2023 International Conference on Smart Cities, Automation & Intelligent Computing Systems
Bali, Indonesia, 6-8 December 2023

2023 3rd International Conference on Smart Cities, Automation & Intelligent Computing Systems (ICON-SONICS) | 979-8-3503-1829-6/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICON-SONICS59898.2023.10435006

Development of Intrusion Detection Models for IoT Networks Utilizing CICIoT2023 Dataset

1st Nadia Thereza
Department of Electrical Engineering
Universitas Indonesia
Depok, Indonesia
nadia.thereza@ui.ac.id

2nd Kalamullah Ramli
Department of Electrical Engineering
Universitas Indonesia
Depok, Indonesia
kalamullah.ramli@ui.ac.id

***Abstract*— The Internet of Things (IoT) is a rapidly growing technology that enables devices to communicate and exchange data with minimal human intervention. However, this growth increases the volume of sensitive data, making it more vulnerable to security attacks. DDoS is a perilous form of attack that targets IoT networks frequently. Intrusion detection systems (IDSs) are a solution for protecting IoT devices by monitoring network activities and detecting real-time threats and attacks. However, implementing IDS in IoT networks presents several challenges, including power and memory constraints imposed on IoT devices and implementation datasets requiring greater comprehensiveness to accurately define the features of IoT networks. Thus, this study developed intrusion detection models using lightweight ML algorithms, such as decision tree, k-nearest neighbors, random forest, and Naïve Bayes, to identify network DDoS attacks. The latest dataset, CICIoT2023, which includes multiple attacks unavailable in previous IoT datasets, was utilized. We evaluated the model's performances using accuracy, false positive rate, F1-score, recall, precision, and training and testing time usage. The results show that the random forest and decision tree models outperformed other detection models with 100% accuracy. Regarding time usage, the decision tree model outperformed other models, which could classify 2,926,588 instances in 1 second.**

Keywords—intrusion detection system, internet of things, DDoS, attack, dataset, machine learning

I. INTRODUCTION

The Internet of Things (IoT) is a rapidly developing technological innovation that allows devices to recognize their surroundings, connect via the internet, and exchange data with minimal human intervention, enabling remote management and administration [1-5]. The internet of things (IoT) is anticipated to be utilized by billions of devices globally in the coming years, augmenting networks' quantity and magnitude [6, 7]. This escalation increases the volume of sensitive and confidential data produced, making it more susceptible to being targeted and abused. IoT systems, which include embedded devices and advanced servers, are vulnerable to security flaws [8], such as Mirai, denial-of-service (DoS), distributed denial-of-service (DDoS), and brute force attacks. These malicious attacks can cause damage to Internet of Things services and smart-environment applications [9].

DDoS is a harmful type of attack. For instance, on July 27, 2023, an attack on Kenya's digital systems and infrastructure made it challenging to access services provided by the Kenyan government, specifically the e-Citizen portal. Apart from that, several partially affected companies include electricity supply, digital banking, mobile money, railways, transportation, and licensing services. This attack was known to be identified as a DDoS attack. DDoS is an attack carried out by flooding a network with unwanted traffic so that it will damage regular traffic and services on the network. Aside from targeting servers and other network infrastructure, attackers frequently seize control of remote services to prevent system and network proprietors from accessing them. Attackers employ a multitude of computer system devices to generate substantial volumes of malicious network traffic [10].

Intrusion detection systems (IDS) are preventive technologies that detect and classify network intrusions and security policy violations. It is prominent for protecting genuine IoT devices through monitoring network activities for unauthorized activity and reporting to oversight entities, and it can also detect data transmissions in real-time [11, 12]. IDS is embedded within a firewall connected to IoT servers, forming a network architecture. The IDS architecture in an IoT network is depicted in Fig. 1.

