and E. Maia, "SoK: Realistic adversarial attacks and defenses for intelligent network intrusion detection," *Comput. Secur.*, vol. 134, Nov. 2023, Art. no. 103433. doi: 10.1016/j.cose.2023.103433.

[41] Y. Zhang and Z. Wang, "Feature engineering and model optimization based classification method for network intrusion detection," *Appl. Sci.*, vol. 13, no. 16, Jan. 2023, Art. no. 16. doi: 10.3390/app13169363.

[42] W. Yao, L. Hu, Y. Hou, and X. Li, "A lightweight intelligent network intrusion detection system using one-class autoencoder and ensemble learning for IoT," *Sensors*, vol. 23, no. 8, Jan. 2023, Art. no. 8. doi: 10.3390/s23084141.

[43] M. H. L. Louk and B. A. Tama, "Dual-IDS: A bagging-based gradient boosting decision tree model for network anomaly intrusion detection system," *Expert. Syst. Appl.*, vol. 213, Mar. 2023, Art. no. 119030. doi: 10.1016/j.eswa.2022.119030.

[44] M. Saied, S. Guirguis, and M. Madbouly, "Review of artificial intelligence for enhancing intrusion detection in the internet of things," *Eng Appl. Artif. Intell.*, vol. 127, Jan. 2024, Art. no. 107231. doi: 10.1016/j.engappai.2023.107231.

[45] Y. Akhiat, K. Touchanti, A. Zinedine, and M. Chahhou, "IDS-EFS: Ensemble feature selection-based method for intrusion detection system," *Multimed. Tools Appl.*, vol. 83, no. 5, pp. 12917–12937, Feb. 2024. doi: 10.1007/s11042-023-15977-8.

[46] M. Wang, N. Yang, Y. Guo, and N. Weng, "Learn-IDS: Bridging gaps between datasets and learning-based network intrusion detection," *Electronics*, vol. 13, no. 6, Jan. 2024, Art. no. 6. doi: 10.3390/electronics13061072.

[47] S. Biswas, and Md. S. A. Ansari, "Securing IoT networks in cloud computing environments: A real-time IDS," *J. Supercomput.*, vol. 80, no. 10, pp. 14489–14519, Jul. 2024. doi: 10.1007/s11227-024-06021-z.

[48] A. Paya, S. Arroni, V. García-Díaz, and A. Gómez, "Apollon: A robust defense system against Adversarial Machine Learning attacks in Intrusion Detection Systems," *Comput. Secur.*, vol. 136, Jan. 2024, Art. no. 103546. doi: 10.1016/j.cose.2023.103546.

[49] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, Oct. 2001. doi: 10.1023/A:1010933404324.

[50] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, Sep. 1995. doi: 10.1007/BF00994018.

[51] E. Y. Boateng, J. Otoo, and D. A. Abaye, "Basic tenets of classification algorithms K-nearest-neighbor, support vector machine, random forest and neural network: A review," *J. Data Anal. Inf. Process.*, vol. 8, no. 4, pp. 341–357, Sep. 2020. doi: 10.4236/jdaip.2020.84020.