Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 215 (2022) 742–751

Procedia Computer Science

www.elsevier.com/locate/procedia

4th International Conference on Innovative Data Communication Technology and Application

Machine Learning for IoT based networks intrusion detection: a comparative study

Marwa Baich, Touria Hamim, Nawal Sael, Yman Chemlal

Laboratory of Information Technology and Modeling, Faculty of sciences Ben M'Sik, Casablanca 7955, Morocco

Abstract

Internet of Things (IoT) refers to technologies that enable the connection of objects to the internet and also collect data with minimal human intervention. Nowadays IoT is applied to all areas, particularly in logistics, industry, and health. These new IoT applications expand the functionality of smart objects and also introduce security vulnerabilities. In this context, the development of intrusion detection systems (IDS) adopted to IoT networks has become an important field of research. Machine learning (ML) and deep learning (DL) approaches are solutions used by many researchers in IDS. In this work, we propose a state of the art on IoT network intrusion detection using ML techniques during the last few years. We aim to detect the most used and efficient machine learning techniques. To support and confirm the proposed state-of-the-art results, we carried out an experiment to compare the machine learning techniques extracted from our analysis on the same dataset. we also analyzed the performance and execution time of two feature selection techniques and a feature extraction technique for intrusion detection in the case of binary classification, and multi-classification (5 classes: Normal, DOS, Probe, U2R, and R2L), The experimental results reveal that the Decision Tree with Fisher score gave the best performance with an accuracy of 99.26% and a minimum prediction time of 0.4 seconds.

Keywords: Machine learning; Internet of Things; Intrusion Detection Systems;

10.1016/j.procs.2022.12.076

1. Introduction

Despite its strong presence in our daily lives, the IoT remains a difficult concept to understand. Indeed, it implies a profound evolution of the Internet. It is no longer a question of connecting tablets, computers, and telephones to each other, but of communicating all the elements of the physical world, abolishing the boundaries between physical objects and the virtual world in a way [1].

The huge connected object systems are faced with vulnerabilities and security issues. In addition, they are the target of several network attacks, in particular of the routing protocols. Unfortunately, using encryption and authentication is not enough to ensure these systems' security.

With the evolution of connected devices, security has become a very important area. In particular, IoT systems are directly connected to the Internet and share their data with a claimed level of trust. Thus, most cyberspace attacks are possible in IoT systems like the denial of service attack (DoS) [2]. To overcome all these challenges, an intrusion detection system (IDS) is more and more required to ensure the security of IoT systems. An IDS is a system that reacts quickly to protect sensitive data and uses Artificial Intelligence (AI) algorithms to recognize and block various attacks.

Many researchers use DL and ML techniques to develop IDS capable of detecting attacks in an IoT environment. In this paper, we first analyze the potential of various ML and DL techniques adopted in research to develop IDS. Then develop a comparative experiment to strengthen our literature review findings and compare the most used ML techniques over the same dataset.

The rest of this work is structured as follows: Section I displays the pertinent terms used in the background of this work; section II describes the methodologies proposed by researchers for IDS using DL approaches. The final section is a case study in which we used various ML techniques on the same dataset in the case of a binary-class and multi-class target, and at last, we finish with a conclusion and some perspectives.

2. IDS Background

Intrusion Detection Systems, are aimed at detecting malicious activities on computer systems. They can be either network or host-based detection systems. To counteract security risks faced by IoT devices, the IDS are adopted to continuously monitor inbound and outbound network traffic produced by IoT devices in order to detect any cyber-attacks. The two categories of this attack detection are, signature-based and anomaly-based [3].

- Signature-based method: A list of known threats and their Indicators of Compromise (IOC) is pre-programmed, a known byte sequence, a file hash, a malicious domain, or specific behavior that typically accompanies a hostile network attack can all be IOCs. An IDS with a signature-based method compares packets to a database of known IOCs or attack signatures as they pass over the network, looking for any suspicious activity.
- Anomaly-based method: In anomaly-IDS systems, the baseline of normal behavior systems is recognized using ML training. Each of the network activities is then compared to the normal behavior baseline. Anomaly-based IDS simply identifies any unusual behavior to trigger alerts.

IDS adds extra levels of security to a protected system. Administrators and managers can tune, organize, and comprehend what these information systems are saying to them, thanks to the intrusion detection system, which frequently flags issues before a loss happens. Attacks detected by IDS can be divided into four major classes: DOS, R2L, U2R, and Probes [2].