Cybersecurity solutions frequently use artificial intelligence to implement intrusion detection systems [13]. These systems are developed using deep learning (DL) and machine learning (ML) techniques. Implementation of intrusion detection systems is challenging, especially in IoT networks. An IoT system's security relies on individual devices' security, a matter influenced by power and memory considerations. The constraints imposed by power and memory exertions present security challenges within IoT systems. In order to tackle these issues, it is imperative to develop lightweight security approaches that can be universally applied across various IoT domains while simultaneously satisfying security prerequisites without compromising the quality of service [4]. The application of DL techniques has high complexity and high utilization of hardware computational resources [14, 15]. Therefore, implementing DL on IDS for IoT networks is a challenge. In this study, we decided to build an IDS model using the ML methods, as ML methods provide high-accuracy performance, require less time to construct a model, and are less dependent on hardware computational resources [16].

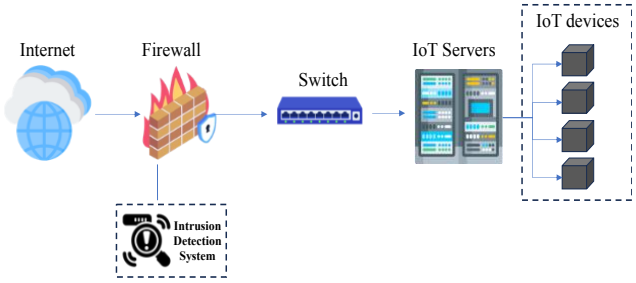

Fig. 1. Intrusion detection system architecture

In this study, we aim to build intrusion detection system models for detecting DDoS attacks on IoT networks using machine learning algorithms. In developing effective intrusion detection models, dataset characteristics representing real IoT networks are necessary. Previous research primarily relied on datasets that required greater comprehensiveness to accurately define the features of the Internet of Things networks. The main contributions of this research are that we implemented the latest actual IoT datasets to build intrusion detection models, namely the CICIoT2023 dataset, which includes multiple attacks not available in other IoT datasets [17], and we evaluated and analyzed the performance of ML-based intrusion detection models in detecting attacks. The evaluation metrics employed in this study to analyze performance models include false positive rate (FPR), F1-score, recall, precision, and accuracy.

The remaining section of this paper is organized as follows: Section 2 explains the literature review of related studies. Section 3 provides a detailed research methodology that explains the CICIoT2023 dataset, DDoS attack, feature selection, machine learning algorithms, and evaluation metrics. Section 4 discusses the results and discussion of this study. Finally, the conclusion of the study is presented in Section 5.

II. LITERATURE REVIEW

A comprehensive examination of existing literature was executed to ascertain the disparity between the current research and previous research endeavors. The fundamental dissimilarity between this particular investigation and prior studies lies in applying the most recent dataset generated, CICIoT2023, through the empirical inquiry conducted by Neto *et al*. Additionally, in this study, we focused on analyzing the time spent which the models require for training and testing datasets.

A real-time intrusion detection system was developed by Hattarki *et al*. [11] to enhance security in IoT networks. This system was constructed using the random forest technique. The low latency of the system is attributed to the utilization of web sockets, thus rendering it real-time. Although the model exhibited a low accuracy rate for various attacks, including the Scan attack and MITM, resulting in numerous false alarms, as is customary for such models. Nevertheless, real-time testing attained precision, approximately 91.178%, and accurately identified most attacks.

Karanfilovska *et al.* provided a complete model [18] developed using supervised and unsupervised machine learning on the NF-ToN-IoT-v2 dataset. When employing a custom-made automated ML (AE2EML), the Random Forest Classifier achieved an impressive F-Score of 98.6%.

Illavarason and Sundaram [19] examined the detection accuracy using different combinations of feature selection. The methodology employed for feature selection includes a correlation-based approach, information gain ratio-based selection, and principal component analysis. The attacks were classified using the Naïve Bayes, Support Vector Machine, DT, Neural Network, and K-nearest Neighbor algorithms. Integrating the selection technique with machine learning algorithms resulted in a substantial level of accuracy, with correlation-based and KNN achieving 98.57% accuracy, PCA and KNN achieving 99.10%, and Information Gain Ratio and KNN reaching 99.1%.

