Intrusion Detection in IoT Environment Using Hyperparameters Tuned Machine and Deep Learning Models on the CICIoT2023 Dataset

Hajar Fares[1], Noura Aknin[1], Ghita Lazrek2, Mustapha Zeroual[3]
[1]TIMS Lab, Fs, Abdelmalek Essaadi University, Tetouan, 93000
[2]Lab Liasse, Ensa, Sidi Mohamed Ben Abdellah University, Fez, Morocco
[3]STIC Lab, FSJ, Chouaib Doukkali University, El Jadida, 24000
E-mail: Hajar.fares@etu.uae.ac.ma, Noura.aknin@uae.ac.ma, ghita.lazrek@usmba.ac.ma, zeroual.mustapha@ucd.ac.ma

Keywords: internet of things, intrusion detection, machine learning, deep learning, SVM, random forest, VGGNet, data balancing, feature engineering

Received: April 11, 2025

Internet of Things (IoT) technology has made our life connected, simple and smart by integrating physical objects to the internet in various fields. These are also systems capable of creating and transmitting data to users through various services. Since these objects with their limited resources are interconnected with each other via the internet, they are then vulnerables to many attacks. Given the constraints already mentioned, traditional intrusion detection systems (IDS) are inadequate and no longer sufficient. In this paper, we propose an intrusion detection taking on consideration the limited resources of IoT devices, using machine learning and deep learning combined with features engineering, data balancing method and hyperparameters tuning to achieve the best result. Using a wide range of evaluation metrics, including Accuracy, Precision, Recall, F1-score, confusion matrix and execution time, we have evaluated various machine and deep learning models including Support Vector Machine, Random Forest, VGGNet and Deep Neural Network, as well as an approach for features extraction such as features scaling and transformation. This study is carried out using a well-known, benchmark and real time dataset CICIoT2023, generated by IoT devices that includes thirty-three attacks, classified into seven categories, namely DDoS, DoS, Recon, Web-based, Brute Force, Spoofing, and Mirai.
The experiment result demonstrates the effectiveness of Random Forest that accomplished with 91,89% in the accuracy and 92% in the precision, outperformed the others modeles in classifying attacks from normal traffic with a minimum time of execution.

Povzetek: Za IoT varnost je narejena primerjava ML/DL (RF, SVM, DNN, VGGNet) s uravnoteženjem in uglaševanjem na CICIoT2023; RF doseže 91.9% točnost ter 92% natančnost ob hitrem izvajanju.

1 Introduction

Internet of Things (IoT) is a technology that allows connecting a set of objects to each other and to the Internet. IoT is defined as a network of wired and wireless communication technologies [1]. The Internet of Things can monitor and manage a variety of resources in real time by combining sensors, RFID, GPS, and other technologies [2].

It is interconnected and distributed that communicate through embedded systems. IoT devices gathers a large amount of data from sensors nodes, which will subsequently processed. The connectivity is among their enormous potential that could positively affect our lives in different application areas as illustrated in Figure 1.

Healthcare: Since IoT sensors have been used in healthcare by collecting and analyzing medical information in real time, the intervention in the right time have been increased [3].
Transport: One of the most important tasks for every individual and business is transportation, and the Internet of Things has made the system smarter, particularly with the growing market share of electric vehicles.t [4].
Smart-city: various public systems and services, such as parking, garbage management, and street lighting can be improved using IoT technologies [5].
Agriculture: IoT technology helps ensure product quality and end customer satisfaction, to avoid future food resource shortages that could be brought on by several circumstances [6].

IoT devices are low-cost due to their limited storage capacity and small processing and communication resources that are diverse in that they use various hardware and software platforms.

Given that IoT devices are numerous and often deployed in low-traffic areas, it becomes vulnerable easy to compromise by various attacks types and such as conventional threat, routing attacks, Man-In-Middle and DoS attacks and becomes difficult to track and protect [7][8]. Data confidentiality, authentication and access control inside the IoT network, are some security and privacy rules that have been proposed by researchers. IoT networks are nonetheless susceptible to several assaults meant to take down the network, even with various defenses in place. This calls for the necessity for an additional line of defense that is intended to identify potential threats intrusion. To do this, detection systems, or IDSs, are used to analyze the traffic in real-time and identify abnormal behaviors, it can be implemented in hardware or software to automate the intrusion detection process [9]. Given the increasing complexity, automation, and distribution of attacks, traditional IDSs like Snort struggle to address current network security challenges and it can fail to detect unknown attacks or analyze encrypted traffic especially with the evolving cyber threats and large-scale networks. Despite the effort done by an IDS to improve IoT security, is still considering inadequate for an IoT technology, due to many constraints, such as capacity processing and storage of nodes and also the architecture of the network based on multi-hop. Recently, with the progress of artificial intelligence, specifically machine learning, deep learning and Federated learning [10], security can be improved. Machine Learning (ML) techniques offer real-time solutions that can maximize resource utilization in the network, thereby extending its operational lifetime [11]. Other studies [12] have addressed the potential of Deep Reinforcement Learning.

The aim of this research is to find robust security solutions against vulnerabilities and the different types of attacks that threaten IoT, taken in considerations the constraints and challenges of IoT devices.

After analyzing the different solutions provided in previous works, it was concluded that because of the features of IoT, which need the use of more sophisticated methods like machine learning, the security mechanisms of conventional computer systems are ineffective for the security of intelligent systems.

Before starting this study, various Research Questions have been take in consideration:

- Can machine-learning models effectively detect and classify multiple IoT-based cyberattacks using the CICIoT2023 dataset?
- Does balancing the dataset improve classification performance across minority attack classes?
- Does hyperparameters-tuned models will significantly outperform untuned models in terms of accuracy, precision, recall, and F1-score on the CICIoT2023 dataset?

To answer those research questions, this paper introduces a novel framework, providing a new insight into IoT security.

The main contributions of this study can be summarized as follows:

- Providing comprehensive details of IoT technology, limitations, and vulnerabilities to attacks.
- Applying and analyzing the performance of various machine and deep learning models, for detecting intrusions, using a real-time dataset with huge amount of data and various types of attacks.
- Finding the best hyperparameters tuning to optimize the modeles used to achieve the best accuracy of classification.
- Offering a thorough discussion of experiment results, providing a summary of finding, indicating advantages and limitations of our proposed model in comparison with the literature review.

The current research paper is structured as follows: After a general introduction about IoT, section 2, present many recent research studies that have been published, and have as objective IoT security using machine learning based approaches. Followed by section 3, which detailed our methodology of contribution. While section 4, provide and discuss the obtained results. Finally, section 5, conclude our paper, with some suggestions as upcoming researchers.

2 Related works

Several researches have been published in recent years, providing several approaches for IoT security. This section present the latest research in intrusion detection in IoT using different methods.

The study contributed by (Amanullah, M et al. 2020) [13] have carried out an extensive study on the newest developments in big data, IoT security, and deep learning. Additionally, a comparison study and the connections between big data, IoT security, and deep learning technologies have been covered as well as some challenges relied in applying Deep learning in IoT.

A learning-based methodology, introduced by (Islam, N et al. 2021) [14] is designed to recognize and ensure the security of the infrastructure. Three ML classifiers and five DL models have been utilized including Decision Tree, Random Forest, e deep belief network, LSTM and Bi-LSTM have been evaluated using four datasets: NSL-KDD, IoTDevNet, IoTID20, and IoT Botnet. Relying on

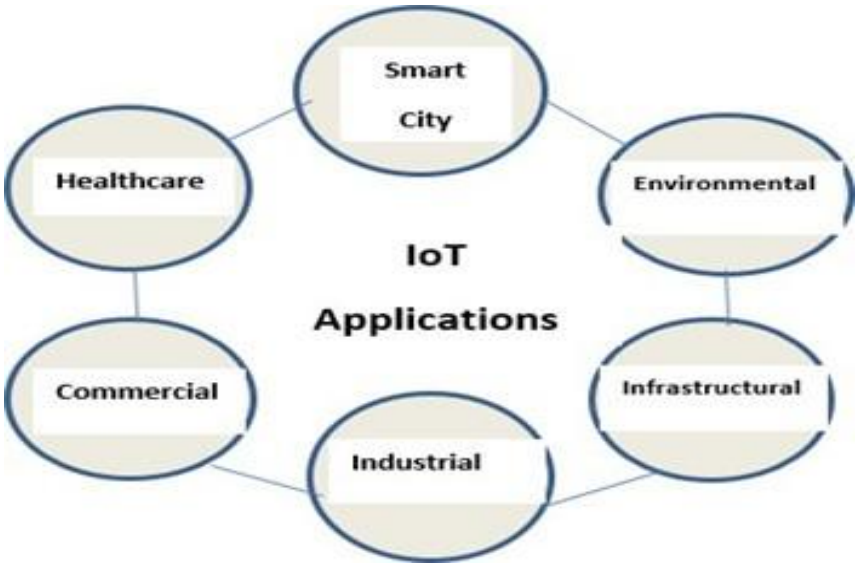

Figure 1: IoT applications

the experimental, Bi-LSTM outperforms the others modeles. However, this research study does not delve deeply into the feature selection process or the representation of data used in training the models, as well as these datasets used may not fully represent the diverse and dynamic nature of real-world IoT networks

Author (Lin, T. 2020) [15] suggest a deep learning (recurrent neural network)-based retrieval technique to aid in the more effective analysis of IoT data. In addition, a study on data retrieval methods to prevent adversarial hacking in adversary machine learning domains is presented in this work. However, Implementing DL algorithms can increase resource computational and energy consumption. The study does not explore methods to mitigate these issues.

