Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2024.053542

ARTICLE

Detection of Real-Time Distributed Denial-of-Service (DDoS) Attacks on Internet of Things (IoT) Networks Using Machine Learning Algorithms

Zaed Mahdi[1,*], Nada Abdalhussien[2], Naba Mahmood[1] and Rana Zaki[3,*]

[1]Information Technology Center, University of Technology, Baghdad, 00964, Iraq

[2]Continuous Education Center, University of Technology, Baghdad, 00964, Iraq

[3]Computer Science Department, University of Technology, Baghdad, 00964, Iraq

*Corresponding Authors: Zaed Mahdi. Email: zaed.s.mahdi@uotechnology.edu.iq; Rana Zaki. Email: rana.m.zaki@uotechnology.edu.iq

Received: 03 May 2024 Accepted: 20 June 2024 Published: 15 August 2024

ABSTRACT

The primary concern of modern technology is cyber attacks targeting the Internet of Things. As it is one of the most widely used networks today and vulnerable to attacks. Real-time threats pose with modern cyber attacks that pose a great danger to the Internet of Things (IoT) networks, as devices can be monitored or service isolated from them and affect users in one way or another. Securing Internet of Things networks is an important matter, as it requires the use of modern technologies and methods, and real and up-to-date data to design and train systems to keep pace with the modernity that attackers use to confront these attacks. One of the most common types of attacks against IoT devices is Distributed Denial-of-Service (DDoS) attacks. Our paper makes a unique contribution that differs from existing studies, in that we use recent data that contains real traffic and real attacks on IoT networks. And a hybrid method for selecting relevant features, And also how to choose highly efficient algorithms. What gives the model a high ability to detect distributed denial-of-service attacks. the model proposed is based on a two-stage process: selecting essential features and constructing a detection model using the K-neighbors algorithm with two classifier algorithms (logistic regression and Stochastic Gradient Descent classifier (SGD), combining these classifiers through ensemble machine learning (stacking), and optimizing parameters through Grid Search-CV to enhance system accuracy. Experiments were conducted to evaluate the effectiveness of the proposed model using the CIC-IoT2023 and CIC-DDoS2019 datasets. Performance evaluation demonstrated the potential of our model in robust intrusion detection in IoT networks, achieving an accuracy of 99.965% and a detection time of 0.20 s for the CIC-IoT2023 dataset, and 99.968% accuracy with a detection time of 0.23 s for the CIC-DDoS 2019 dataset. Furthermore, a comparative analysis with recent related works highlighted the superiority of our methodology in intrusion detection, showing improvements in accuracy, recall, and detection time.

KEYWORDS

Cyber-attacks; distributed denial of service (DDoS); real-time; internet of things

1 Introduction

The IoT has revolutionized the world of the Internet, where hundreds of devices are connected in one way or another. Through the increasingly large growth in Internet of Things technologies, this exposure has led to many threats. This necessitated the need to secure networks and associated devices and protect them from hackers and exploiters. Cyber attacks are an attempt to infiltrate a network or associated devices to steal or manipulate data and thus harm the network. They are carried out by individuals or organizations. Therefore, all measures must be taken to reduce such risks and secure the network well. Cyber attack protection technologies work hand in hand with IoT technologies that monitor the network and detect threats [1,2]. One of these attacks is DDoS attacks that create big traffic on the network [3]. Attackers are using new techniques to develop their attacks, which are difficult to detect [4]. Current research has to focus on modern methods and technologies used in developing intrusion detection systems, accuracy in selecting recent data for training models, and data processing methods to obtain relevant data to be more efficient in training [5,6]. Detection systems rely heavily on machine learning algorithms, which improve the efficiency of systems and detect attacks [7,8]. Another technique that is used is data mining, which is used to prepare data that trains systems and increases the efficiency of discovery [9]. The proposed approach will focus on the weaknesses found in existing studies, which is the process of selecting recent data that contains real traffic on Internet of Things networks. Methods of processing data and extracting only necessary features. Which will contribute to training the model well, and also to obtaining a very short detection time. The feature selection process is conducted in two stages.

First, the contrast thresholding algorithm will be used to eliminate the least important features. Then, in the second stage, features with the highest scores will be selected. Following that, ensemble learning algorithms will be developed to achieve optimal accuracy for the proposed system. The effectiveness of the proposed model in reducing the false alarm rate will be demonstrated. We used an improved dataset consisting of CIC-IoT2023 and CIC-DDoS2019 data. The results of our proposed model demonstrate significantly higher accuracy compared to existing systems, as well as reduced training time.

The contributions of this research are:

1. Using feature selection based on the highest variance significantly reduces the training time of the proposed model.

2. Development of an improved K-neighbors algorithm (KNN) achieved by integrating it with various classifiers through stacking ensemble learning, incorporating logistic regression and SGD classifier.

3. Evaluation of the proposed model's performance on both the (CIC-IoT2023) and (CIC-DDoS2019) datasets, thereby demonstrating its effectiveness in intrusion detection.

The remainder of this essay is organized as follows. Section 2 explains the literature review of the modern models of intrusion detection. Materials, and methodology form Section 3, where a detailed explanation of the requirements and working method of the proposed methodology for selecting features and building the model for intrusion detection for the K-neighbors algorithm is listed. Section 4 presents experimental outcomes for two data sets (CIC-IoT2023 and CIC-DDoS2019). The performance evaluation results of the proposed methodology are presented in Section 5. In Section 6, the conclusion and proposed future ideas are presented.

2 literature Review

This section provides an overview of the literature concerning different models developed for detecting DDoS attacks and identifying their features and explaining the existing weaknesses. It also makes a comparison with the literature, based on the dataset used, for easy comparison between them, as explained in Table 1.

Table 1: Summary of the limitations in literature review on intrusion detection systems (IDS)

Ref.	Year	Algorithm	Dataset	Accuracy	Limitations
[10]	2024	CNN	CIC-DDoS2019	99.68%	The authors relied on only one dataset in training the model and it is considered somewhat old, which negatively affects the accuracy of the model in detecting modern attacks. The study also lacks feature selection methods.
[11]	2023	KNN, DT, SGB, NB, SVM, LR	CIC-DDoS2019	99.6%	The authors in this study did not overcome the challenges of selecting a recent dataset in training the model.
[12]	2023	DCNN	(InSDN, CIC-IDS2017, and CIC-DDoS2019)	99.99%	The datasets used are considered outdated.
[13]	2023	DNN, CNN and RNN	CIC-DDoS2019	99.99%	The datasets used are outdated. The authors use distributed edge computing, which reduces the load on the server and increases the load on the peripheral devices.