Denial of Service (Dos): A computer assault that seeks to stop a service from working preventing its legitimate users from using it. Dos can make it impossible to access a web server or prevent email from being sent across an entire company. Network traffic blockage is the direct result of these successful network attacks.

Remote To Local (R2L): In this case, the attacker will execute a remote access attack in order to acquire unauthorized access to a target system across the network.

User To Root (U2R) attack: Is typically used when legally visiting a local machine to gain root rights. The majority of these attacks are generated by programming flaws.

Probes: The idea behind a probing attack is to place a probe sufficiently close to an electronic component of the circuit in order to observe its electrical activity.

3. State of the arts

3.1. *IDS related works*

Security problems have increased due to the large growth in data transmission through various IoT devices and communication protocols, creating the need for efficient and adapted IDS. Researchers have studied different ML techniques for IDS. In this section, we analyze variant research studies developed in the last years to propose an IDS.

Numerous DL techniques were used for intrusion detection in the cybersecurity field, the authors in [1] suggested a way in which an IoT system's log information, such as an address, location, and service, is recorded in a dataset. The dataset is then preprocessed and converted into a matrix that resembles an image, which is utilized to train a convolutional neural network (CNN) with an average accuracy of 98.9%. In [2], the authors suggest a new network intrusion detection approach that makes use of CNN. The CNN model increases the accuracy of the class with relatively small numbers while simultaneously reducing the false alarm rate. A deep-learning based approach to network intrusion detection is the goal in [3], in order to distinguish between normal connections and intrusions, the system used a deep network to train itself on the patterns of anomalies. The strategy also aims to cut the false alarm rate to an absolute minimum, over the NSL-KDD dataset. The accuracy rate was 87.2%. Tsogbaatar et al. [4] projected 'DeL-IoT' as an intrusion detection framework using Software-Defined Networking (SDN). The handy features have been extracted with deep and stacked autoencoders. The projected model has exhibited higher reliability in detecting the attacks, with 99.5- 99.9% in F-score and, 91.04%-99.95% in MCC.

The Authors in [5], suggested using CNN1D, CNN2D, and CNN3D deep learning in an IoT networks anomaly detection system. The core of this research was the CNN technique. Several IoT-related IDS datasets, including BoT-IoT, IoT-DS-2, IoT-23, and MQTT-IoT-IDS2020, were used to assess the proposed IDS model. This multiclass architecture efficiently and effectively detects a variety of attacks.

Roopak et al. [6] presented an intrusion detection solution that integrates CNN and Integrating Long Short-Term Memory (LSTM) applying it to the CISIDS2017 dataset. For optimal feature selection, they used the Nondominated Sorting Genetic Algorithm optimization algorithm (NSGA). And in [7][8], the authors developed an IDS based on DL for multiclass traffic detection. In [7], the authors worked with two datasets: UNSW-NB15 and CICIDS2017, their model achieved an accuracy of 96.69% and 95.92% on average for UNSW-NB15 and CICIDS2017, respectively. The Authors in [8] evaluated the suggested model using the Bot-IoT dataset., which yielded an accuracy of 99.998% for multiclass detection. In [9], the authors proposed a new intrusion detection schema for IoT networks that apply deep learning principles to categorize traffic for binary and multi-class classification, with results close to 99% in all evaluation metrics (Accuracy, Recall, Precision, and F1-score) in binary classification. The intrusion detection model discussed in [10] for IoT network security solutions is based on a combination of DL models (RNN and CNN), the model achieves a 96.32% of accuracy, an F1 score of 95.74%, a precision of 95.43% and a recall of 96.32%. In [11], the authors developed a network intrusion detection model using convolutional neural networks (CNN-IDS), the initial traffic vector format was converted to an image format to lower the processing cost. They made use of the KDD-CUP99 dataset to assess how well the suggested CNN model performed. According to experimental findings, the model not only considerably decreased classification time but also enhanced classification performance.

Many other researchers focused on developing IDS that capitalize on machine learning methods. For example, in [12], the authors suggested using an intrusion detection system (ML-IDS) based on machine learning to identify IoT network threats over the UNSW-NB15 dataset, they obtained an accuracy of 99.9% and MCC of 99.97%.