The research paper [20] put forth a host-based intrusion detection system (HIDS) that utilizes a modified vector space representation (MVSR) N-gram and MLP model in order to protect the Internet of Things (IoT). To carry out the investigation, the Australian Defence Force Academy Linux Dataset (ADFA-LD) was employed, consisting of vulnerabilities and attacks on various applications. The feature selection process involved utilizing and comparing the Mutual Information (MI) technique and Principal Component Analysis (PCA).

III. PROPOSED METHODOLOGY

In this segment, we outline the framework for our proposed method, as depicted in Fig. 2. In this research, we utilized Python, a high-level programming language, to build machine learning-based intrusion detection models. We implemented Sci-kit-learn using Python 3.9 and 64-bit ARM CPU architecture. There are four main stages to the methodology that we present. The first stage was the pre-processing dataset stage, namely cleaning the dataset by removing features that cannot be calculated, such as features not in integer or float format. This process was also the process of balancing features in the dataset. The second stage was to transform the dataset into a more valuable one by eliminating features that were not relevant. This stage was carried out to improve model performance regarding accuracy, false positive rate, F1-score, recall, precision, and the time required for testing. The dataset, which results from feature selection, was then continued to the third stage, namely the training and testing stage. The dataset was trained and tested using supervised machine learning algorithms, including the decision tree (DT), K-Nearest Neighbors (KNN), random forest (RF), and Naïve Bayes (NB) algorithms. The training and testing process results were then evaluated and analyzed to see which model had the highest level of performance. Overall, research conclusions will be drawn from the evaluation and analysis results.

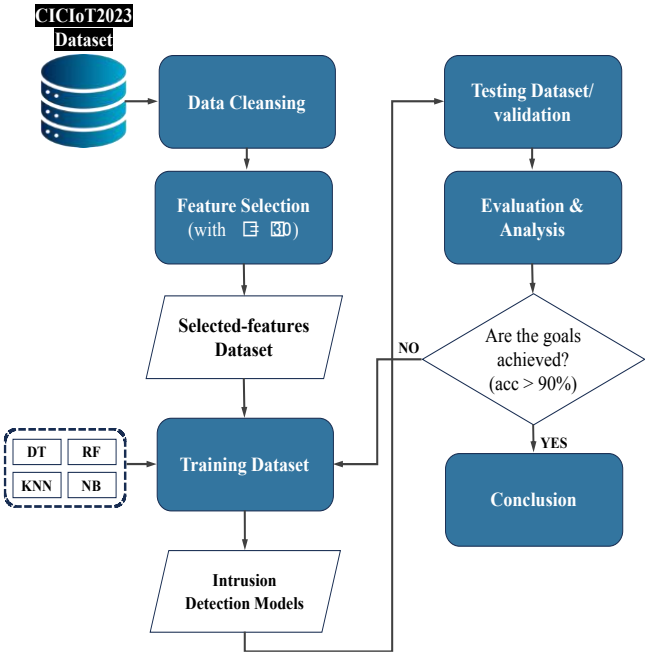

Fig. 2. A conceptual framework of proposed method

A. CICIoT2023 Dataset

The dataset used in this study is the latest dataset, first published in 2023 by Neto *et al.* [17], which they named CICIoT2023. This dataset was obtained using a more expansive IoT network topology, consisting of several real IoT devices acting as intruders and targets.