(Continued)

Table 1 (continued)

Ref.	Year	Algorithm	Dataset	Accuracy	Limitations
[14]	2024	LSTM	CIC-IoT2023	98.75%	Using only one data set, which affects the actual application of the model. The accuracy was 98.75%, and no feature selection methods were used.
[15]	2024	ARF-ADWIN, SRPs-DDM and ARF-DDM, KNN-ADWIN	CIC-IoT2023 and IoTID20	99.33% and 99.54%	The model's accuracy is 99.33%. Not using feature selection methods, lack of accuracy is believed for this reason.
[16]	2024	Multi-class classification	CIC-IoT2023	96.56%	Using the RNN algorithm, it is known to contain a number of problems in its application. Therefore, the model's accuracy was 99.56%.

The authors in [10] presented a model that uses convolutional neural networks (CNNs) to detect DDoS threats to Internet of Things networks. It analyzes traffic within the network. To evaluate the performance, they used the CIC-DDoS2019 dataset. A detection accuracy of 99.68% was obtained for the model. However, the data set used is old, and this leads to a problem in detecting attacks due to the development of attackers' methods. Another challenge is choosing the features that are very important in giving better training time. In [11], the authors proposed a model for detecting distributed denial of service attacks on the Internet of Things, based on multi-classification machine learning algorithms (KNN, decision tree (DT), SGB, Naive Bayes (NB), support vector machine (SVM), Linear regression (LR)). In the process of training the model and evaluating performance, the authors used a dataset (CIC-DDoS2019). The authors show the superiority of the SVM algorithm in results. The authors tried to improve the attack detection model, but the problem remained the same as the previous study, which is the old data used in the training and selection of relevant features. The authors in [12] proposed a modern attack detection approach based on the diffusion-convolutional neural networks algorithm (DCNN). They obtained a model accuracy of 99.99%, which is considered very well, but the same challenges that existed in previous studies remained, which is the use of old datasets In software-defined networking (SDN, CIC-IDS2017, and CIC-DDoS2019). The results may be good, but they will face difficulties in detecting intrusion when implementing it. The authors in [13] proposed a model for detecting DDoS attacks, which is considered one of the good and modern methodologies, and also in feature selection they used heterogeneously integrated feature selection and the random forest algorithm. However, the problem remains that the data set used is outdated. I was hoping

they would use a modern data set. They used the CIC-DDoS2019 dataset, and this may affect the accuracy of the actual data detection even though they obtained a very high model accuracy of 99.99%. The authors in [14] overcome the challenges in previous studies, using a very modern data set (CIC-IoT2023). Where they proposed a system for detecting Cyber Attacks on IoT networks based on the LSTM algorithm and used the CIC-IoT2023 dataset to evaluate the system performance. The system accuracy was 98.75%. The accuracy of the model was not very high, and it is believed that if one of the feature selection methods had been used, the accuracy results would have been higher. The authors in [15] proposed an adaptive framework for detecting DDoS traffic on Internet of Things networks. Based on ARF-ADWIN, SRPs-DDM and ARF-DDM, KNN-ADWIN algorithms. To evaluate the performance, two datasets were used CIC-IoT2023 and IoTID20, and the accuracy of the model was 99.33% and 99.54%. Here the problem exists, as in the previous study, which is that the accuracy of the model is not good. It is believed that the problem lies in the method of choosing appropriate algorithms or the large number of algorithms that may have led to overfitting. The authors in [16] proposed a model based on multi-class classification to detect Cyber Attacks on the IoT network. They used Deep Neural Networks (DNN), CNN and Recurrent Neural Network (RNN). To evaluate the performance, they used the CIC-IoT2023 dataset. The authors show that the highest accuracy of the model was with RNN, which is 96.56%. The accuracy of the model is considered average, but it is believed that if the authors had used the Long Short-Term Memory algorithm (LSTM), which is an improved version of RNN, the results would have been higher, especially if it had a good feature selection.

Through previous studies, we conclude that the challenges facing intrusion detection systems are choosing a modern data set for training and the method of selecting appropriate features, as these are the challenges that are responsible for training the model and giving us good results.

3 Materials and Methodology

This section describes the procedures for data processing, feature selection, and methodology used in designing the proposed model.

3.1 Intrusion Detection System (IDS)

IDS is one of the popular technologies used to detect malicious activities and threats to the network [17]. It classifies and monitors data traffic within the network, and determines whether it is normal or abnormal traffic [18]. It can be used within networks or within devices. It can be in the form of systems or devices [19,20]. It plays a crucial role in detecting external threats to the network that may result in intrusion or data breaches, as shown in Fig. 1, illustrating the workings of an intrusion detection model [21,22].

3.2 Data Mining (DM) Technology with (IDS)

Machine learning and data mining are highly esteemed in the field of cyberattack detection and prediction [23]. Often referred to as knowledge discovery, this approach involves extracting insights from data, identifying connections within the data, and recognizing patterns, thus improving decision-making accuracy [24]. This process enables the identification of significant, previously unknown data and the creation of models or patterns, such as instance-based instances, rules, and decision chains [25].

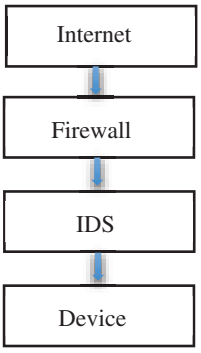

Figure 1: IDS model

3.2.1 Feature Select, Technique Variance Threshold and Select K Best (Chi2)

Feature selection plays a vital role in identifying relevant data for a given task, ultimately improving model accuracy and training speed. However, it is a challenging task that necessitates a profound understanding of the data [26,27]. Given the vastness of datasets, feature selection is considered one of the main challenges in machine learning and an essential part of the data processing and analysis pipeline [28,29].

1–There are various methods available, each tailored to the specific characteristics of the data. The variance threshold technique is useful for selecting features with significant variation while discarding those with minimal contrast, which are considered irrelevant to the task. High-variance features that are considered more important can be used, while low-variance features lead to poor model training [30].

$$\text{Variance Threshold } = p\,(1-p) \text{ [30]} \tag{1}$$

This equation is used to calculate the minimum variance threshold for a binary variable. Where if we have a binary variable, it will take the value of 1 or 0.

p = proportion of those who chose value feature 1.

$1-p$ = The probability that the variable takes the value 0.

2– Select K best is one of the most widely used techniques that select relevant features based on the K value specified by us. The select K best technique also includes calculating the Chi-squared between features and classes and calculating the feature dependency. Leave features that are of low value [31].

$$\text{K best} = n_1(\overline{x}_1 - \overline{x})^2 + n_2(\overline{x}_2 - \overline{x})^2 \tag{2}$$

This equation selects the best features that are most closely related to the output (target variable).

n_1 = number of class 1, n_2 = number of class 2, $\overline{x}$ mean all class, $\overline{x}_1$ mean class 1, $\overline{x}_2$ mean class 2.

$n_1\ (\overline{x}_1 - \overline{x})^2$ Multiply the variance by the size of Category 1, this shows how distinct Category 1 is based on its size and the variance from the overall average.