Parth et al. [13] introduced an attack detection solution based on ML algorithms for anomaly detection on a decision module. Isolation Forest, Self-Organizing Map (SOM), One-Class Support Vector Machines (OCSVM Gaussian), and Mixture Modeling (GMM) are the components of this hybrid approach, which provides 98% accuracy. In [14], a brand-new feature selection methodology called CorrAUC based on the Area Under Curve (AUC) measure and wrapper approach was developed. C4.5, Naive Bayes (NB), RF, and SVM were then trained in a multi-class dataset. The proposed method achieved >96% on average with the Bot-IoT dataset.

In [15], the main objective was to build ML models in order to identify attacks in IoT network with the Decision Tree technique, their proposal successfully executed the MitM attack on their IoT test bed with a performance of 100% in all parameters. The authors in [15] and in [16] infer that the decision tree accuracy rate is the highest at 98,23% among all the classifier models (SVM, Naïve Bayes and Adaboost-based). The authors in [17] presented an advanced

approach for designing IDS for IoT networks with Random Forest (RF) Swarm Optimization Algorithm (PSO) classifier for feature selection, the proposed system achieves 98% accuracy for binary classification and 83% for multiclass classification on the IoTID20 dataset, The authors in [17][18] performed three key phases: preprocessing, feature selection, modeling, and evaluation for the proposed IDS.

The authors in [18] introduce a brand-new machine learning (ML) technique built on a security architecture that automatically addresses the growing security concerns associated with the IoT domain. For the purpose of reducing various vulnerabilities, this framework makes use of both Software Defined Networking (SDN) and Network Function Virtualization (NFV) enablers. The architecture incorporates an IDS for sensor data anomaly detection. Adopting One-Class, for the majority of the suggested data set combinations, SVM had a detection accuracy higher than 98%.

In [19], the authors presented an IDS for IIoT (Industrial Internet of Things) that was implemented using the Random Forest (RF) model for the GA fitness function and the Genetic Algorithm (GA) for feature selection, the results showed that, for the binary classification process, the model reached a TAC of 87.61% with 16 features, and TAC of 77.64% for the multiclass classification. With the use of 17 attributes, the model was evaluated with the UNSW-NB15 dataset.

In [20], Soe, Santosa, and Hartanto suggested an artificial neural network (ANN) to detect DDoS vulnerabilities. To address the issue of data imbalance, the Minority Oversampling Technique (SMOTE) method was used. Prior to supplying the input data to the ANN, the authors additionally performed function normalization on the data. With the Bot-IoT dataset, the final proposed technique achieved 100% precision, recall, and F1-score.

3.2. *Discussion*

The development of IDS based on various ML and DL approaches has been the main focus of research studies to address security and privacy challenges in IoT networks. Researchers have developed their proposed solutions with two approaches: framework and model. From this related works study, we found that 14 of the research studies realized model-based approaches, and 6 articles proposed frameworks for intrusion detection.

From a dataset point of view, UNSW-NB15, BOT-IOT, and NSL-KDD datasets are the most frequently used by the authors. The proposed approaches give different performances depending on the selected datasets and the input characteristics. However, using the same learning approaches and techniques does not always yield the same results for a variety of different possible attack classes. For example, using the BOT-IOT dataset, the authors in [8] found a performance accuracy of 99.97%, while using the same dataset, the authors in [20] found a 100% accuracy, and both papers used DL techniques for intrusion detection. The NSL-KDD dataset achieved 98.7% performance accuracy in [18] and 88% accuracy in [3]. Both of these papers used machine learning in their solutions. Whereas the UNSW-NB15 dataset achieved significant performance accuracy of 99.98%, in [12] using a deep learning technique and 96.99% accuracy in [19] using machine learning techniques.

In addition to the techniques cited above, variant ML techniques approaches have been adopted for intrusion detection. Fig.1 presents these variant techniques and the number of papers adopting each one. We can notice that various ML techniques are used for IDS, namely SVM, RF, DT, etc. These techniques have different performances depending on the dataset and the prediction process.