IoT attack datasets are represented in rows and columns, where one row shows an instance, which can be malicious attack data or normal (benign) data. Classes serve to group instances. The CICIoT2023 dataset consists of eight classes, seven classes of attacks, and one class of normal or benign samples. As shown in Fig. 3, this dataset consists of 33 types of attacks grouped into seven categories: Mirai, spoofing, brute force, web-based, recon, DoS, and DDoS. The total number of rows in this dataset is more than 30 million; this means that in this dataset, there are more than 30 million instances that can be either attacks or benign, where the instance data is obtained from the capture process from the IoT environment. The number of rows for Mirai attacks is 2,634,124 instances; for spoofing attacks, 486,504 samples; for brute force attacks, 130,64 samples; for web-based attacks, 24,829 instances; for reconnaissance attacks, 354,565 samples; for DoS attacks, 8,090,738 instances; and for DDoS attacks, 33,984,560 instances. The columns in the dataset show the attributes or characteristics of the instances. Each instance has attributes. The specifications and values of these attributes are called features. A dataset has many features that contain traffic information. CICIoT2023 comprises 46 features, including $flow\_duration$, $header\_length$, $protocol\ type$, $rate$, ack_count, IPv, $variance$, etc. The more features there are in a dataset, the higher the dimensions of the dataset. However, the higher the dimensions, the higher the diversity of the data, which means that if irrelevant features are not eliminated for developing the intrusion detection model, this diversity will affect the detection system's accuracy level.

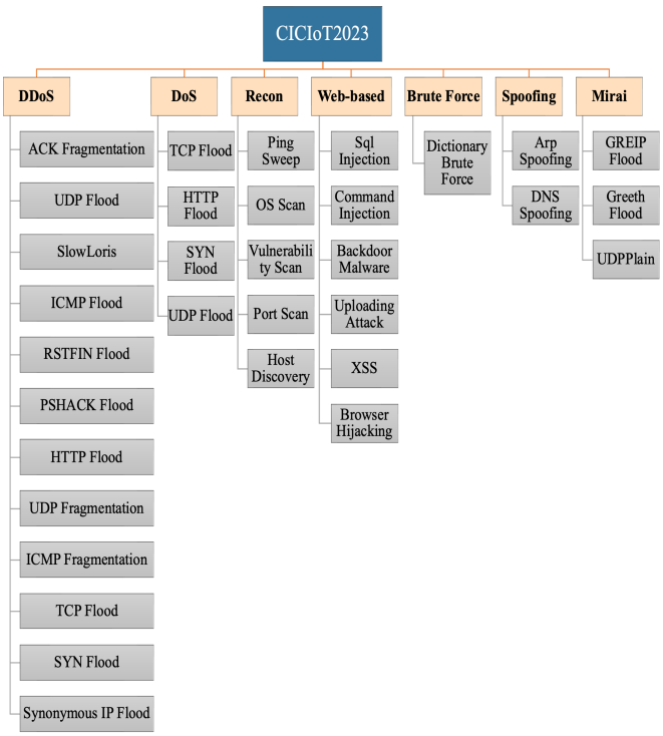

Fig. 3. Class composition of CICIoT2023 dataset

B. Distributed Denial-of-Service (DDoS)

DDoS attacks are carried out by paralyzing the target IoT network system or infrastructure by flooding it with attacks to disrupt the availability of IoT operations. In CICIoT2023, the attacks executed to flood objects consist of 12 types (Fig. 3). The first is ACK fragmentation, where, in most cases, routers, firewalls, and intrusion prevention systems can effectively transmit and manage these ACK fragmentation packets. The second is Slowloris, which targets the application layer by employing fragmentary HTTP requests through open connections to a Web server that is the target. Then, ICMP, HTTP, UDP, and TCP packets are executed to flood the target. The RST-FIN flood also damages network capabilities by continuously sending RST-FIN packets to the target. Apart from that, PSH-ACK flood is also implemented to reduce server operations by flooding using PUSH and ACK requests. The other attacks are UDP fragmentation flooding, which consumes more bandwidth while reducing the number of packets, and ICMP fragmentation flooding, which employs identical fragmented IP packets that comprise a segment of a fragmented ICMP message. The subsequent flooding is using SYN flood packets. The SYN flooding never completes the handshake by transmitting the final ACK (acknowledge) packet to the targeted server; instead, it bombards the server with SYN (synchronize) packets in large numbers. The last one is Synonymous IP Flood, which refers to many manipulated TCP-SYN packets that utilize the server's resources to handle incoming traffic, with the source and destination addresses serving as the target addresses [17].