$n_2\ (\overline{x}_1 - \overline{x})^2$ Multiplying the variance by the size of class 2 obtains.

When these values are collected, the total variance between the categories for the selected feature will appear, and then the features that distinguish between the categories will be selected.

Feature selection techniques transform the used data set into a new, well-characterized data set that contains only the relevant and necessary features.

3.2.2 K-Neighbors Classifier

The K-neighbors classifier is a widely used classification algorithm known for its simplicity. It classifies data by comparing it with its nearest neighbors and makes predictions based on their proximity. This classifier is capable of parallel operations, As shown in Fig. 2. It is worth noting that the process of adjusting parameters helps improve the accuracy of the model and its effectiveness in working, and among these parameters are: the number of neighbors, the weights of the neighbors, the distance function, feature scaling and linkage solution strategies [32,33].

$$D = \sqrt{(x_a - x_b)^2 + (y_a - y_b)^2} \tag{3}$$

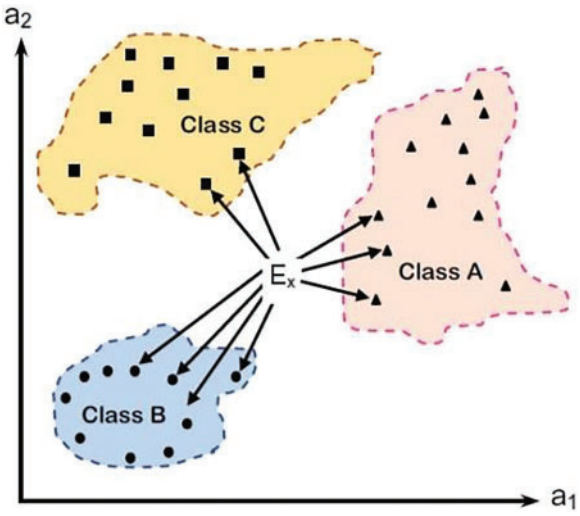

Figure 2: K-neighbors classifier

This equation calculates the straight line distance between two points in a second-dimensional space, then the difference between the corresponding coordinates is calculated, squared and summed, then the square root of the whole is taken.

$(x_a - x_b)$ is the horizontal distance a (4)

$(y_a - y_b)$ is the horizontal distance a (5)

3.2.3 Logistic Regression Classifier

The logistic regression classifier is a frequently used statistical tool that calculates probabilities to determine a final outcome based on input variables. It produces binary results: either correct or incorrect. This classifier assigns a scale to each predictor, measuring its independent contribution to changes in dependent variables [34]. The parameters of logistic regression are learning rate, batch size, controlling regularization and number of iterations. The process of adjusting these parameters helps stabilize the model and obtain good results. Note that the higher the number of repetitions, the better accuracy we obtain at the expense of time [35].

First, the linear value is calculated

$$y = \beta 0 + \beta 1x1 + \beta 2x2 + .. + \beta nxn \tag{6}$$

y = Synthetic linear value. $\beta 0$ = Objection. βi = Model coefficients for features. xi = The values of feature.

In the second stage, the logistic function is calculated

$$P(y = 1 \mid x) = 1 + e - z \tag{7}$$

$P\,(y = 1 \mid x)$: The probability of outcome y is 1 for features x. e: It is the natural basis of logarithms.

3.2.4 SGD Classifier

The SGD classifier is a linear algorithm known for its effectiveness in decision-making. It determines decision boundaries or optimal decisions by maximizing the distance between data points belonging to different feature categories using the gradient of the loss function [36,37]. In the SGD Classifier there are several parameters that affect the model's operation, which are number of iterations, learning rate, batch size and regularization. The process of adjusting these parameters helps improve the model's performance.

The way SGD classifier works is:

- Compute the loss function.

- Compute Gradient.

- Update parameters, each time the parameters are updated using the gradient.

3.2.5 Ensemble Machine Learning (Stacking)

Ensemble machine learning, specifically stacking, involves aggregating or combining multiple machine learning models. By using multiple classification algorithms, it combines the outputs of individual classifiers to make a final decision. This technique usually results in a final classifier with high accuracy. However, selecting an incorrect algorithm may result in longer training and intrusion detection times [38]. In the cumulative machine learning process, parameters play an important role in the model's performance. Among these parameters are appropriate cross-validation methods, optimal weights for base models and an effective meta-learner. The process of working and calculating cumulative machine learning is:

- Making predictions for basic models and compiling predictions.

3.3 Methodology

In this section, the proposed methodology is explained, including preprocessing and feature selection, which are essential for designing the model. This will be done by implementing the K-neighbors algorithm, logistic regression, and the SGD classifier. See Fig. 3 for a description of the IDS proposed.