This article addresses how the Internet of Things (IoT) can detect and mitigate cyberattacks, encrypt edge data transfer, and synthesizes previous studies to investigate the potential of deep learning (DL) in improving the security architecture of IoT. Additionally included in this article is security research in application domains such Internet of Vehicles, Industrial IoT, Smart Grid, Smart Home, and Smart Medical (Li, Y et al. (2021)) [16]. However, the paper does not thoroughly explore how the proposed DL techniques can scale to accommodate the growing number of devices and the increasing volume of data in large-scale IoT deployments.

An innovative Deep Learning model for intelligent security risk assessment in Internet of Things. has been proposed by authors (Abbass, W et al. 2018) [17]. They assessed the correctness and performance of the recommended model. Thus, the results verify that there is a significant performance optimization when Deep Learning is used to Security Risk Assessment. However, the paper lack of detailed information about the dataset used and the preprocessing step. without transparency regarding dataset characteristics, it's challenging to assess the generalizability and robustness of the proposed models across different IoT environments.

To develop a security model for an IoT environment using the BoT-IoT dataset, (Pokhrel, S. et al., 2021) [18] used a method that combined feature engineering and data balancing based on the SMOTE technique, along with various machine learning algorithms. including KNN, Naive Bayes, and (MLP-ANN). This study doesn't used enough evaluation metrics, which might misrepresent actual model performance in real intrusion scenarios.

Machine learning models have been implemented as anomaly detection methods to identify anomalous network activity. The authors (loannou, C et al. 2019) [19] suggest using the (SVM) to identify anomalies in the Internet of Things. SVM bases the creation of its normal profile hyperplane on local sensor activity, both malicious and benign. However, this study employs binary classification, distinguishing only between normal and abnormal activities. This approach may not adequately address the complexity of real-world IoT networks, where multiple types of attacks can occur simultaneously.

This work (Chirra, D. R. (2023)) [20] investigates the use of deep learning techniques for anomaly detection to strengthen the security and privacy of IoT devices. The authors have examined a range of deep learning techniques, such as Convolutional Neural Networks (CNNs), Long Short-Term Memory (LSTM) networks, and Autoencoders, using two datasets, KDD'99 and CICIDS2017. They highlight the effectiveness of these techniques in identifying anomalies in IoT data streams. However, without real-time dataset evaluation, it's challenging to determine the model's effectiveness in detecting anomalies as they occur.

The employment of IoT security protocols is examined this study (A. Berqia and others, 2024) [21]. The proposed approach utilizes the CIC2023 IoT Dataset and several machine learning algorithms, including logistic

regression, K-Nearest Neighbors (KNN), and deep neural networks (DNN), to efficiently detect and categorize DDoS attack patterns. This work have .However, it did not extensively explore hyperparameter tuning and this lack of optimization may have resulted in suboptimal model performance.

This study (El Yamani et al. (2024)) [22] demonstrates how effectively deep learning and resampling techniques collaborate to improve IoT security. To address class imbalance in security datasets, authors have employed a CNN-LSTM hybrid model in conjunction with balanced resampling strategies. The "N-BaIoT" dataset, which contains information from IoT devices attacked by BASHLITE and Mirai botnet assaults, was used for the experiment.

An approach for identifying intrusions in IoT networks was developed by (Sarhan, M. et al. (2024) [23]. The authors have evaluated several machine learning and deep learning models, including Deep Feed Forward (DFF), CNN, RNN, DT, LR, and Naïve Bayes (NB), as well as three feature extraction algorithms: PCA, LDA, and Auto-encoder, using three benchmark datasets: UNSW-NB15, ToN-IoT, and CIC-IDS2018.

A review of the literature on IDS in the Internet of Things, is given by (Elouardi, S. et al., 2024) [24]. To find intrusions in the data, a range of IDS techniques are used, such as Machine Learning (ML), Deep Learning (DL), and Large Language Models (LLMs).

An intrusion detection method designed by (Ayyaz-ul-Haq Qureshi, B. et al (2019)) [25] using KDD dataset achieving an accuracy from 85.5% to 95.25% for RNN-IDS, surpassed the the other algorithms applied, namely, J48, SVM, NB, NB Tree, MLP, RF, RF Tree, and ANN.

Using two datasets, NSL-KDD and UNSW-15, and a number of classifiers, including RF, CNN, BiLSTM, CNN-BiLSTM, AlexNet, and LeNet-5, (Jiang, K. et al. (2020))[26] have presented an intrusion detection method. The experiment's findings show that CNN-BiLSTM fared better than other models, with accuracy rates of 83.58% and 77.16% for the datasets in question, respectively. (Dushimimana, A. et al. (2020)) [27] using 10% of KDD dataset, they have evaluated three modeles, including Bidirectional Recurrent Neural Network (Bi-RNN), RNN and Gated Recurrent Neural Network (GRNN) for detecting intrusions within an IoT environment. The experiment result demonstrate that Bi-RNN outperform the two other modeles.

LSTM, GRU, Bi-LSTM, and Broad Learning System (BLS) algorithms were applied to the NSL-KDD dataset for different known intrusion classifications (Li, Z. et al., 2018) [28]. According to the performance analysis, the BLS shortens the time needed to train the model, with an overall accuracy of 84.15% and 72.64%.

Table 1 below, provide a summary of literature review

Table 1: Summary of literature review

Reference	**Approach used**	**Dataset used**	**Limitations**
[14] (Islam, N et al. 2021)	DT, RF, SVM DBN, LSTM, Bi-LSTM	NSL-KDD IoTDevNet, IoTID20, and IoT Botnet	No data balancing have been used
[15] (Lin, T. (2020))	RNN	-	Considering one deep learning model doesn't provide a general result
[16] (Li,Y et al. (2021))	Deep Learning modeles	-	Collection of private data to carry the experiment
[17] (Abbass, W et al. 2018)	Deep Learning modeles	-	No enough information about dataset used and preprocessing
[18] (Pokhrel, S et al. (2021))	KNN NLP-ANN	BoT-IoT dataset	A single dataset are considered for performance comparison.
[19] (loannou, C et al. (2019))	SVM	-	*No variation of modeles *Lack of effeciency
[20] (Chirra, D.	CNN and LSTM	KDD'99 and	No enough

R. (2023))		CICIDS2017	learning modeles used
[21] (Berqia, A et al. (2024)).	KNN and DNN	CIC2023IoT Dataset	No enough learning modeles used
[22] (El Yamani et al.(2024))	CNN-LSTM hybrid model	N-BaIoT" dataset	Only one dataset is taken into account for comparing performance.
[23] (Sarhan, M. et al.(2024))	DFF, CNN, RNN, DT, LR, NB	UNSW-NB15 ToN-IoT CIC-IDS2018	Not mentioned
[24] (Elouardi, S. et al (2024))	CNN and LLM		Not mentioned
[25] (Ayyaz-ul-Haq Qureshi, B. et al (2019))	J48, SVM, NB, NB Tree, MLP, RF, RF Tree, ANN and RNN	NSL-KDD	When comparing performance, only one dataset is taken into consideration.
[26] (Jiang, K. et al. (2020))	RF, CNN, Bi-LSTM, CNN, BiLSTM, AlexNet	NSL-KDD, UNSW-15	No performance comparison with related studies.
[27] (Dushimimana, A. et al. (2020))	Bi-RNN, RNN, GRNN	10% of KDD dataset	*Lack of generalizability *failed to analyze the performance of the studied modeles
[28] (Li, Z. et al.(2018))	LSTM, GRU Bi-LSTM, BLS	NSL-KDD	*Contribution based only on one dataset and *No hyper-parameter tuning was done. *Low accuracy achieved

3 Materials and methods

This section introduces our research method. To evaluate the security model suggested in this article in the IoT environment, extensive experiments are conducted in this section.

The simulation environment is executed on the Ubuntu system 16 GB of ram, installed in a Virtual Box platform, on the Windows operating system. To run our model, we have used Jupyter notebook, and we have installed the necessary packages such as Pandas, Imblearn, NumPy, matplotlib and sklearn. For reasonable performance, we have chosen GPU, after installing it using the following command:

!pip install tensorflow-gpu

A series of steps was carried out, starting by the dataset selected, the preprocessing step and the features extraction, and finally, the metrics of evaluation chosen, in order to analyze rigorously the performance of the chosen models to identify intrusions in an IoT environment as illustrated in Figure 2.