C. Feature Selection

Feature selection technique is a method for selecting a specific group of predictive features and eliminating irrelevant ones. This stage can improve an intrusion detection system's accuracy performance and reduce computational time [21], as in the paper by Disha *et al*. [22], which used the Gini Impurity-based Weighted Random Forest (GIWRF) algorithm to select features. They stated that the application of feature selection significantly improved model performance. Simon *et al.* [23] also indicated in their research that selecting features is essential to enhance classification accuracy.

This study utilized the Chi-square algorithm for the feature selection stage. The algorithm employed in this study aims to ascertain the degree of independence between a given characteristic and its corresponding class label. We determined the number of selected features for this research to be 30 from all the dataset's features. We labeled the selected features with the symbol k $(k = 30)$.

D. Machine Learning Algorithms

In this study, we used several supervised machine learning algorithms to train the dataset and to test the intrusion detection models. The following is a brief explanation of the algorithms.

- Decision Tree (DT)

  For classification and regression application problems, the DT algorithm is a learner that is one of the best solutions. This model requires just minimal data preprocessing. Depending on the attributes in question, the DT algorithm consistently segments the data in various ways. It informs itself of the decision

principles that may be deduced from the data attributes and then uses those principles to make estimations for the value of the objective variable [24, 25]. This algorithm categorizes data based on a previously learned dataset [26]. Decision trees (DTs) are a straightforward and effective machine learning classifier that anyone without prior statistical training can understand. DTs may handle regression and classification issues and require minimal data preparation [23].

The DT algorithm works by finding the variance of various attributes in the dataset. In a tree structure, it begins with a root node, which is the first node occupied by the attribute with the highest value. Then, it continues with a decision node, which separates the nodes into sub-nodes. The root node is very crucial in producing the correct tree structure. To determine the root node, we must look for the variance of each specific class of attributes. To determine the variance, we have to find the entropy value using the following equation:

$$E(S) = \sum_{i=1}^{n} -P_i \log_2 P_i \qquad (1)$$

Where S is the set of attributes and n is the number of attributes. While Pi is the proportion of Si to S. Variance is defined as irregularity, where the higher the size of the irregularity, the lower the purity value. DT works to reduce irregularities in attributes by utilizing information gain techniques. The information gain equation is shown as follows:

$$Information\ Gain\ (b,a) = E(b) - E(b \mid a) \qquad (2)$$

Where a is the predictor attribute and b is the target attribute. Meanwhile, $E(b)$ is the entropy of the target attribute or object, and $E(b \mid a)$ is the average entropy of the predictor attribute on the target attribute. The higher the value of information gain, the higher the information value of an attribute. The attribute chosen for its high information gain value will become the root node. Next, a tree branch search is done by utilizing information gain and decision nodes to produce the proper tree rules [27].

- K-Nearest Neighbour (KNN)

  K-nearest neighbors, often known as K-NN, is a machine learning method that distinguishes an unfamiliar sample by determining the most identical samples to the new instance [28]. It is a non-parametric classifier, meaning it is not constrained by assumptions regarding the population characteristics from which the sample is derived [29]. This algorithm uses instance-based learning to group samples based on the adjacent training samples [30].

  In classifying unlabeled data, this algorithm uses the mathematical formula Euclidean distance, which is used to determine the distance between unlabeled data points and points from the dataset [31]. Next, KNN assigns each unlabeled data point to the class of the most identically labeled data by finding patterns in the dataset. The training samples are denoted in symbol A, and each test sample is denoted in symbol a, so our goal is to determine the distance between each a and A using the Euclidean distance formula showed in equation (3) as follows:

$$d(a,A) = \sqrt{\sum_{i=1}^{n} (a_i - A_i)^2} \qquad (3)$$

  This study used the k-nearest neighbor algorithm, which calculates the shortest distances for distinguishing unfamiliar samples for classification. The symbol k represents the total number of nearest neighbors included in the majority voting procedure.