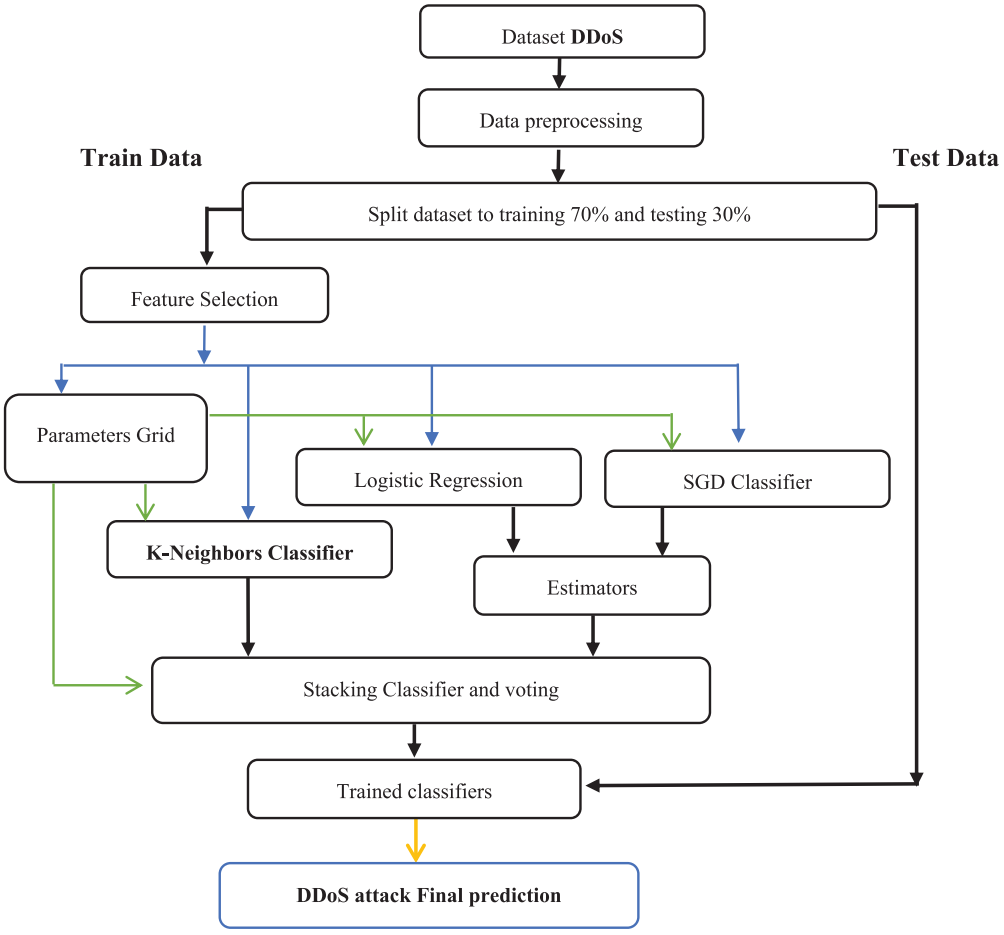

Figure 3: Model proposed structure

3.3.1 Preparing and Preprocessing the Dataset

In the proposed model, two datasets, namely CIC-IoT2023 and CIC-DDoS2019, will be used. These datasets represent real-time network traffic and include clusters of attacks. The main goal of using up-to-date datasets is to evaluate the performance of the proposed methodology, determine the accuracy of the proposed system, and assess its ability to detect false alarms. The preprocessing stages involve the following:

- Entering data into the system is the first step where the data is combined into one data set to prepare for the model.
- Identify and address missing values in rows. By [np.inf, -np.inf], [np. nan] and deleting Null and Nan from all rows of the dataset.
- Randomly dividing the data into training (70%) and testing (30%) sets.
- Employing the Sklearn variance threshold to eliminate features with low variability and identical values, as these features only reflect expected results. This step serves as the initial stage in filtering the data.

Standardizing the size of the data frames to ensure uniformity using standard scaling [39].

$$\text{Scale} = (x - m)\backslash s \tag{8}$$

x = Original Value, s = the Standard Deviation, m = The Mean.

To facilitate the learning and statistical analysis process.

- A crucial stage involves assessing the standard deviation. The proximity of data to the mean determines the variance in the data. As the standard deviation increases, the data becomes more irregular, thereby affecting expectations [40].

$$SD = \sqrt{\frac{\sum(X - U)^2}{N}} \tag{9}$$

X = the value in the data, U = value mean of the data, N = the number of points in the data.

3.3.2 Features Selection

After completing the initial stage of feature deletion, the next step involves selecting the most important features using the feature selection technique. This stage is crucial for identifying and reducing features to enhance model performance. Specifically, single features with only one variable are isolated using the Chi-squared (Chi2) technique in conjunction with the select K best algorithm. This approach allows for the identification of the most relevant features based on their statistical significance, thereby refining the dataset and improving the effectiveness of the model. See Fig. 4:

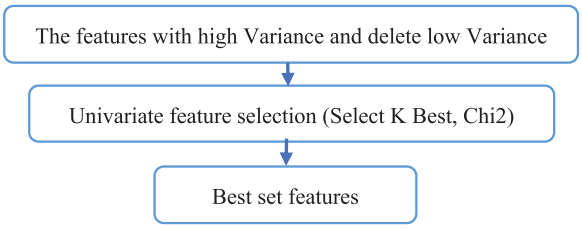

Figure 4: Overall structure to features sets

3.3.3 Proposed Model's Training

The proposed model uses a collective learning approach to create a multiple classifier by combining two algorithms, namely logistic regression and SGD classifier, with the development of the K-neighbors classifier algorithm. In addition, hyperparameters are used to determine the best parameters, aiming to achieve favorable results.

- First classification stage

In the first stage, the pre-filtered data is utilized to train the first classifier. This classifier employs two algorithms, logistic regression and SGD classifier, with the best parameters. The output of the current classification phase serves as input for the second classification stage. The final prediction is generated through the generalization stack, allowing the utilization of each algorithm's capabilities by feeding their predictions into the first stacking estimator. See Algorithm 1:

Algorithm 1: Estimators first classification

Input: Logistic Regression, SGD
Output: The estimators first
Begin
1 Get the first classifier (Logistic Regression) in the training data
2 Get the second classifier (SGD) in the training data
3 Learn a new classifier
4 Add the new classifier in estimators
5 Return estimators level 1

- Second classification stage

In the second stage, the K-neighbors classifier algorithm of the second classifier is executed using the outputs obtained from the first stacking estimator. Employing the best parameters for this algorithm, a second classifier is established. The K-neighbors classifier algorithm classifies data and generates predictions regarding the presence of data or the grouping of its individual points based on the nearest neighbor. It is based on comparing and contrasting points with the nearest neighbors and with each other, using a grid search of the training data to search for the best parameters.

- The last classifier by ensemble

In the concluding stage, an ensemble stacking classifier is created based on the outcomes of the first stage and integrated with the results of the second stage classifier using the best parameters determined through a network search of the training data. Subsequently, cross-validation is employed to maximize the efficacy of the classifier. See Algorithm 2:

Algorithm 2: Last ensemble by stacking

Input: The estimators first, K-Neighbors Classifier
Output: final prediction ensemble model
Begin
1 Get the estimators first classifier in the training data
2 Get the second classifier (K-Neighbors Classifier) in the training data
3 Learn a final ensemble model
4 End

3.3.4 The Model's Testing

The testing phase depends on the pre-selected test set. Strategies during the test differ from those during the training phase because the primary goal of use is to obtain decision stats and test the model and evaluate the results of its testing. In the testing phase, the same features as in the training phase are used. Performance is evaluated by computing a confusion matrix for each type of attack in the datasets and compute Measure the accuracy, false alarm rate, detection rate, F-measure, recall and precision. See Algorithm 3:

Algorithm 3: The model testing

Input: Testing samples selected from dataset
Output: A confusion matrix and decision statistics
Begin
1 Testing the model on all instances of the testing dataset
2 Comparing every output class of the instances with the real one of the testing dataset.
3 Compute confusion matrix.
4 Compute Measure the accuracy, precision, recall, detection rate, F-measure and false alarm rate.
5 End

4 Experiment Details

This section presents a thorough explanation of the performance evaluation carried out on the proposed model for real time DDoS detection on Internet of Things. The evaluation process comprises two levels: the first level comprises removing low-contrast features and selecting the most significant essential features. The second level involves constructing the proposed model by combining the K-neighbors algorithm with other algorithmic classifiers. This integration aims to achieve high system accuracy, reduce training time, and enhance intrusion detection capabilities.

4.1 Dataset

The datasets used and the characteristics of each dataset are explained.