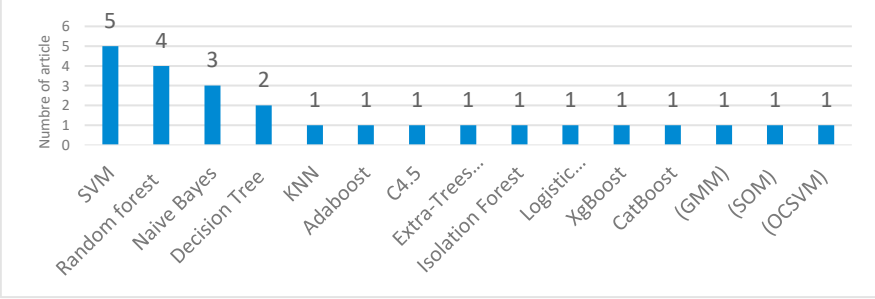

Fig1: Distribution of the ML techniques used

The statistics in figure 1 show that the most used ML techniques are SVM, Decision Tree, Random Forest, and Naive Bayes with high accuracy in all proposed models, the SVM technique obtained an accuracy of 99.98% with the Unsw-NB15 dataset in [12], and also the RF technique achieved the same accuracy with the BOT-IOT dataset in [14], and 97.49% accuracy with the NB technique. DT was the best performing technique with 100% performance in accuracy, precision, and F1-score metrics in [15] with the Sensor480 dataset. To strengthen the analysis of this research, we have developed in the following section, a comparison experiment between the most used ML techniques which are: DT, SVM, NB, and RF.

4. Experimentation

In this experiment, we compare the potentials of the most used ML techniques for intrusion detection retrieved from the literature review which are: Decision Tree (DT), Random Forest (RF), Naive Bayes (NB), and SVM. We explore these techniques combined with two feature selection methods which are Pearson correlation method and Fisher score, and PCA as feature extraction one. All these predictive models are developed to classify anomalies into two classes and then into 5 classes.

The dataset used in this research is NSL-KDD. The performance of our suggested models was evaluated using accuracy (AC), precision (PR), recall (RC) F1-Score (F1S), and Matthew's correlation coefficient (MCC) metrics.

4.1. *Dataset*

The dataset adopted in this project is NSL-KDD which has been proposed to address some of the KDD'99 dataset issues. It contains 148517 records, and each record is described with 14 features. The target classes contain 39 distinct attacks, and can be considered in two ways:

- As a binary class (attack and normal).
- As Multiclass (Normal, DoS, Probe, Privilege, and Access).

Due to the lack of publicly available datasets for IDSs based on networks and although this new version of the KDD dataset still has some of some problems raised by McHugh [21] and is not entirely representative of the actual existing networks, researchers still think it can be used as a useful benchmark data set to aid researchers in comparing various intrusion detection techniques [22].

The NSL-KDD training and test sets contain a respectable quantity of records. As a result, studies may be conducted on the entire set. As a result, the evaluation outcomes of various research projects will be similar and consistent.

4.2. *Experimentation process*

The process of our suggested study is shown in Figure 2. We first realize the dataset pre-processing to deal with missing values and reduce noise. We next apply feature selection using the Pearson correlation method, Fisher Score, and feature extraction using PCA to improve the data quality and choose the most performant features for our experiment. The modeling and the assessment are the last steps. Four ML models were applied which are Decision Tree (DT), Random Forest (RF), Naive Bayes (NB), and SVM, and evaluated by accuracy, the precision (PR), recall (RC), F1-Score (F1S), MCC, and prediction Time.