- Random Forest (RF)

  The algorithm known as random forests is a proficient form of nonlinear supervised learning that is comprised of multiple decision trees. These decision trees (DTs) are responsible for categorizing new samples based on the values of their respective features [28]. The input data is divided into several categories, and each tree in the DTs selects and classifies a subset of that data arbitrarily. When a testing data item receives a label from a tree, also referred to as a vote, the forest can provide a classification result based on the tree with the most votes [32]. This algorithm is an ensemble learning method that uses several DTs to reduce variance, although at the cost of a minor increase in bias. Superior models generally yield better outcomes due to the substantial reduction in variation. Since all DT predictions are combined to generate the final output, this method is often called the ensemble approach [33, 34].

  For intrusion detection model building using the RF algorithm, $n\_estimator$ in Python programming is used. The $n\_estimator$ is the number of trees to be implemented in the RF algorithm. The more trees created, the more diverse the predictions produced, and the accuracy value can increase. The majority predicted value will be the final result of the approach using the RF algorithm [27]. For this study, we determined the number of trees to be 50 ($n\_estimator = 50$), as any increase would significantly lengthen the required prediction time. The illustration in Fig. 4 presents DTs implemented in the RF algorithm. Label 0 states that the prediction is a negative DDoS attack, while label 1 states that the prediction is a positive DDoS attack. From the prediction results for each tree, it is shown that the majority prediction is 1. So, the final result of sample identification using the RF algorithm is that the sample is positive for a DDoS.

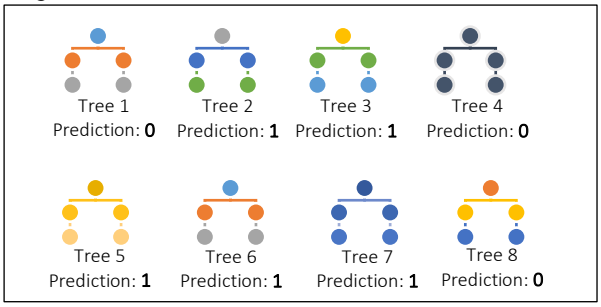

Fig. 4. Implementation of several decision trees of random forest algorithm concept

- Naïve Bayes (NB)

  This algorithm is a straightforward yet effective probabilistic estimator that uses Bayes' postulate under the presumption that all of the examined attributes are independent. i.e., each attribute affects the outcome on its own [21]. The Naïve Bayes algorithm reads and processes the previous training data. This algorithm calculates the probabilities, previous class probabilities, and previous probabilities of the predictors and stores them in a dictionary. The algorithm pre-processes the test data. Using the probability dictionary, the previous probabilities of the classes, and the predictors of the training module, the test records will be evaluated and marked with a class with the maximum likelihood [35]. Equation (4) represents Bayes' theorem, which classifies a sample of objects into the most probable class utilizing highest probability estimates. In this study, we concentrated on detecting DDoS attacks; therefore, we designated the label 1 to the DDoS attack and the label 0 to the remaining attacks. The symbol $q = \{q_1, q_2\} = \{DDoS, rest\}$ denotes DDoS attacks and other types of attacks and vectors of data-dependent feature symbols defined in $r = \{r_1, r_2, \dots, r_n\}$.

$$P(q \mid r_1, r_2 \dots, r_n) = \frac{P(r_1, r_2 \dots, r_n \mid q)\, P(q)}{P(r_1, r_2 \dots, r_n)} \qquad (4)$$

  The probability that q will exist given the occurrence of $(r_1, r_2, \dots, r_n)$ is denoted by $P(q \mid r_1, r_2 \dots, r_n)$. $P(q)$ relates to a prior probability. $P(r_1, r_2 \dots, r_n \mid q)$ is the likelihood probability of $(r_1, r_2, \dots, r_n)$ occuring, given q has occured. $P(r_1, r_2 \dots, r_n)$ is a marginal probability [21].