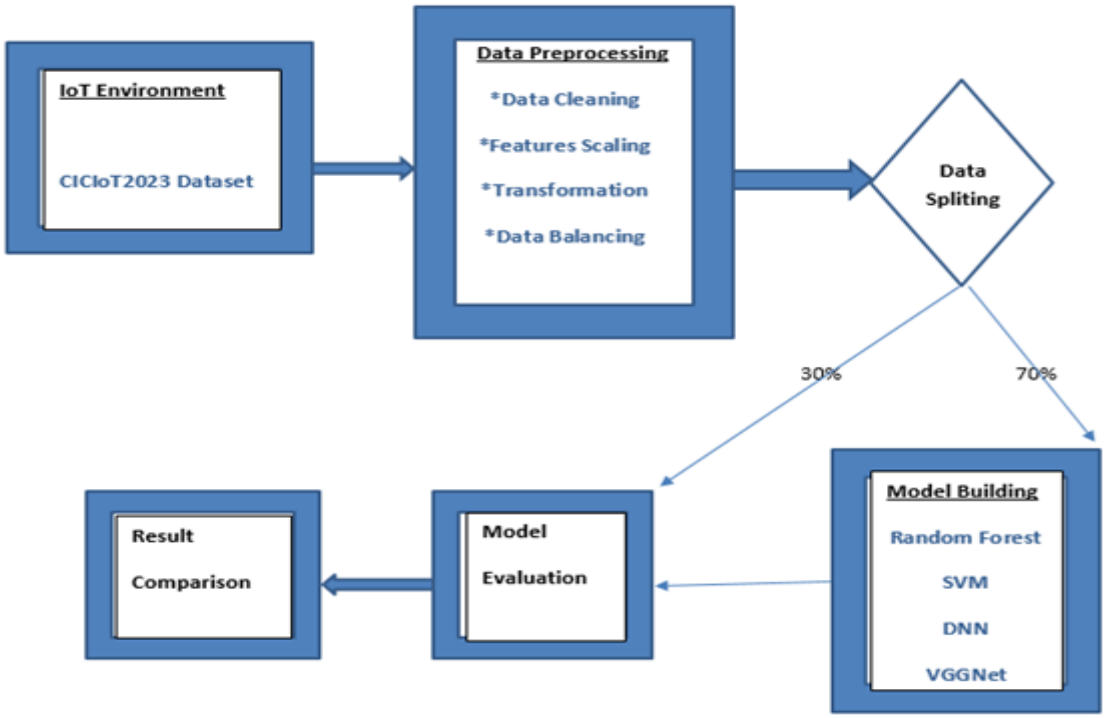

Figure 2: Proposed classifier model

3.1 Dataset selected

The completeness and quality of the datasets used for training and evaluation is greatly influence the proposed IDS. The Canadian Institute for Cybersecurity (CIC), which has a notable presence in the cybersecurity ecosystem and a track record of making significant contributions, provided the dataset utilized in this work, CICIoT2023[29].

On this IoT architecture, thirty-three different attacks is carried out as shown in (Figure 3). These assaults fall into seven categories: brute force, spoofing, Web-based, DDoS, DoS, Recon, and Mirai. The distribution details of dataset are illustrated in Table 2.

3.2 Preprocessing

The preprocessing steps significantly impact the quality of the model's input data, thereby influencing its performance. Effective data preprocessing is a crucial step in preparing the datasets for model training. The preprocessing workflow involves several key tasks:

Column name cleaning: Removing spaces from column names to ensure consistency and prevent errors during data manipulation.

Irrelevant data removal: Eliminating useless columns and rows that do not contribute to the model's predictive capability.

Handling missing values: removing missing values or incomplete rows to prevent biases and inaccuracies in the model. There are various methods used for handling missing values. In our contribution, we have used the Median imputation.

Duplicate removal: Ensuring data integrity by removing duplicate records to avoid overfitting.

Label mapping: clear label mapping ensures reproducibility and interpretability. In the CICIoT2023 dataset used, the number of classes is large, we have considered using Target Encoding to reduce dimensionality.

Data balancing: Adjusting the dataset to prevent model bias towards any particular class. This is typically achieved through techniques such as oversampling the minority class or under sampling the majority class to ensure a balanced representation of each class.

In this study, to resolve this issue of data balancing, we have used SMOTE method, which is an oversampling technique that generates synthetic data for the minority class by interpolating between existing samples and their neighbors, effectively increasing the number of minority class samples as illustrated in (Figure 4 and Figure 6).

Feature scaling and transformation: Normalizing the data make sure that every feature adds the same amount to the models learning. This often involves scaling features to a standard range, such as [0, 1], and applying transformations to stabilize variance and enhance model performance. In the study [35], authors provide a comparative analysis of various

normalization methods, while another study [36] demonstrate the effectiveness in applying z_score normalization for classification improvement. In this study, we have used the standardization method or called also (Z_Score normalization) defined by the following formula (1):

$$\text{Z\_Score} = \frac{X-u}{\sigma} \quad (1)$$

Where: X = raw feature value, μ = mean of the feature, and σ = standard deviation of the feature

Data splitting: Dividing the dataset into training and testing sets to evaluate the model's performance. In this contribution, we have used 70% of data for training and 30% for testing.

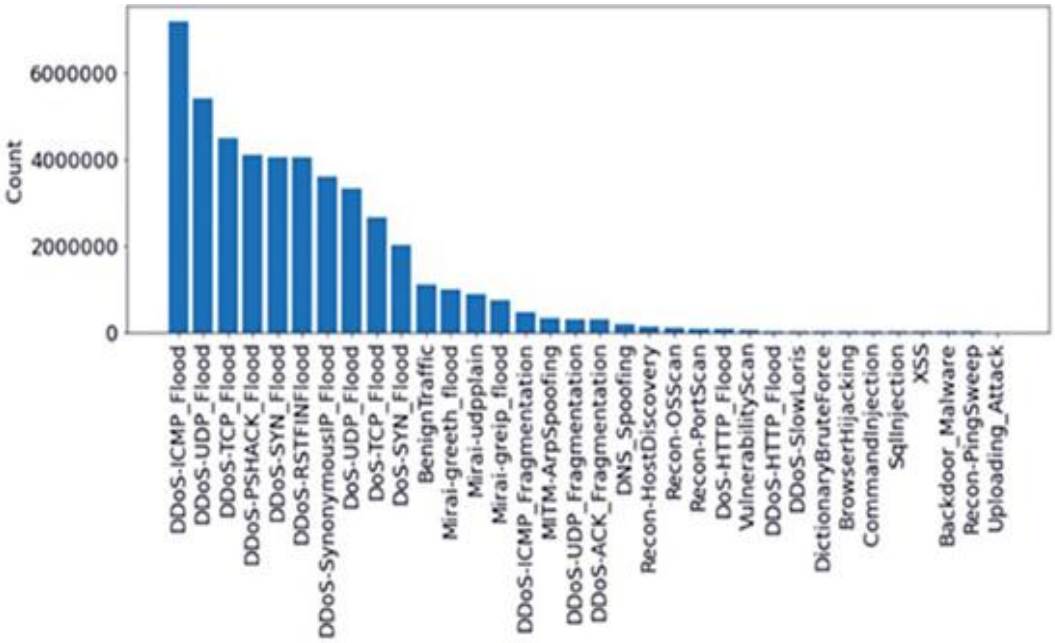

Figure 3: The instances count for each attack (Neto et al. (2023))

Table 2: Description of CICIoT2023 dataset (Neto et al. (2023)

Type	Target	Total Number of Records	Class Distribution
Benign	Benign	1,098,195	2.35%
DDoS	Attack	33,984,560	72.79%
DoS	Attack	8,090,738	17.33%
Mirai	Attack	2,634,124	5.64%
Recon	Attack	354,565	0.76%
Spoofing	Attack	486,504	1.04%
WebBased	Attack	24,829	0.05%
Bruteforce	Attack	13,064	0.03%
Total		**46,686,579**	**100%**