4.1.1 CIC-IoT2023 Dataset

The CIC-IoT2023 dataset is a real-time dataset from the Canadian Cyber Security Institute. It contains large-scale IoT attacks, consisting of 7 categories of attacks on IoT devices, and consisting of 47 features [41]. Due to the huge volume of data in this data set, part of it was used to train the model, and only for DDoS attacks. 25 features with high contrast were selected using the proposed feature selection technique. The dataset was split into training and testing sets using cross-validation, with 876,169 instances allocated to training and 375,501 instances allocated to testing. One of the challenges facing this dataset is that it contains a very large number of restrictions. Which may be an obstacle in training if appropriate techniques are not used. This data set is not biased toward any type of attack. It contains a large number of attacks, but for our part, DDoS attacks were chosen because our methodology highlights this type of attack on Internet of Things networks and devices.

4.1.2 CIC-DDoS 2019 Dataset

The CIC-DDoS 2019 dataset, containing 88 features and 162,590 total data records, was utilized for model training. This dataset includes samples of normal behavior and attacks [42]. 15 features with high contrast were selected using the proposed feature selection technique. The training set comprised 110,729 instances, with 47,456 instances allocated for testing purposes. It is worth noting that this dataset is biased towards DDoS attacks. It does not contain any other types of attacks. Its use is due to the fact that the proposed model highlights DDoS attacks only on Internet of Things networks.

4.2 System Specifications

The experimental setup utilized the following system specifications: a 64-bit operating system, Intel Core i5-12350G CPU @1.8 GHz, 32 GB RAM, running on Windows 11, Python 3.10.4, and

utilizing Jupyter Notebook (Server: 6.4.11). The proposed IDS were trained and evaluated using the CIC-IoT2023 and CIC-DDoS2019 datasets to detect DDoS attacks on IoT networks.

5 Experiment Results

The efficacy of the proposed methodology was assessed through the calculation of various performance metrics, including model accuracy, detection rate, and precision, recall, F1-score, and detection time.

Accuracy: It is a measure to evaluate the accuracy of the model in correct predictions, whether positive or negative. It is worth noting that in some cases of imbalance, the accuracy is misleading, and therefore other measures are used such as (Recall and F1-score) to obtain a good evaluation. According to [43], it is calculated using the following equation:

$$\text{Accuracy (Acc)} = \frac{\text{TN} + \text{TP}}{\text{All (TP} + \text{TN} + \text{FP} + \text{FN)}} \tag{10}$$

Precision: It is a measure for evaluating classification performance, as it evaluates the percentage of positive predictions and their accuracy, which were obtained after implementing the model. It is considered an important measure because it is used in decisive predictions and must be balanced with the recall measure [43]. It is calculated using the following equation:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{11}$$

Recall: It is a measurement of the model's classification, showing the percentage of actual positive correct predictions. It shows the model's ability to detect positive samples. Its importance also lies in dangerous predictions that cannot be wrong, such as categorizing types of diseases or detecting attacks [43]. It is calculated using the following equation:

$$\text{Recall or DR} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{12}$$

F-score: It is a measure that works on the basis of calculating Precision and Recall, combining them to provide a balanced assessment. It gives a more comprehensive picture of the model's performance. It can also be used in cases where there is a balance between positive and negative predictions [43]. It is calculated using the following equation:

$$\text{F} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{13}$$

Detection rate: It works like a recall measure, as it calculates the correct positive predictions after executing the model, and it is also used in critical predictions that do not accept errors.

5.1 Evaluation of the Proposed Methodology Using the CIC-IoT2023 Dataset

To assess the first performance of the proposed methodology, the CIC-IoT2023 dataset was utilized. In the first stage, 40 features with high contrast were selected from the total features [0 1 2 3 4 5 6 7 8 9 10 11 14 15 16 17 18 19 20 24 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45]. This contributes to preparing features for the second stage of feature selection, and also helps reduce complexity in the data and increase the speed of training the proposed model. and In the second stage, the best 25 features from 40 features [0 1 3 4 5 8 11 13 15 16 18 20 24 27 28 29 30 31 32 34 35 36 37 38

39] were selected using the (select K best) technique. In Table 2 and Fig. 5, the results of the confusion matrix of DDoS attacks resulting from the implementation of our proposed model are presented.

Table 2: Confusion matrix to CIC-IoT2023 dataset

Actual class	Predicted class	
	Normal	Attack
Normal	11619	20
Attack	111	363751

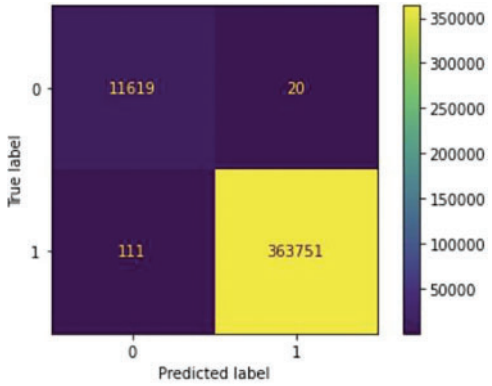

Figure 5: Confusion matrix to CIC-IoT2023

Fig. 5 and Table 2 showcase the predictions obtained from our proposed model, highlighting its capability to minimize false alarms and enhance the detection accuracy of genuine attacks. Furthermore, Fig. 6 and Table 3 delineate the outcomes of the proposed model as obtained from the confusion matrix.

Figure 6: Evaluate the first performance for the model proposed in the CIC-IoT2023

Table 3 illustrates the accuracy of the proposed model in detecting DDoS attacks in CIC-IoT2023, demonstrating an increase in detection rates and a reduction in error rates. Additionally, it depicts the time invested in intrusion detection. Challenges encountered while utilizing this dataset include dealing

with a large number of features, selecting features with high variance, and the associated complexities in processing them.

Table 3: The results of the proposed model for CIC-IoT2023 dataset

Acc	Recall	Precision	D.R	FAR	F1_score	E.R	Time detection
99.965%	99.828%	99.058%	99.82%	0.003%	2.0	0.003%	0.20 s

5.2 Evaluation of the Proposed Methodology Using the CIC-DDoS2019 Dataset

In the second performance of the proposed methodology, the CIC-DDoS2019 dataset was utilized. In the first stage, 68 features with high contrast were selected from the total features [0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 33 34 35 36 37 38 39 40 41 43 44 46 47 48 50 51 52 53 55 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73]. And in the second stage, the best 15 features from the 68 features [3 4 11 17 18 19 34 35 38 44 46 47 48 55 67] were selected using the (select K best) technique. In Table 4 and Fig. 7, the results of the confusion matrix of DDoS attacks resulting from the implementation of our proposed model are presented.