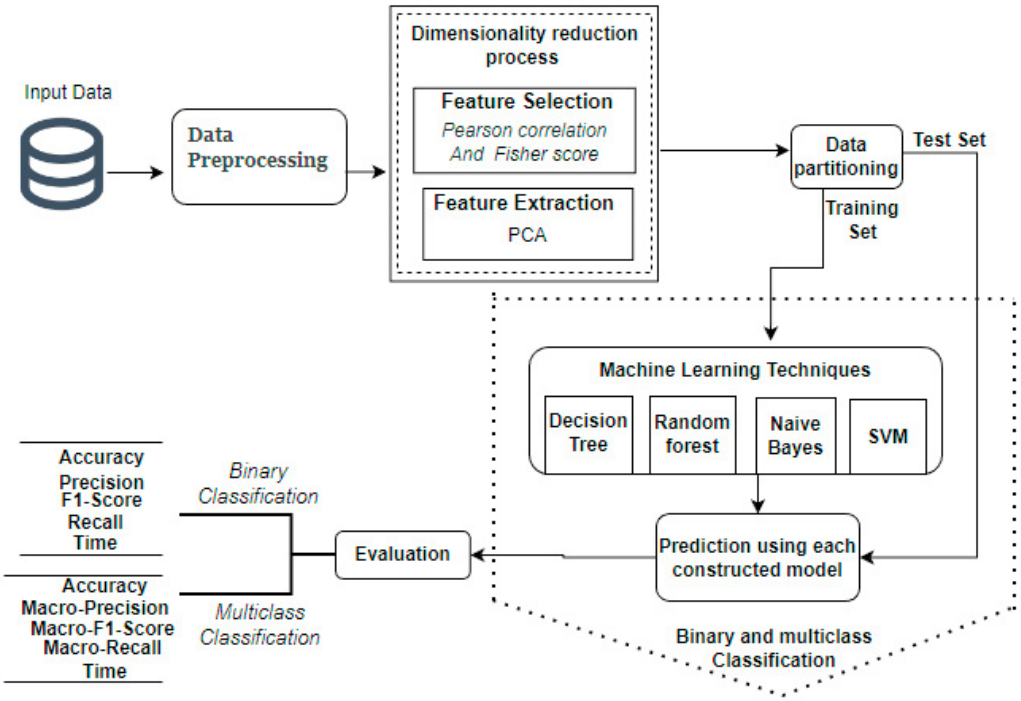

Fig 2: Experimentation process

Data Pre-processing: In ML, Data pre-processing is an essential step for cleaning and organizing data to make it suitable for building and training ML models. After fetching our dataset, we added a column that encodes the class "normal" values as 0 and all other values as 1, we used this as a target for a simple binary model that identifies either intrusion as an attack or normal access. For multi-class classification, four classes have been used out of the 39 attacks [23][24] as shown in Table 1.

Table 1: NSL-KDD dataset description

Binary class	Multi-class	NSL-KDD dataset intrusion detail
Normal	Normal	Normal access
Attacks	Denial of service (DOS)	Land, Neptune, Smurf, Worm, Teardrop, Apache2, Udpstorm, Processtable, Pod, back.
	Remote to user attacks (R2L)	Ftp_write, Snmpgetattak, Imap, Httptunnel, Sendmail, Named, Xlock, Xsnoop, Snmpguess, Spy, Wearsclient, Warezmaster, Guess_passwd, Multihop, Phf.
	User to Root (U2R)	Buffer_overflow, Perl, Rootkit, Sqlattack, Xterm, ps, Loadmodule.
	Probe	Ipsweep, Satan, Portsweep, Mscan, Nmap, Saint.

To perform the normalization task, we define the StandardScaler function. Next, we used “one-hot encoding” to deal with categorical variables.

Feature Selection (FS): After the pre-processing (one-hot encoding), the number of features in our dataset has become 128. FS allows us to choose the most relevant feature subset for building the model, either automatically or manually. In this phase, we apply and compare four ML models (DT, SVM, NB, and RF) with two feature selection techniques (Pearson correlation and Fisher score) and one feature extraction technique (PCA).

In statistics, A measurement of the linear correlation between two sets of data is the Pearson correlation coefficient

(CCP), it is always between -1 and 1 when comparing the covariance of two variables to the product of their standard deviations, hence it is a normalized measure of covariance. This measure captures the linear connection between variables. In our experiment, for binary classification, we focused on the 12 features that have a strong correlation with the target class (i.e., >0.5 in absolute value), whereas, for multiclass classification, we selected the attributes with a correlation index >0.3, and 7 features were selected, noting that this choice is due to the fact that, for multiclass case, the threshold of the correlation did not exceed 0.5. We note that the attributes ('logged_in', 'serror_rate', 'same_srv_rate', 'Count', 'dst_host_serror_rate', and 'dst_host_same_srv_rate') have been selected in both cases of feature selection.

Fisher score: among the most well-liked supervised feature selection techniques. It has the advantage of taking into consideration the classes of the target variable to give the most important features. In our experiment, we chose 22 features evaluated as important for binary classification and 44 for multiclassification. All features have an importance >=0.1 since the majority of features were considered less important with a threshold close to 0.

Feature Extraction: Principal Component Analysis (PCA), is used to extract information from a high-dimensional environment. It aims to remove the non-essential portions with less fluctuation in the data while retaining the portions that are crucial and have higher volatility.

Data Partitioning: The train-test splitting technique was used to calculate how well machine learning algorithms perform while making predictions on untrained data.

Model Evaluation: After the creation of our model, we evaluated its performance in order to measure the risks and compare different methods. We used the performance metrics: Accuracy, Precision, Recall, F-Score, and Matthew's correlation coefficient (MCC). These metrics are computed using the following parameters: True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) where TP is the number of correctly labelled positive samples, FP is the number of negative samples incorrectly labelled as positive, TN is the number of correctly labelled negative samples, and FN is the number of positive samples incorrectly labelled as negative.