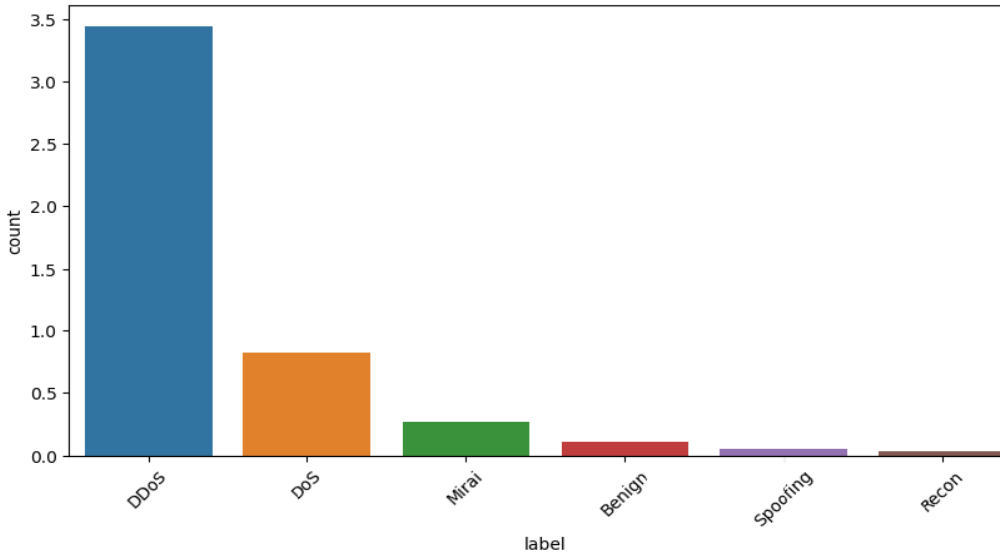

Figure 4: Data distribution of CICIoT2023 before balancing

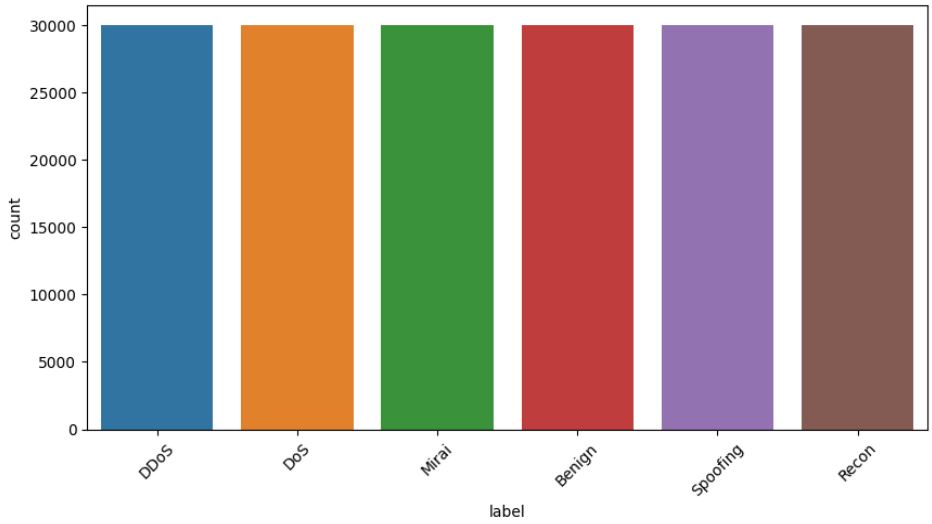

Figure 5: Data distribution after balancing

In Figure 6, feature analysis that displays how each dataset feature relates to other characteristics. The degree of relationships between features is provided by the correlation matrix. The highest relationship in these relationships is denoted by 1 and the lowest by -1.

3.3 Model building

In this study, we have chosen various learning models (Machine Learning models and Deep Learning models) including Support Vector Machine (SVM), Random Forest (RF), Deep Neural Network (DNN) and VGGNet. These learning models have been selected according to their high accuracy rate in classification in various research domains.

The optimization of those models is based on choosing the best hyperparameters. There are various techniques such as: Grid Search, RandomizedSearch and Bayesian Optimization.

In this study we have applied randoizedSearch provided by scikit-learn to get the best hyperparameters for SVM and Random Forest models. This technique is the most adequate and efficient when we have a high dimensional data.

- **Support Vector Machine:** Is a supervised classifier that can be used for classification or regression problems, it's main goal is to discover a hyperplane that maximizes the margin between classes while simultaneously minimizing classification errors (Hiri, M et al. 2023) [30]. This hyperplane is defined by the equation (2):

$$\boldsymbol{w} * \boldsymbol{x} + \boldsymbol{b} = \mathbf{0} \qquad \textbf{(2)}$$

Where: w is the weight vector (perpendicular to the plane), x is the input vector (training data) b is the bias (constant term)

In this simulation, The SVM model was built using the SVC class from the **sklearn.svm** module and the parameters were optimized for the best performance are (Table 3):

Regularization: C=2.0, Kernel=" rbf", and Gamma="auto"

- **Random Forest:** Is the famous supervised learning model, used for both: classification and regression. It can be defined as an ensemble method uses multiple decision trees. The principle of this model revolves around the idea that: by combining the predictions of each tree in the forest, we can avoid overfitting and we can improve the accuracy rate of classification.

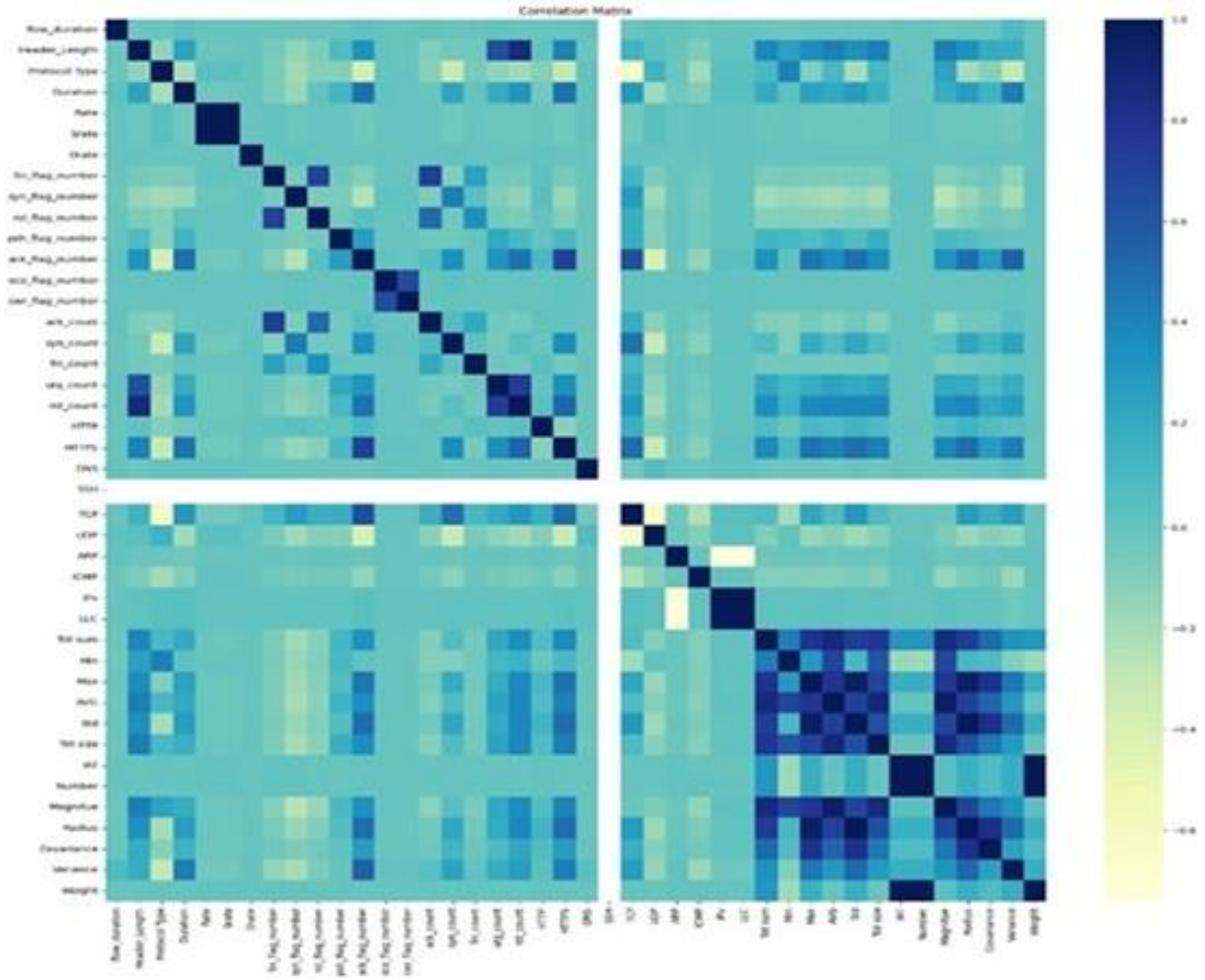

Figure 6: Correlation matrix results for the CICIoT2023 dataset

According to many research, random forest is a powerful machine learning model, because it can handle a large and huge amount of data. The Random Forest model was built using the RandomForestClassifier class from the **s**klearn.ensemble module and the parameters were optimized for the best performance (as illustrated in Table3):

Number of trees in the forest: n_estimators=200,

Maximum depth of the tree: max_depth=10

Deep Neural Network: Deep learning is a branch of ML that utilizes artificial neural networks for classification. It involves representing data in several layers, from a lower to a higher layer, extracting important features and more complex patterns in data to get the best possible outcome (Abbas, A et al 2022) [31]. Deep Neural Networks works as follows:

Input layer: is the first layer, where, each node represents a feature in the data.