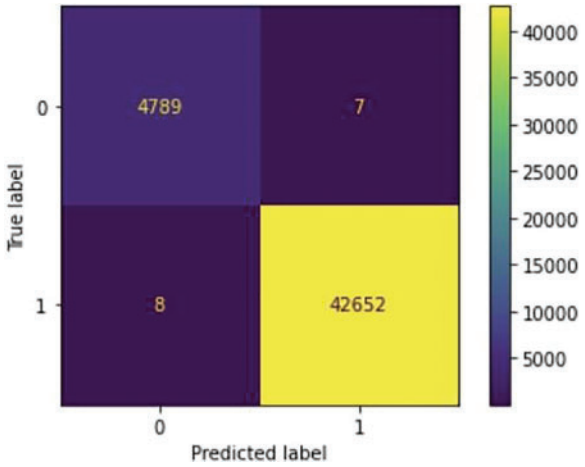

Figure 7: Confusion matrix to CIC-DDoS2019

Table 4: Confusion matrix to CIC-DDoS2019 dataset

Actual class	Predicted class	
	Normal	Attack
Normal	4789	7
Attack	8	42652

Fig. 7 and Table 4 illustrate the predictions obtained from our proposed model by CIC-DDoS2019 dataset, demonstrating its effectiveness in reducing false alarms and improving the detection accuracy of genuine attacks. Additionally, Fig. 8 and Table 5 provide a detailed breakdown of the results obtained from the confusion matrix by our proposed model.

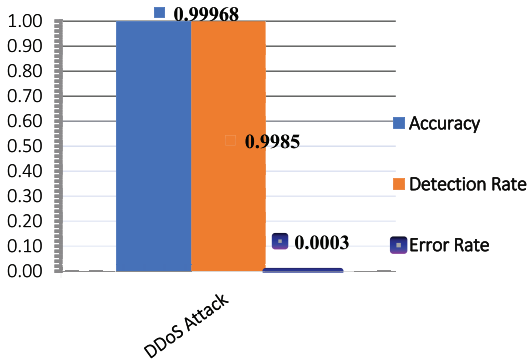

Figure 8: Evaluate the second performance for the model proposed in the CIC-DDoS2019

Table 5: The results of the proposed model for CIC-DDoS2019 dataset

Acc	Recall	Precision	D.R	FAR	F1_score	E.R	Time detection
99.968%	99.854%	99.833%	99.854%	0.001%	2.0	0.003%	0.23 s

In Table 5, the accuracy of the proposed model in detecting DDoS attacks was demonstrated in CIC-DDoD2019, showing the quality of the results in the detection rate and reducing error rates. These results demonstrate the superiority of the proposed model in terms of accuracy, detection rate, and reducing false alarms.

5.3 Evaluation of the Proposed Methodology on Types of DDoS Attacks Using the CIC-DDoS2019 Dataset

For the purpose of knowing the effect of different types of DDoS attacks on the performance of the proposed model, an evaluation of the performance of the proposed model was conducted on different types of DDoS attacks separately, since the data set used contains several types of DDoS attacks.

Table 6 shows the performance of the proposed model on a different set of DDoS attacks. The results show that the model obtained high accuracy in detecting different attacks. In order to adapt the proposed model to new and emerging attack vectors, the training dataset must be updated continuously to detect new attack patterns.

Table 6: The results of the proposed model on types of DDoS attacks for CIC-DDoS2019 dataset

Type attack	ACC	Recall	Precision	Time detection
DDoS (LDAP)	99.972%	99.894%	99.857%	0.19 s
DDoS (DoS_DNS)	99.961%	99.843%	99.797%	0.22 s
DDoS (MSSQL)	99.980%	99.892%	99.916%	0.30 s
DDoS (DoS_MSSQL)	99.964%	99.774%	99.792%	0.24 s
DDoS (NetBIOS)	99.959%	99.749%	99.765%	0.20 s

5.4 Comparison with Literature Review

5.4.1 Comparing the Performance Evaluation of the Proposed Methodology with Previous Studies

In Table 7, a comparative analysis with previous research was conducted based on performance measures.

Table 7: Comparison between the literature review and proposed IDS

Ref.	dataset	Method	ACC	Recall	Precision
[10]	CIC-DDoS2019	CNN	99.9%	–	–
[11]	CIC-DDoS2019	KNN, DT, SGB, NB, SVM, RF	97%, 96%, 99%, 53%, 97%, 96%.	99%, 98%, 99%, 66%, 99%, 98%.	98.8%, 98%, 99%, 65%, 99%, 98%
[12]	(InSDN, CIC-IDS2017, and CIC-DDoS2019)	DCNN	99.99%	99.99%	99.99%
[13]	CIC-DDoS2019	DNN, CNN and RNN	99.99%	–	–
[14]	CIC-IoT2023	LSTM algorithm	98.75%	–	–
[15]	CICIoT2023	ARF-ADWIN, SRPs-DDM and ARF-DDM, KNN-ADWIN	99.33%	96.53%	99.88%
[16]	CICIoT2023	multi-class classification	96.56%	–	–
Proposed model	**CIC-IoT2023**		**99.965%**	**99.828%**	**99.058%**
	CIC-DDoS2019		**99.957%**	**99.854%**	**99.72%**

Upon comparing the results of the proposed methodology with those of previous studies, it becomes evident that our approach excels in terms of intrusion detection accuracy, minimized system errors, and reduced intrusion detection time, in addition to the reduction in features. Notably, the proposed model demonstrates superior performance in terms of training speed and intrusion detection accuracy, which can be attributed to the streamlined feature reduction process and the utilization of efficient algorithms. This has led to remarkably high system accuracy compared to prior works.

5.4.2 Comparison with Related Works Will Be Made on the Basis of Statistical Tests and Confidence Intervals, to Determine the Accuracy of the Proposed Model

Initially, a comparison was made between the first performance evaluation, which is based on the (CIC-IoT2023) data set, with previous studies using the same dataset, namely [15], and the results were as follows: T-statistic = 1.0596 and Confidence Interval = (0.6044, 1.8938). Through the results obtained, The T-statistic was a positive value, and this indicates that the average accuracy of the proposed model is higher than the average accuracy of the mentioned study.

Also, a comparison was made between the second performance evaluation based on the (CIC-DDoS2019) data set, with previous studies that used the same dataset [13], and the results were as follows: T-statistic = 0.9849 and Confidence Interval = (0.6007, 1.8936). The value of the T-statistic indicates positive, and this proves that the average accuracy of the proposed model is higher than the average accuracy of the mentioned study, but the difference in superiority is very small.

Through the first comparison with performance evaluation metrics and the second comparison of the statistical test with previous studies, the superiority of our proposed model in accurately detecting DDoS attacks on Internet of Things networks is proven.

5.5 Challenges and Limitations to the Proposed Methodology

In the proposed work, a set of challenges were faced, which were overcome and a highly accurate model was obtained. The first challenge faced is dealing with the selected data sets, including the method of analyzing the data and getting rid of ambiguous and useless data, the method of extracting only DDoS attacks from the data set, because our proposed research only highlights DDoS attacks, and also the method of selecting relevant features was done through two methods. This played an effective role in the model training process. The second challenge is how to deal with the parameters of the selected algorithms because parameters play an essential role in the work of algorithms. This was overcome through the process of analyzing data and conducting numerous experiments on the model to reach the final parameters in the proposed model. It is necessary to mention the limitations that exist in our proposed research. These limitations can be addressed in future ideas. It is to use a newer data set than the one that was used, and also to expand the detection of all cyber attacks instead of dealing only with the DDoS attack. The model is constantly updated to keep pace with the development of attackers.