Accuracy is a significant metric because it shows the proportion of accurate predictions, it measures the rate of correct predictions on all the observations.

$$Accuracy = \frac{(TP + TN)}{\left(TP + TN + FN + FP\right)} \tag{1}$$

Precision: tells how precise/accurate the model is: out of those predicted positive, how many of them are actual positive.

$$\Pr ecision = \frac{TP}{\left(TP + FP\right)} \tag{2}$$

Recall, tells us how many positives our model correctly predicted through the total actual positive.

$$\mathrm{Re}\, ccal = \frac{TP}{\left(TP + FN\right)} \tag{3}$$

F1-Score: Precision and recall are delicately combined in the F1-Score. Because the quantity of true negatives (TN) is ignored, it is intriguing and even more intriguing than accuracy. Additionally, in unbalanced class scenarios, the preponderance of true negatives drastically alters our view of the algorithm's performance. A significant number of true negatives (TN) will not affect the F1-Score.

$$F1Score = 2 \times \frac{(\mathrm{Re}\,call \times \mathrm{Pr}\,ecision)}{(\mathrm{Re}\,call + \mathrm{Pr}\,ecision)} \tag{4}$$

We may measure the effectiveness of a classification model using the Matthews correlation coefficient (MCC). This metric is especially helpful when the two classes are imbalanced, meaning that one class appears substantially more frequently than the other.

$$MCC = \frac{(TN \times TP - FN \times FP)}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{5}$$

4.3. *Results and discussion*

In this experiment, we put two classification methods into practice. First, we used a binary classification algorithm with a binary target (Normal or Attack) with and without feature selection. Table 2 shows the performance of ML algorithms for binary classification. We notice that the DT algorithm performs better with an accuracy of 99.45% without performing feature selection, and the NB algorithm gave the lowest accuracy (81.39%). With the use of "Pearson" feature selection, the RF algorithm obtained the greatest accuracy of 93.19% and a precision of 93.06% using 12 features but with a high execution time of 72.8 seconds. If we take into consideration the execution time, the DT algorithm is the best with an accuracy of 92.77% and a prediction time of 0.1 seconds. With the use of PCA, the RF algorithm has a high accuracy of 99.16%. With feature selection 'Fisher score', DT obtained a performance greater than 99% in all performance metrics.

Table 2: Binary Classification results

Algorithm	Features	Accuracy	Precision	F-score	Recall	MCC	Time (s)
DT	All	0,9945	0,9944	0,9943	0,9943	0,9890	1,5
SVM		0,9560	0,9719	0,9538	0,9363	0,9124	773,7
NB		0,8139	0,9930	0,7637	0,6204	0,6704	0,6
RF		0,9560	0,9719	0,9538	0,9363	0,9124	147,3
DT	Pearson	0,9277	0,9282	0,9252	0,9222	0,8553	0,1
SVM	(12 features)	0,8778	0,8816	0,8726	0,8639	0,7554	167,6
NB		0,8555	0,9391	0,8343	0,7505	0,7230	0,1
RF		0,9319	0,9306	0,9297	0,9288	0,8637	72,8
DT	Fisher Score	0,9926	0,9923	0,9923	0,9924	0,9851	0,4
SVM	(22 features)	0,9212	0,9426	0,9164	0,8917	0,8432	228,1
NB		0,8789	0,9317	0,8662	0,8094	0,7628	0,1
RF		0,9212	0,9426	0,9164	0,8917	0,8432	76,8
DT	PCA	0,9875	0,9878	0,9871	0,9863	0,9750	4,7
SVM		0,9242	0,9485	0,9194	0,8920	0,8494	390,2
NB		0,8847	0,9148	0,8760	0,8404	0,7710	0,1
RF		0,9916	0.9932	0,9913	0,9894	0,9832	485,35

We used the multiclass classification procedure in the second step, the process detects the types of anomalies according to the 5 classes: Normal, DOS, Probe, U2R, and R2L. The experimental findings showed that the DT was the best model without the use of features selection with 99.43% accuracy as shown in Table 3, and 92.33% with 7

features and an execution time of 0.1 seconds.