Hidden layer: These layers connect the input and output layers to allow the network to learn intricate features and representations of the data. The more layers and neurons we implement, the deeper and more complex the network becomes.

Activation function: it's a function applied by each neuron, to affect it a corresponding weight

Output layer: which represents the final prediction of the model

Flatten layer: it's used to convert multidimensional input into a 1D vector before feeding it to dense layers

Dropout Layer: it's used to reduce overfitting by randomly deactivating a portion of neurons during training.
In this study, Figure 8 illustrate the architecture used for DNN.

Table 3: Summary of hyperparameters used

Model	Range of Hyperparameters	Best Hyperparameters used	Module
SVM	* Kernel Types: Linear, Polynomial Radial Basis Function (RBF): Sigmoid * Gamma: *C(Regularization Parameter) range is 10^{-3} to 10^{3}	Regularization: C=2.0 *Kernel=" rbf" *Gamma="auto	**SVC** class from the **sklearn.svm**
Random Forest	* **n_estimators**: Number of trees in the forest. Range: Usually between **10 to 1000** *** max_depth**: Maximum depth of each tree. Range: Typically between **1 and 50**	*Number of trees in the forest: n_estimators=200 *Maximum depth of the tree: max_depth=10	**RandomForestClassifier** class from the **sklearn.ensemble**
VGGNet	* range of Batch Size is 16 to 256 * range of Epoch is 10 to 100 *Dropout Rate range is 0.3 to 0.5 *Pooling Layers: Type: Max pooling Kernel size: 2x2	*three blocks of convolutional layers *2 fully connected layers with dropout layers in between. *The VGGNet model was trained for 10 epochs with a batch size of 64	**tensorflow.keras.models**
DNN	***** Batch Size Range**: 32, 64, 128, 256, 512** ***** Number of Layers Range**: 2 to 50+** ***** Activation Function**: ReLU, Leaky ReLU, Sigmoid, Tanh, ELU** ***** Optimizer**: SGD, Adam, RMSprop,**	*The DNN model was trained for 10 epochs ***Optimizer**: Adam * **Batch Size**: 64 * **Number of Layers**: 3 ***activation functions**=ReLu * **Dropout regularization**	**tensorflow.keras.models**

```
Epoch 1/10
2250/2250 ━━━━━━━━━━━━━━━━━━━━ 121s 53ms/step - accuracy: 0.7350 - loss: 0.6194 -
Epoch 2/10
2250/2250 ━━━━━━━━━━━━━━━━━━━━ 116s 51ms/step - accuracy: 0.7698 - loss: 0.5044 -
Epoch 3/10
2250/2250 ━━━━━━━━━━━━━━━━━━━━ 117s 52ms/step - accuracy: 0.7715 - loss: 0.4963 -
Epoch 4/10
2250/2250 ━━━━━━━━━━━━━━━━━━━━ 116s 52ms/step - accuracy: 0.7725 - loss: 0.4911 -
Epoch 5/10
2250/2250 ━━━━━━━━━━━━━━━━━━━━ 114s 51ms/step - accuracy: 0.7762 - loss: 0.4837 -
Epoch 6/10
2250/2250 ━━━━━━━━━━━━━━━━━━━━ 115s 51ms/step - accuracy: 0.7824 - loss: 0.4762 -
Epoch 7/10
2250/2250 ━━━━━━━━━━━━━━━━━━━━ 114s 51ms/step - accuracy: 0.7948 - loss: 0.4594 -
Epoch 8/10
2250/2250 ━━━━━━━━━━━━━━━━━━━━ 115s 51ms/step - accuracy: 0.8279 - loss: 0.4093 -
Epoch 9/10
2250/2250 ━━━━━━━━━━━━━━━━━━━━ 115s 51ms/step - accuracy: 0.8483 - loss: 0.3660 -
Epoch 10/10
2250/2250 ━━━━━━━━━━━━━━━━━━━━ 113s 50ms/step - accuracy: 0.8513 - loss: 0.3536 -
```

Figure 7: Architecture used for DNN model

```
dnn_model.summary()
```

Model: "sequential_3"

Layer (type)	Output Shape	Param #
dense_9 (Dense)	(None, 46, 256)	512
dropout_6 (Dropout)	(None, 46, 256)	0
dense_10 (Dense)	(None, 46, 256)	65,792
dropout_7 (Dropout)	(None, 46, 256)	0
flatten_1 (Flatten)	(None, 11776)	0
dense_11 (Dense)	(None, 6)	70,662

Total params: 136,966 (535.02 KB)
Trainable params: 136,966 (535.02 KB)
Non-trainable params: 0 (0.00 B)

Figure 8: Epoch used for DNN model

```
vgg_model.summary()
```

Model: "sequential"

Layer (type)	Output Shape	Param #
conv1d (Conv1D)	(None, 46, 64)	256
conv1d_1 (Conv1D)	(None, 46, 64)	12,352
max_pooling1d (MaxPooling1D)	(None, 23, 64)	0
conv1d_2 (Conv1D)	(None, 23, 128)	24,704
conv1d_3 (Conv1D)	(None, 23, 128)	49,280
max_pooling1d_1 (MaxPooling1D)	(None, 11, 128)	0
conv1d_4 (Conv1D)	(None, 11, 256)	98,560
conv1d_5 (Conv1D)	(None, 11, 256)	196,864
max_pooling1d_2 (MaxPooling1D)	(None, 5, 256)	0
flatten (Flatten)	(None, 1280)	0
dense (Dense)	(None, 256)	327,936
dropout (Dropout)	(None, 256)	0

Figure 9: VGGNet Architecture

DNN is considered as sequential modeles of layers and Dropout as illustrated in Figure 7.

Dropout regularization is a method for decreasing overfitting and enhancing deep neural network generalization. Fixed dropout for each unique model is taken into consideration for increased performance, which is assessed based on accuracy and assessment time (as shown in Table 3).

The number of epochs is a crucial algorithmic hyperparameters. It indicates how many iterations, or full runs of the training dataset, the algorithm will go through during training or learning. Figure 8 shown the number of epoch used in our simulation.

VGGNet: is a deep learning architecture, developed by the Visual Graphics Group (VGG) at Oxford University. Originally designed for image classification, VGGNet can be adapted for other complex tasks, including intrusion detection by transforming data into a format 2D that the model can process. This approach has been inspired by recent study introduced by authors (Bouijij, H et al (2025)) [32] that have successfully proposed an approach-based CNN architectures for non-image domains (phishing attack detection).

The architecture explores the relationship between network depth and performance, utilizing small convolution kernels and maximum pooling layers to build deep networks comprising 11 to 19 layers (Zhou, S et al. 2018) [33] as illustrated in Figure 9:

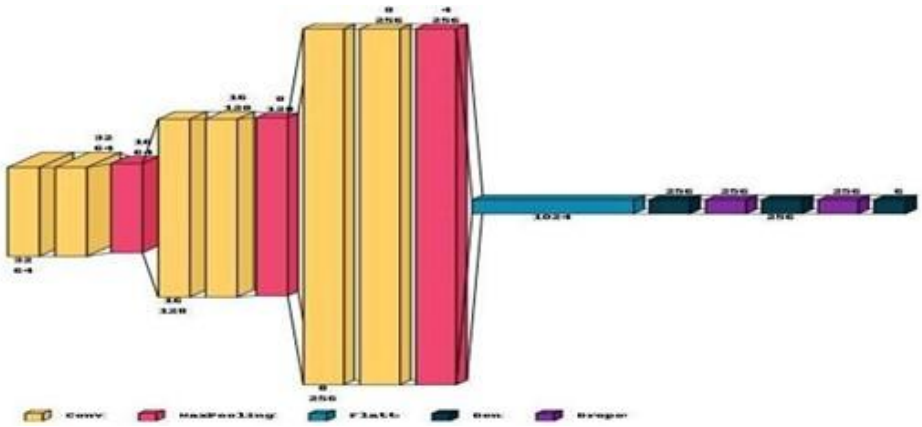

Figure 10: Architecture used for VGGNet model

Various studies have introduised VGGNet model for intrusion detection (Manjula, P et al, 2022) [34]. In this study, Figure 10 shown the architecture used for VGGNet, with a number of the epoch used to train it. It was a lightweight adaptation inspired by VGGNet**,** designed to reduce training time and computational cost for the CICIoT2023 dataset. The model was built using the Sequential class from the keras. model's module. It generally consists of three blocks of convolutional layers followed by max-pooling layers, and then 2 fully connected layers with dropout layers in between (as illustrated in Table 3). The VGGNet model was trained for 10 epochs with a batch size of 64.

3.4 Evaluation metrics

To assess the efficiency of our proposed approach, we have used a wide range of metrics as shown in Table 4. These metrics are based on the values: False Positive (FP): indicates that a normal class is wrongly predicted as an attack False Negative (FN): is false alarm, generated when the actual class is an attack, but predicted as normal. (TP): is a successful identification of a class as attack. (TN): indicates no attack occurred and no alarm generated.