6 Conclusions

Numerous systems have been proposed to detect cyber attacks on the Internet of Things, especially DDoS attacks, due to the recent challenges posed by technological advancements, data proliferation, and the wide range of applications in use. These systems employ various methods and machine learning algorithms, primarily relying on data mining techniques. In this study, we introduced a two-stage model for feature selection, which utilizes the contrast threshold method and combines select K best with Chi2. Additionally, we improved the K-neighbors algorithm by using ensemble machine learning (stacking) with logistic regression and the SGD classifier algorithm. Our proposed methodology was evaluated using the CIC-IoT2023 and DDoS2019 datasets, resulting in remarkably high accuracy rates compared to previous works. Specifically, our model achieved accuracy rates of 99.965% and 99.957%, with intrusion detection times of 0.20 and 0.23 s, respectively. Furthermore, a comparative analysis with recent works confirmed the superiority of our proposed methodology. One of the challenges faced was the large amount of data used for training and the way it was processed. The challenge was how to select only relevant features to help reduce training time. We recommend taking advantage of the proposed model to improve cyber security for Internet of Things networks because the proposed model provides a rapid response to DDoS attacks. The model can be integrated with existing security systems. It is worth noting that it can be applied in several fields, including in the field of health and in smart cities. As for future work, we suggest several works to develop the current research, including, we recommend testing the proposed model in real environments after it has been trained on a real data set. We also recommend discovering new data, such as (CIC-IoV2024), that contains new types of cyber attacks to keep pace with the development of attackers and face the challenges of processing

data and choosing relevant features. We also recommend using deep learning algorithms and knowing the accuracy of the model detection.

Acknowledgement: The authors would like to thank all those who helped us morally in this work.

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: Zaed Mahdi, Rana Zaki; data collection: Nada Abdalhussien; analysis and interpretation of results: Naba Mahmood, Zaed Mahdi; draft manuscript preparation: Zaed Mahdi, Nada Abdalhussien, Naba Mahmood. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The data is available on https://www.unb.ca/cic/datasets/iotdataset-2023.html (accessed 01/06/2024) and https://www.unb.ca/cic/datasets/ddos-2019.html (accessed 01/06/2024).

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

References

[1] S. Sharma, V. Kumar, and K. Dutta, "Multi-objective optimization algorithms for intrusion detection in IoT networks: A systematic review," *Internet Things Cyber-Phys. Syst.*, vol. 4, no. 1, pp. 258–267, 2024. doi: 10.1016/j.iotcps.2024.01.003.

[2] I. A. Zahid, S. A. Hussein, and S. M. Mahdi, "Measuring individuals cybersecurity awareness based on demographic features," *Iraqi J. Electr. Electron. Eng.*, vol. 4, no. 1, pp. 58–67, 2023. doi: 10.37917/ijeee.

[3] M. H. Ali *et al.*, "Threat analysis and distributed denial of service (DDoS) attack recognition in the internet of things (IoT)," *Electronics*, vol. 11, no. 3, pp. 494, 2022. doi: 10.3390/electronics11030494.

[4] Z. Zhao *et al.*, "DDoS family: A novel perspective for massive types of DDoS attacks," *Comput. Secur.*, vol. 138, pp. 103663, 2024. doi: 10.1016/j.cose.2023.103663.

[5] 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, 2024. doi: 10.1007/s11042-023-15977-8.

[6] C. Liu, Z. Gu, and J. Wang, "A hybrid intrusion detection system based on scalable K-Means+ random forest and deep learning," *IEEE Access*, vol. 9, pp. 75729–75740, 2021. doi: 10.1109/ACCESS.2021.3082147.

[7] M. Wang, K. Zheng, Y. Yang, and X. Wang, "An explainable machine learning framework for intrusion detection systems," *IEEE Access*, vol. 8, pp. 73127–73141, 2020. doi: 10.1109/ACCESS.2020.2988359.

[8] X. Cai *et al.*, "Stability analysis of networked control systems under DoS attacks and security controller design with mini-batch machine learning supervision," *IEEE Trans. Inf. Forensics Secur.*, vol. 19, pp. 3857–3865, 2023. doi: 10.1109/TIFS.2023.3347889.

[9] J. Liu *et al.*, "Data mining and information retrieval in the 21st century: A bibliographic review," *Comput. Sci. Rev.*, vol. 34, no. 4, pp. 100193, 2019. doi: 10.1016/j.cosrev.2019.100193.

[10] V. Tila Patil, S. Shivaji Deore, K. Ibrahim Osamah, S. Algburi, and H. Hamam, "IoT-defender: A convolutional approach to detect DDoS attacks in internet of things," *Int. J. Comput. Digit. Syst.*, vol. 16, no. 1, pp. 1–11, 2024.

[11] A. M. Salama, M. A. Mohamed, and E. AbdElhalim, "Enhancing network security in IoT applications through DDoS attack detection using ML," *Mansoura Eng. J.*, vol. 49, no. 3, pp. 10, 2024. doi: 10.58491/2735-4202.3181.