Table 3: Multi-class Classification prediction with and without feature selection process

Algorithm	Features	Accuracy	Precision	F-score	Recall	MCC	Time (s)
DT	All	0,9943	0,8853	0,8928	0,9015	0,9904	1,4
SVM		0,9778	0,9047	0,8737	0,8480	0,9625	199,2
NB		0,7976	0,6089	0,5162	0,7201	0,6686	0,6
RF		0,9778	0,9047	0,8737	0,8480	0,9625	124,9
DT	Pearson (7 features)	0,9233	0,6755	0,6476	0,6337	0,8704	0,1
SVM		0,8590	0,5018	0,4956	0,4919	0,7566	167,6
NB		0,7173	0,5213	0,4712	0,6065	0,6008	0,1
RF		0,8590	0,5018	0,4956	0,4919	0,7566	72,8
DT	Fisher Score (44 features)	0,9926	0,8262	0,8307	0,8360	0,9875	0,5
SVM		0,9294	0,5659	0,5645	0,5642	0,8803	148,4
NB		0,7856	0,6013	0,5583	0,7079	0,6905	0,1
RF		0,9294	0,5659	0,5645	0,5642	0,8803	87,6
DT	PCA	0,9865	0,8456	0,8507	0,8577	0,9772	6
SVM		0,9281	0,9023	0,6542	0,6101	0,8776	554,1
NB		0,5617	0,4558	0,4330	0,6201	0,3984	0,2
RF		0,9913	0,9278	0,8951	0,8709	0,9853	608

RF was the most efficient algorithm with an accuracy of 99.13% using the PCA method but with a long execution time of 608 seconds. Thus, the DT algorithm remains the most efficient with an execution time of 6 seconds and an accuracy of 98.65%. With the use of the "Fisher score", the DT algorithm also produced a high performance of 99.26% compared to other algorithms. The experiment results show also that the algorithm SVM has a higher execution time in both classification processes. In addition, feature selection did not improve our model performance in this experiment but it has a positive impact on response time. For example, in binary classification, DT has an accuracy of 99.45% with a time of 1.5 seconds without the use of feature selection, when using Pearson feature selection, DT achieved 92.77% accuracy with a response time of 0.1 seconds, same for the multi-class classification.

5. Conclusion

In this article, we carried out a comparison of ML approaches for intrusion detection for the IoT network, we first analysed twenty articles, compared ML learning techniques, datasets, feature engineering techniques used, and performance indicators. We found that CNN, SVM, and DT are the most commonly used for attack detection, while DT has the best performance over the Sensor480 dataset. In addition to state-of-the-art analysis, we realized an experiment to compare the most used ML techniques which are Decision Tree, Random Forest, Naive Bayes, and SVM. We explored the potential of these four techniques associated with two feature selection techniques (Pearson correlation and Fisher score) and a feature extraction technique (PCA) to improve the modelling performance. The main objective of this experiment is, first to detect whether an attack is malicious or benign (binary classification), and also to detect the type of attack, whether it is a Dos, Probe, U2R, or even R2L attack (multiclass classification). We found that the Decision Tree algorithm performs well in binary and multiclass classification.

In our future works, we intend to expand our experiment to study the most performant Deep learning techniques adopted for IDS. Our objective is to develop an IDS that overcomes the actual challenges of these systems and assures a better performance.

References

[1] Van Huong, P., & Hung, D. V. (2019, December). Intrusion detection in IoT systems based on deep learning using convolutional neural network. In 2019 6th NAFOSTED Conference on Information and Computer Science (NICS) (pp. 448-453). IEEE.

[2] Wu, K., Chen, Z., & Li, W. (2018). A novel intrusion detection model for a massive network using convolutional neural networks. Ieee Access, 6, 50850-50859.

[3] Gurung, S., Ghose, M. K., & Subedi, A. (2019). Deep learning approach on network intrusion detection system using NSL-KDD dataset. International Journal of Computer Network and Information Security, 11(3), 8-14.

[4] Tsogbaatar, E., Bhuyan, M. H., Taenaka, Y., Fall, D., Gonchigsumlaa, K., Elmroth, E., & Kadobayashi, Y. (2021). DeL-IoT: A deep ensemble learning approach to uncover anomalies in IoT. Internet of Things, 14, 100391.

[5] Ullah, I., & Mahmoud, Q. H. (2021). Design and development of a deep learning-based model for anomaly detection in IoT networks. IEEE Access, 9, 103906-103926.