The evaluation also includes the execution time as a key metric to analyze the practicality of each model in real-time IoT environment.

Table 4: Summary of the standard evaluation metrics used

Metric	Formula
Accuracy	$\frac{TP + TN}{TP + TN + FP + FN}$
Precision	$\frac{TP}{TP + FP}$
Recall	$\frac{TP}{TP + FN}$
F1-score	$\frac{2TP}{2TP + FN + FP}$

4 Results and discussion

In this section, we present the experiment result of our chosen learning modeles using a real-time and benchmark dataset CICIoT2023, as well as a summary of comparison between their results. The assessment of the research findings was carried out meticulously, employing a variety of measures to measure the effectiveness of the three modeles in detecting various types of attacks, such as Denial of Service (DoS) and DDoS attacks, particularly in an IoT environment.

SVM result: Table 5 and Figure 11 illustrate summary of the experiment result of the SVM model

Table 5: Accuracy report of SVM model

SVM Classification Report	precision	recall	f1-score	support
Benign	0.69	0.85	0.76	5933
DDoS	0.84	0.63	0.72	5976
DoS	0.70	0.88	0.78	5922
Mirai	1.00	0.99	1.00	6086
Recon	0.79	0.75	0.77	6119
Spoofing	0.82	0.69	0.75	5964
accuracy	0.80022222			
macro avg	0.81	0.80	0.80	36000
weighted avg	0.81	0.80	0.80	36000

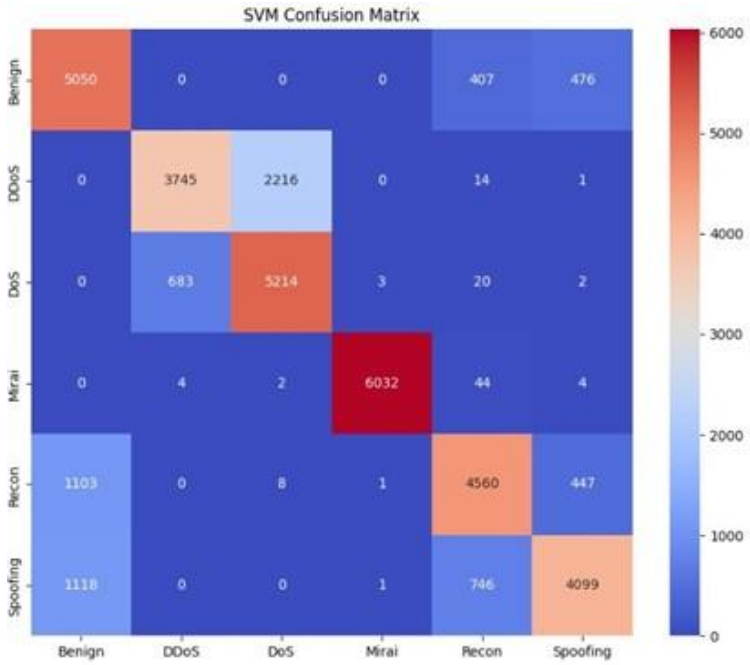

Figure 11: Confusion matrix of SVM model

According to the report accuracy (Table 5), the SVM model does not achieve a good accuracy to classify Begnin, DoS, DDoS and Recon.

SVM in this case is not a good classifier, because SVMs are inherently designed for binary classification and also SVMs are not ideal for datasets with a very large number of trainings. The use of SVM in this study has as objective demonstrating the limitation of traditional ML models in comparison with DL approaches.

- **Random Forest result**

Table 6 and Figure 12 illustrate a detailed summary of the experiment result of the Random Forest model.

According to the accuracy report (Table 6) and the confusion matrix (Figure 12), Random Forest model achieved strong overall performance but still exhibited some misclassifications. While Random Forest performed well, it did not achieve perfect accuracy, especially traffic classes. Generally, Random Forest is a good classifier for several reasons:

- The combination of predictions of multiple Decision Trees in the forest improve the accuracy rate
- The RF model is suitable when the datasets contains huge amount of data and many features, because the model is able to select the best features for each tree.

Table 6: Accuracy report of random forest model

Random Forest Classification Accuracy: 0.918138888888				
	precision	**recall**	**f1-score**	**support**
Benign	**0.81**	**0.88**	**0.84**	**5933**
DDoS	**1.00**	**0.99**	**1.00**	**5976**
DoS	**1.00**	**0.99**	**1.00**	**5922**
Mirai	**1.00**	**0.99**	**1.00**	**6086**
Recon	**0.81**	**0.84**	**0.82**	**6119**
Spoofing	**0.90**	**0.81**	**0.86**	**5964**
accuracy	**0,92**			
macro avg	**0.92**	**0.92**	**0.92**	**36000**
weighted avg	**0.92**	**0.92**	**0.92**	**36000**

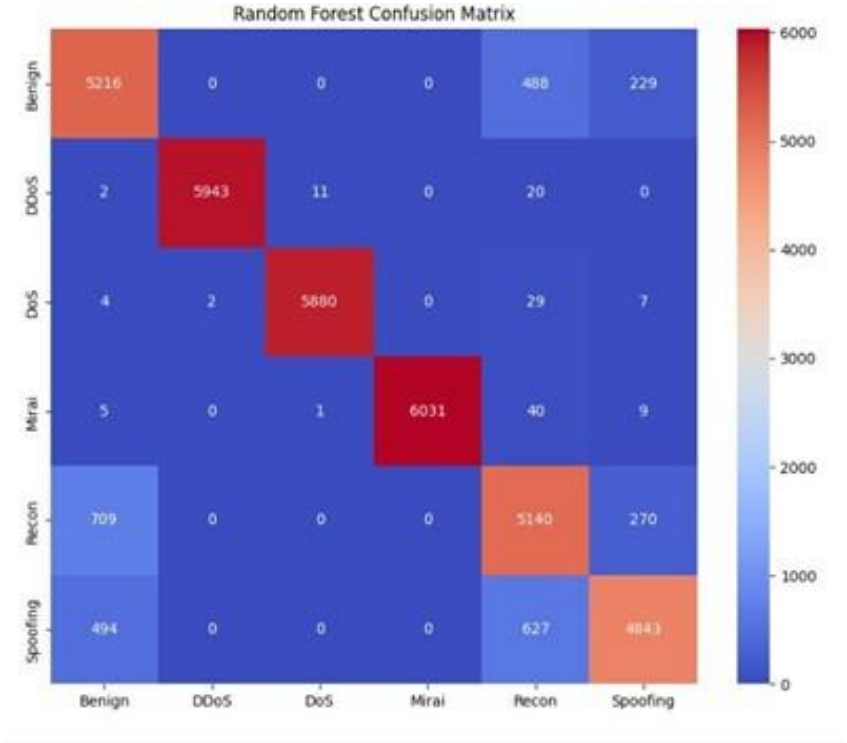

Figure 12: Confusion matrix of Random Forest model

Table 7: Accuracy report of VGGNet model

VGGNet Classification Accuracy: 0.89275				
	Precision	**recall**	**f1-score**	**support**
Benign	**0.75**	**0.87**	**0.81**	**5933**
DDoS	**1.00**	**0.99**	**1.00**	**5976**
DoS	**1.00**	**1.00**	**1.00**	**5922**
Mirai	**1.00**	**1.00**	**1.00**	**6086**
Recon	**0.78**	**0.80**	**0.79**	**6119**
Spoofing	**0.85**	**0.70**	**0.77**	**5964**
accuracy	**0.89**			
macro avg	**0.90**	**0.98**	**0.89**	**36000**
weighted avg	**0.90**	**0.98**	**0.89**	**36000**

- **VGGNet result**

Table 7 and Figure 13 below, provide a detailed summary of the experiment result of the VGGNet model

According to the report accuracy (Table 7), the VGGNet model follow the random forest model as a second-best classifier, with an accuracy rate of 89,27%. VGGNet have successfully classified almost all the traffic, especially DoS, DDoS and Mirai attacks.

VGGNet is a Deep Learning models excels in feature extraction from data, but reliance on convolutional layers may limit its capacity to generalize successfully. VGGNet's somewhat lower overall accuracy indicates that it might have trouble differentiating innocuous traffic from more nuanced or changing attack patterns, even while it is highly effective at classifying DoS, DDoS, and Mirai attacks. However, Random Forest's ability to alter its decision boundaries enables it to function reliably against different kinds of attacks.

Furthermore, since decision tree-based models are better at capturing hierarchical relationships in tabular datasets, they may be more appropriate for the structured nature of network traffic data. Convolutional neural networks (CNNs), such as VGGNet, on the other hand, are better suited for the extraction of spatial features, which makes them very useful for tasks involving images but possibly less effective for categorical and sequential data.