[12] V. Hnamte and J. Hussain, "An efficient DDoS attack detection mechanism in SDN environment," *Int. J. Inf. Technol.*, vol. 15, no. 5, pp. 2623–2636, 2023. doi: 10.1007/s41870-023-01332-5.
[13] R. Ma, Q. Wang, X. Bu, and X. Chen, "Real-time detection of DDoS attacks based on random forest in SDN," *Appl. Sci.*, vol. 13, no. 13, pp. 7872, 2023. doi: 10.3390/app13137872.
[14] A. I. Jony and A. K. B. Arnob, "A long short-term memory based approach for detecting cyber attacks in IoT using CIC-IoT2023 dataset," *J. Edge Comput.*, vol. 3, no. 1, pp. 28–42, 2024. doi: 10.55056/jec.648.
[15] Y. K. Beshah, S. L. Abebe, and H. M. Melaku, "Drift adaptive online DDoS attack detection framework for IoT system," *Electronics*, vol. 13, no. 6, pp. 1004, 2024. doi: 10.3390/electronics13061004.
[16] S. Abbas *et al.*, "Evaluating deep learning variants for cyber-attacks detection and multi-class classification in IoT networks," *PeerJ Comput. Sci.*, vol. 10, pp. e1793, 2024. doi: 10.7717/peerj-cs.1793.
[17] A. Heidari and M. A. Jabraeil Jamali, "Internet of Things intrusion detection systems: A comprehensive review and future directions," *Cluster Comput.*, vol. 26, no. 6, pp. 3753–3780, 2023. doi: 10.1007/s10586-022-03776-z.
[18] R. Vinayakumar, M. Alazab, K. Soman, P. Poornachandran, A. Al-Nemrat and S. Venkatraman, "Deep learning approach for intelligent intrusion detection system," *IEEE Access*, vol. 7, pp. 41525–41550, 2019. doi: 10.1109/ACCESS.2019.2895334.
[19] S. M. Kasongo, "A deep learning technique for intrusion detection system using a recurrent neural networks based framework," *Comput. Commun.*, vol. 199, no. 1, pp. 113–125, 2023. doi: 10.1016/j.comcom.2022.12.010.
[20] X. Cai, K. Shi, Y. Sun, Y. Soh, and Z. Tian, "Performance analysis and design of intelligent optimising integral-based event-trigger control for autonomous ground vehicles under DoS attacks," *IEEE Trans. Intell. Vehicles*, vol. 9, no. 1, pp. 2149– 2159, 2023. doi: 10.1109/TIV.2023.3317238.
[21] T. Kaur, V. Malhotra, and D. Singh, "Comparison of network security tools-firewall, intrusion detection system and honeypot," *Int. J. Enhanced Res. Sci. Technol. Eng.*, vol. 3, no. 2, pp. 200–204, 2014.
[22] R. M. Zaki and H. B. A. Wahab, "4G network security algorithms: Overview," *Int. J. Interact. Mobile Technol.*, vol. 15, no. 16, pp. 127, 2021. doi: 10.3991/ijim.v15i16.24175.
[23] G. Mumtaz *et al.*, "Classification and prediction of significant cyber incidents (SCI) using data mining and machine learning (DM-ML)," *IEEE Access*, vol. 11, pp. 94486–94496, 2023. doi: 10.1109/ACCESS.2023.3249663.
[24] A. Sahasrabuddhe, S. Naikade, A. Ramaswamy, B. Sadliwala, and P. Futane, "Survey on intrusion detection system using data mining techniques," *Int. Res. J. Eng. Technol.*, vol. 4, no. 5, pp. 1780–1784, 2017.
[25] M. Al-Janabi, M. A. Ismail, and A. H. Ali, "Intrusion detection systems, issues, challenges, and needs," *Int. J. Comput. Intell. Syst.*, vol. 14, no. 1, pp. 560–571, 2021. doi: 10.2991/ijcis.d.210105.001.
[26] B. H. Nguyen, B. Xue, and M. Zhang, "A survey on swarm intelligence approaches to feature selection in data mining," *Swarm Evol. Comput.*, vol. 54, no. 3, pp. 100663, 2020. doi: 10.1016/j.swevo.2020.100663.
[27] A. M. Ali and A. K. Farhan, "A novel multi-biometric technique for verification of secure e-document," *Int. J. Electr. Comput. Eng.*, vol. 14, no. 1, pp. 662–671, 2024. doi: 10.11591/ijece.v14i1.pp662-671.
[28] A. Bommert, X. Sun, B. Bischl, J. Rahnenführer, and M. Lang, "Benchmark for filter methods for feature selection in high-dimensional classification data," *Comput. Stat. Data Anal.*, vol. 143, no. 3, pp. 106839, 2020. doi: 10.1016/j.csda.2019.106839.
[29] S. H. Jafer, "Optimize network intrusion detection system based on PCA feature extraction and three naïve bayes classifiers," *J. Phys.: Conf. Ser.*, vol. 2322, pp. 012092. doi: 10.1088/1742-6596/2322/1/012092.
[30] Y. S. Ambarwati and S. Uyun, "Feature selection on magelang duck egg candling image using variance threshold method," in *2020 3rd Int. Semin. Res. Inf. Technol. Intell. Syst. (ISRITI)*, Yogyakarta, Indonesia, Dec. 10–11, 2020, pp. 694–699.
[31] R. T. Merlin and R. Ravi, "Empowering smart city IoT network intrusion detection with advanced ensemble learning-based feature selection," *Int. J. Electr. Electron. Res.*, vol. 12, no. 2, pp. 367–374, 2024. doi: 10.37391/IJEER.

[32] H. A. Abu Alfeilat *et al.*, "Effects of distance measure choice on k-nearest neighbor classifier performance: A review," *Big Data*, vol. 7, no. 4, pp. 221–248, 2019. doi: 10.1089/big.2018.0175.
[33] V. Prasatha *et al.*, "Effects of distance measure choice on KNN classifier performance-a review," arXiv preprint arXiv: 1708.04321, 2017.
[34] F. A. M. Solomon, G. W. Sathianesan, and R. Ramesh, "Logistic regression trust-a trust model for internet-of-things using regression analysis," *Comput. Syst. Sci. Eng.*, vol. 44, no. 2, pp. 1125–1142, 2023. doi: 10.32604/csse.2023.024292.
[35] S. Dreiseitl and L. Ohno-Machado, "Logistic regression and artificial neural network classification models: A methodology review," *J. Biomed. Inform.*, vol. 35, no. 5–6, pp. 352–359, 2002. doi: 10.1016/S1532-0464(03)00034-0.
[36] O. Osho and S. Hong, "An overview: Stochastic gradient descent classifier, linear discriminant analysis, deep learning and Naive Bayes classifier approaches to network intrusion detection," *Int. J. Eng. Tech. Res.*, vol. 10, no. 4, pp. 294–308, 2021.
[37] D. Kalimeris *et al.*, "SGD on neural networks learns functions of increasing complexity," arXiv preprint arXiv: 1905.11604, 2019.
[38] M. Lu *et al.*, "A stacking ensemble model of various machine learning models for daily runoff forecasting," *Water*, vol. 15, no. 7, pp. 1265, 2023. doi: 10.3390/w15071265.
[39] M. M. Ahsan, M. P. Mahmud, P. K. Saha, K. D. Gupta, and Z. Siddique, "Effect of data scaling methods on machine learning algorithms and model performance," *Technologies*, vol. 9, no. 3, pp. 52, 2021. doi: 10.3390/technologies9030052.
[40] X. Cui and L. Dong, "An algorithm to compute composition skyline based on standard deviation," *AIP Conf. Proc.*, vol. 2122, no. 1, pp. 020030. doi: 10.1063/1.5116469.
[41] H. Q. Gheni and W. L. Al-Yaseen, "Two-step data clustering for improved intrusion detection system using CICIoT2023 dataset," *e-Prime*, vol. 9, pp. 100673, 2024.
[42] D. C. Can, H. Q. Le, and Q. T. Ha, "Detection of distributed denial of service attacks using automatic feature selection with enhancement for imbalance dataset," in *Intell. Inf. Database Sys. 13th Asian Conf.*, Phuket, Thailand, Apr. 7–10, 2021, pp. 386–398.
[43] D. M. Powers, "Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation," arXiv preprint arXiv:2020.16061, 2010.