[6] Roopak, M., Tian, G. Y., & Chambers, J. (2020, January). An intrusion detection system against ddos attacks in iot networks. In 2020 10th annual computing and communication workshop and conference (CCWC) (pp. 0562-0567). IEEE.

[7] Le, K. H., Nguyen, M. H., Tran, T. D., & Tran, N. D. (2022). IMIDS: An intelligent intrusion detection system against cyber threats in IoT. Electronics, 11(4), 524.

[8] Derhab, A., Aldweesh, A., Emam, A. Z., & Khan, F. A. (2020). Intrusion detection system for internet of things based on temporal convolution neural network and efficient feature engineering. Wireless Communications and Mobile Computing, 2020.

[9] Ge, M., Fu, X., Syed, N., Baig, Z., Teo, G., & Robles-Kelly, A. (2019, December). Deep learning-based intrusion detection for IoT networks. In 2019 IEEE 24th pacific rim international symposium on dependable computing (PRDC) (pp. 256-25609). IEEE.

[10] Lopez-Martin, M., Carro, B., Sanchez-Esguevillas, A., & Lloret, J. (2017). Network traffic classifier with convolutional and recurrent neural networks for Internet of Things. IEEE access, 5, 18042-18050.

[11] Yihan, X. I. A. O., Cheng, X. I. N. G., Taining, Z. H. A. N. G., & Zhongkai, Z. H. A. O. (2019). An intrusion detection model based on feature reduction and convolutional neural networks [J]. IEEE Access, 7, 42210-42219.

[12] Saheed, Y. K., Abiodun, A. I., Misra, S., Holone, M. K., & Colomo-Palacios, R. (2022). A machine learning-based intrusion detection for detecting internet of things network attacks. Alexandria Engineering Journal, 61(12), 9395-9409.

[13] Bhatt, P., & Morais, A. (2018, December). HADS: Hybrid anomaly detection system for IoT environments. In 2018 international conference on internet of things, embedded systems and communications (IINTEC) (pp. 191-196). IEEE.

[14] Shafiq, M., Tian, Z., Bashir, A. K., Du, X., & Guizani, M. (2020). CorrAUC: a malicious bot-IoT traffic detection method in IoT network using machine-learning techniques. IEEE Internet of Things Journal, 8(5), 3242-3254.

[15] Kiran, K. S., Devisetty, R. K., Kalyan, N. P., Mukundini, K., & Karthi, R. (2020). Building a intrusion detection system for IoT environment using machine learning techniques. Procedia Computer Science, 171, 2372-2379.

[16] Gaber, T., El-Ghamry, A., & Hassanien, A. E. (2022). Injection attack detection using machine learning for smart IoT applications. Physical Communication, 52, 101685.

[17] Sarwar, A., Hasan, S., Khan, W. U., Ahmed, S., & Marwat, S. N. K. (2022, March). Design of an Advance Intrusion Detection System for IoT Networks. In 2022 2nd International Conference on Artificial Intelligence (ICAI) (pp. 46-51). IEEE.

[18] Bagaa, M., Taleb, T., Bernabe, J. B., & Skarmeta, A. (2020). A machine learning security framework for iot systems. IEEE Access, 8, 114066-114077.

[19] Kasongo, S. M. (2021). An advanced intrusion detection system for IIoT based on GA and tree-based algorithms. IEEE Access, 9, 113199-113212.

[20] Soe, Y. N., Santosa, P. I., & Hartanto, R. (2019, October). Ddos attack detection based on simple ann with smote for iot environment. In 2019 fourth international conference on informatics and computing (ICIC) (pp. 1-5). IEEE.

[21] McHugh, J. (2000). Testing intrusion detection systems: a critique of the 1998 and 1999 darpa intrusion detection system evaluations as performed by lincoln laboratory. ACM Transactions on Information and System Security (TISSEC), 3(4), 262-294.

[22] Nsl kdd. https://www.unb.ca/cic/datasets/nsl.html. last accessed

[23] Kumar, V., Chauhan, H., & Panwar, D. (2013). K-means clustering approach to analyze NSL-KDD intrusion detection dataset. International Journal of Soft Computing and Engineering (IJSCE), 3(4), 1-4.

[24] Wu, T., Fan, H., Zhu, H., You, C., Zhou, H., & Huang, X. (2022). Intrusion detection system combined enhanced random forest with SMOTE algorithm. EURASIP Journal on Advances in Signal Processing, 2022(1), 1-20.