Table 8 and Figure 14 provide a detailed summary of the experiment result of the DNN model. DoS, DDoS and Mirai have been Well classified by DNN, according to all the measurement metrics.

- **DNN result**

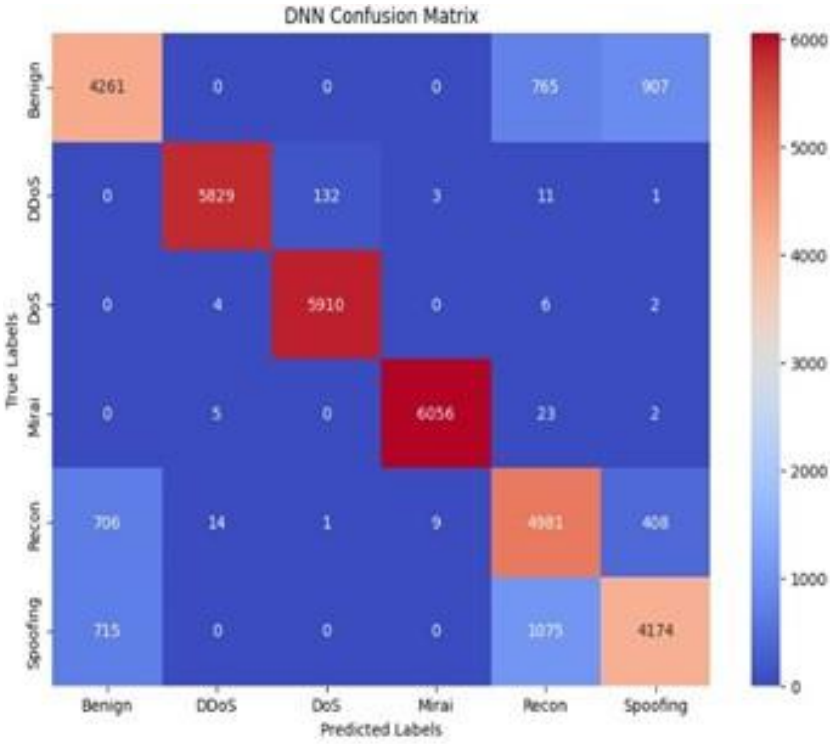

Figure 13: Confusion matrix of VGGNet model

Table 8: Accuracy report of DNN model

DNN Classification Report	precision	recall	f1-score	support
Benign	0.75	0.72	0.73	5933
DDoS	1.00	0.98	0.99	5976
DoS	0.98	1.00	0.99	5922
Mirai	1.00	1.00	1.00	6086
Recon	0.73	0.81	0.77	6119
Spoofing	0.76	0.70	0.73	5964
accuracy	0.86697222222222			
macro avg	0.87	0.87	0.87	36000
weighted avg	0.87	0.87	0.87	36000

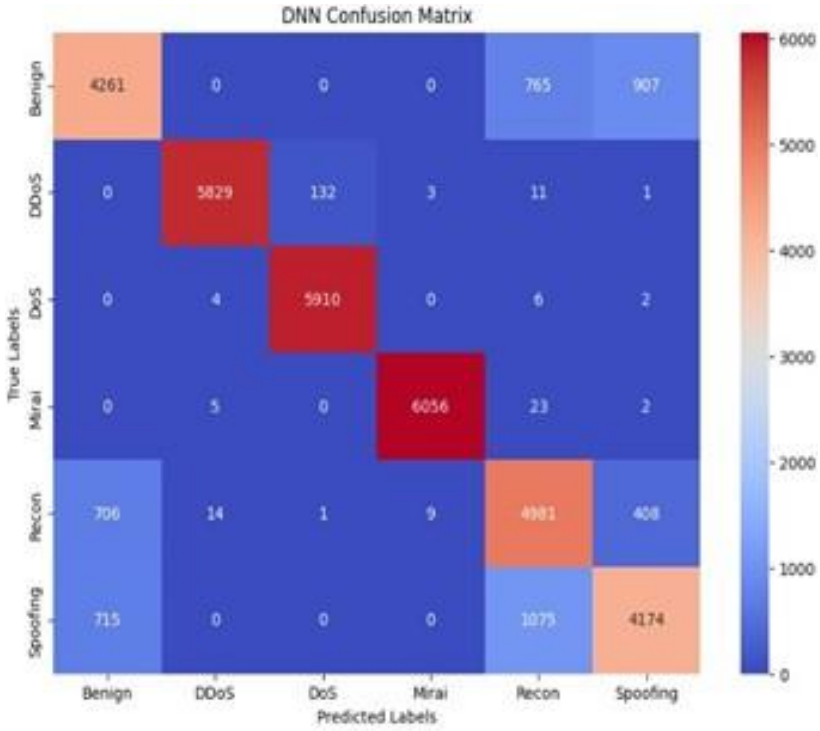

Figure 14: Confusion matrix of DNN model

According to the accuracy report (Table 8), DNN model have proven to be a good classifier, achieving a high accuracy in classifying DoS, DDoS and Mirai attacks.

Unlike traditional machine learning modeles, DNN and thanks to its multi-layers, it can learn from complex data and extract the most significant features even with big datasets with a large amount of data, this ability make DNN as a good classifier

As summary

Our study, underscore several key advancements over prior works in both methodological rigor and empirical performance. In contrast to earlier studies that often relied on outdated datasets or limited benchmarking. Our approach integrates a broader and more contemporary range of test scenarios. Notably, prior models frequently underperformed due to insufficient hyperparameters

We can conclude that this study, provide an effecient outfinding in comparison with the study provided by (Dushimimana, A. et al. (2020)), who has used a small portion of the NSL-KDD, consequentely this work does not provide a general result

(Ayyaz-ul-Haq Qureshi, B. et al (2019)) and (Jiang, K. et al. (2020)), have evaluated various algorithms of classification, however, attackers enhance their behavioral continuously, consequently, not using a recent and real-time dataset doesn't provide an efficient result.

The work by Berqia et al. (2024) and this study are comparable in a number of ways, especially in their emphasis on applying machine learning methods to identify DDoS attacks in Internet of Things settings using tuning—a limitation we explicitly address through a systematic and automated optimization process across all major model parameters.

According to the experiment result, all the model building demonstrate significant performance in classifying normal and abnormal activities within IoT environment (Table 9). Our work builds on and extends the capabilities of previous studies.

The study presented by (Chirra, D. R. (2023)), explored deep learning models for enhancing IoT network intrusion detection. While deep learning models can offer performance gains, they also introduce complexity and require more computational resources.

While, the result obtained by (El Yamani et al. (2024)) and (Pokhrel, S et al. (2021)) achieve a high accuracy rate of attacks classification, however the complexity in term of execution time and computational the same dataset CICIoT2023. To improve model performance and generalizability, we use sophisticated hyperparameter tweaking strategies in our work, which sets it apart.

Our study provides several advantages:

- High Accuracy in detecting a wide range of network intrusions.
- Scalability: That making it suitable for real-time intrusion detection.
- Resource Efficiency: Optimized for deployment in resource-constrained environments such as IoTs, where computational and energy resources are limited.

These advancements highlight the potential of machine and deep learning models to significantly, enhance the robustness and reliability of IDS in protecting modern network infrastructures against evolving cyber threats.

Despite the promising results, our study has certain limitations. One of the primary challenges is the computational complexity associated with deep learning models, which requires significant processing power and memory. VGGNet and DNN models involve many parameters during training, as well as, while hyperparameters tuning significantly enhance the accuracy rate, but often require a lot of processing power (CPU and storage). This can be a constraint in the context of IoT, where resources devices are typically limited.

Table 9: Summary of evaluation results

Model	Approach	Accuracy	Precision	Recall	Training time(s)
SVM	Machine learning	80,02%	81%	80%	2400
Random Forest	Machine learning	91,81%	92%	92%	1800
VGGNet	Deep learning	89,27%	90%	89%	3400
DNN	Deep learning	86,69%	87%	87%	2400

5 Conclusions

IoT technology still attracked researchers thanks to its proficiency of changing physical items of diverse application domains into Internet hosts. There are now more security and privacy concerns as a result of the Internet of Things' exponential growth. Device vulnerabilities brought on by hacker cybercrime and inappropriate system resource usage are the main cause for a large number of these threats. However, attackers may also take advantage of the IoT tremendous potential as a new means to harm users' privacy and security. In this paper, we have evaluated various machine and deep learning models including Support Vector Machine, Random Forest, VGGNet and Deep Neural Network, using a well-known, benchmark and real time dataset CICIoT2023. Various evaluation metrics have been used to analyze in depth our approach. The experiment results demonstrate the robustness of Random Forest model among the other entire model in classifying normal and abnormal traffic. Finally, this research paper highlights the robustness of machine and deep learning models to transform the landscape of intrusion detection, as well as finding the best hyperparameters to fine-tuning the chosen models. The advancements presented in this research facilitate and smooth the way for more safe and robust network systems, capable of resistant the challenges posed by the ever-evolving cyber threat land- scape. Future research should focus on addressing the limitations identified in this study, by exploring techniques such as generative adversarial networks and federated Learning approach for optimizing the Deep Learning architecture and deployment in resource-constrained environments.

To facilitate the reproducibility and further research, the implementation details of the proposed framework is publically available on the following GitHub Repository: https://github.com/users/FAREShajar-doc/Ml-and-Dl-for-Intrusion-Detection

References

[1] Madakam, S., Ramaswamy, R., & Tripathi, S. (2015). Internet of Things (IoT): A literature review. Journal of Computer and Communications, 3(5), 164-173.

[2] Zhao, J. (2025). Intelligent Logistics Path Optimization Algorithm Based on Internet of Things Sensing Technology. Informatica, 49(19).

[3] De Michele, R., & Furini, M. (2019, September). Iot healthcare: Benefits, issues and challenges. In Proceedings of the 5th EAI international conference on smart objects and technologies for social good (pp. 160-164).

[4] Zantalis, F., Koulouras, G., Karabetsos, S., & Kandris, D. (2019). A review of machine learning and IoT in smart transportation. Future Internet, 11(4), 94

[5] Jeong, Y. S., & Park, J. H. (2019). IoT and smart city technology: challenges, opportunities, and solutions. Journal of Information Processing Systems, 15(2), 233- 238

[6] Farooq, M. S., Riaz, S., Abid, A., Umer, T., & Zikria, Y. B. (2020). Role of IoT technology in agriculture: A systematic literature review. Electronics, 9(2), 319.

[7] Moslehi, M. M. (2025). Exploring coverage and security challenges in wireless sensor networks: A survey. Computer Networks, 111096.

[8] Abomhara, M., & Køien, G. M. (2015). Cyber security and the internet of things: vulnerabilities, threats, intruders and attacks. Journal of Cyber Security and Mobility, 65-88

[9] Zarpelˆao, B. B., Miani, R. S., Kawakani, C. T., & De Alvarenga, S. C. (2017). A survey of intrusion detection in Internet of Things. Journal of Network and Computer Applications, 84, 25-37.

[10] Huang, H., Xiao, D., Wang, M., Liang, J., Li, M., Chen, L., & Liu, Y. (2025). Fog-driven communication-efficient and privacy-preserving federated learning based on compressed sensing. Computer Networks, 259, 111043.

[11] Hussain, F., Hussain, R., Hassan, S. A., & Hossain, E. (2020). Machine learn- ing in IoT security: Current solutions and future challenges. IEEE Communications Surveys & Tutorials, 22(3), 1686-1721.

[12] Du, J., Wang, X., & Zhang, H. (2025). Secure Power Management in Wireless Sensor Networks for Power Monitoring Using Deep Reinforcement Learning. Informatica, 49(19).

[13] Amanullah, M. A., Habeeb, R. A. A., Nasaruddin, F. H., Gani, A., Ahmed, E., Nainar, A. S. M., ... & Imran, M. (2020). Deep learning and big data technologies for IoT security. Computer Communications, 151, 495-517

[14] Nahida Islam1, Fahiba Farhin1 , Ishrat Sultana1 , M. Shamim Kaiser1 , Md. Sazzadur Rahman1 , Mufti Mahmud2 , A. S. M. Sanwar Hosen3 and Gi Hwan Cho. Towards Machine Learning Based Intrusion Detection in IoT Networks. Computers, Materials & Continua. DOI:10.32604/cmc.2021.018466

[15] Lin, T. (2020, November). Deep learning for IoT. In 2020 IEEE 39th International Performance Computing and Communications Conference (IPCCC) (pp. 1-4).

[16] Li, Y., Zuo, Y., Song, H., & Lv, Z. (2021). Deep learning in security of internet of things. IEEE Internet of Things Journal, 9(22), 22133-22146.

[17] Abbass, W., Bakraouy, Z., Ba¨ına, A., & Bellafkih, M. (2018, October). Classifying IoT security risks using deep learning algorithms. In 2018 6th International Conference on Wireless Networks and Mobile Communications (WINCOM) (pp. 1-6). IEEE

[18] Pokhrel, S., Abbas, R., & Aryal, B. (2021). IoT security: botnet detection in IoT using machine learning. *arXiv preprint arXiv:2104.02231*.

[19] Ioannou, C., & Vassiliou, V. (2019, May). Classifying security attacks in IoT networks using supervised learning. In 2019 15th International conference on distributed computing in sensor systems (DCOSS) (pp. 652-658). IEEE.

[20] Chirra, D. R. (2023). Deep Learning Techniques for Anomaly Detection in IoT Devices: Enhancing Security and Privacy. Revista de Inteligencia Artificial en Medicina, 14(1), 529-552

[21] Berqia, A., Bouijij, H., Merimi, A., & Ouaggane, A. (2024, May) Detecting DDoS attacks using machine learning in IoT environment. In 2024 International Conference on Intelligent Systems and Computer Vision (ISCV) (pp. 1-8). doi: 10.1109/ISCV60512.2024.10620122.

[22] El Yamani, Y., Fadili, Y., Kilani, J., El Kamoun, N., Baddi, Y., & Bensalah, F. (2024, July). Hybrid Models for IoT Security: Tackling Class Imbalance. In 2024 11th International Conference on Wireless Networks and Mobile Communications (WINCOM) (pp. 1-6). doi: 10.1109/WINCOM62286.2024.10656654.

[23] Sarhan, M. Layeghy, S. Moustafa, N. Gallagher, M, Portmann, M. (2024). Feature extraction for ML based Intrusion Detection in IoT networks. Digital Communications and Networks. No 10 (2024): 205-216

[24] Elouardi, S., Motii, A., Jouhari, M., Amadou, A. N. H., & Hedabou, M. (2024). A survey on Hybrid-CNN and LLMs for intrusion detection systems: Recent IoT datasets. *IEEE Access*

[25] B. Ayyaz-ul-Haq Qureshi, H. Larijani, J. Ahmad and N. Mtetwa, "A heuristic intrusion detection system for internet-of-things (IoT)," in Intelligent Computing: Proc. of the 2019 Computing Conf., London, United Kingdom, pp. 86–98, 2019

[26] K. Jiang, W. Wang, A. Wang and H. Wu, "Network intrusion detection combined hybrid sampling with deep hierarchical network," IEEE Access, vol. 8, pp. 32464–32476, 2020

[27] A. Dushimimana, T. Tao, R. Kindong and A. Nishyirimbere, "Bi-directional recurrent neural network for intrusion detection system (IDS) in the internet of things (IoT)," International Journal of Advanced Engineering Research and Science, vol. 7, no. 3, pp. 524–539, 2020.

[28] Z. Li, P. Batta and L. Trajkovic, "Comparison of machine learning algorithms for detection of network intrusions," in 2018 IEEE Int. Conf. on Systems, Man, and Cybernetics, Miyazaki, Japan, pp. 4248–4253, 2018

[29] Neto, E. C. P., Dadkhah, S., Ferreira, R., Zohourian, A., Lu, R., & Ghorbani, (2023). CICIoT2023: A real-time dataset and benchmark for large-scale attacks in IoT environment. Sensors, 23(13), 5941.

[30] Hiri, Mustafa; Chrayah, Mohamed; Ourdani, Nabil; Aknin, Noura. (2023). Machine Learning Techniques for Diabetes Classification: A Comparative Study. International Journal of Advanced Computer Science and Applications. 14. 10.14569/IJACSA.2023.0140982.

[31] Abbas, A. M. A. Khan, S. Latif, M. Ajaz, A. A. Shah, (2022)"A new ensemble- based intrusion detection system for internet of things," Arabian Journal for Science and Engineering, Springer

[32] Bouijij, H., & Berqia, A. (2025). Intelligent phishing URL classification using CNN. In Recent Advances in Internet of Things Security (pp. 122-130). CRC Press.

[33] Zhou, S., Liang, W., Li, J., & Kim, J. U. (2018). Improved VGG model for road traffic sign recognition. Computers, Materials & Continua, 57(1), 11-24

[34] Manjula, P., & Priya, S. B. (2022). An effective network intrusion detection and classification system for securing WSN using VGG-19 and hybrid deep neural network techniques. Journal of Intelligent & Fuzzy Systems, 43(5), 6419-6432

[35] Sholeh, M., & Nurnawati, E. K. (2024, July). Comparison of Z-score, min-max, and no normalization methods using support vector

machine algorithm to predict student's timely graduation. In AIP Conference Proceedings (Vol. 3077, No. 1). AIP Publishing.

[36] Qi, D., Junaidi, A., Howe, C. W., & Zain, A. M. (2023, October). Improving Unbalanced Security X-Ray Image Classification Using VGG16 and AlexNet with Z-Score Normalization and Augmentation. In International Conference on Electronics, Biomedical Engineering, and Health Informatics (pp. 205-217). Singapore: Springer Nature Singapore.