E. Evaluation Metrics

The performance of the intrusion detection model is evaluated using evaluation metrics as a benchmark. This study evaluated the performance of false positive rate, accuracy, precision, recall, and F1-score. The following parameters are required to calculate these metric values: TP, TN, FP, and FN. TP, or true positive, is the condition under which the model can accurately detect attacks, whereas TN, or true negative, is the condition under which the model can correctly identify normal samples. FP, or false positive, is a condition in which the model cannot accurately detect normal samples as benign, whereas FN, or false negative, is when the model cannot correctly identify an attack as an attack. The evaluation metrics formulas are shown in Table I below.

TABLE I. DESCRIPTION OF EVALUATION METRICS [21]

Metric	Formula
False positive rate: number of valid traffic incorrectly classified as an attack.	$\frac{FP}{FP + TN}$
Accuracy: the degree of how close the measurement results made by the model are to the actual value	$\frac{(TP + TN)}{(TP + TN + FP + FN)}$
Precision: the capability of a classification system to properly recognize attacks out of the total number of positive predictions.	$\frac{TP}{TP + FP}$
Recall: the classification capability to accurately anticipate attacks from real attacks	$\frac{TP}{TP + FN}$
F1-score: the harmonic mean of precision and recall	$\frac{2TP}{2TP + FP + FN}$

IV. RESULTS AND DISCUSSION

A. Results

In this section, we explain the results of the research that has been carried out. This research utilized Python programming to build machine learning-based intrusion detection models. Sci-kit-learn was deployed using Python 3.9 and a 64-bit ARM CPU architecture. In the initial stage, before the training dataset, the feature selection stage was carried out first using the Chi-square algorithm. The number of features we determined was 30 ($k = 30$). Then, the dataset was trained using four machine learning algorithms, described in the previous section, to produce intrusion detection models. We utilized 23,412,707 instances randomly from the CICIoT2023 dataset for training. In this study, the model's training and testing duration were measured. After the training process and the models were generated, we continued with the testing or model validation stage to determine whether the system performance had met the goals to be achieved. The system was tested using 0.01 of the total data samples ($test\_size = 0.01$), or around 234,127 samples. Table II lists the time the models require for the training and testing dataset.

TABLE II. TIME USAGE BY THE MODELS FOR TRAINING AND TESTING CICIOT2023 DATASET

Time	Model			
	Decision Tree	**K-Nearest Neighbor**	**Random Forest**	**Naïve Bayes**
Training (s)	240.06	5993.93	3799.48	456
Testing (s)	0.08	53.29	2.15	0.57

The number of samples of TP, TN, FP, and FN from the test results is shown in Table III. Table IV displays the performance test results of accuracy (Acc), precision (Prec), recall, F1-score, and false positive rate (FPR) of intrusion detection models when detecting DDoS attacks. We present a graph of the testing results in Fig. 5 to compare the results of the performance evaluation of intrusion detection models.

TABLE III. NUMBER OF SAMPLES FROM DDOS ATTACK DETECTION TESTS

Model	True positive (TP)	True negative (TN)	False positive (FP)	False negative (FN)
Decision Tree	5929	228198	0	0
K-Nearest Neighbor	5818	228142	111	2
Random Forest	5929	228198	0	0
Naïve Bayes	2521	45297	3408	67

TABLE IV. PERFORMANCE EVALUATION RESULTS

Model	Acc (%)	Prec (%)	Recall (%)	F1-Score (%)	FPR (%)
Decision Tree	**100**	**100**	**100**	**100**	**0**
K-Nearest Neighbor	99.95	98.13	99.97	99.04	0.05
Random Forest	**100**	**100**	**100**	**100**	**0**
Naïve Bayes	93.23	42.52	97.41	59.2	6.99

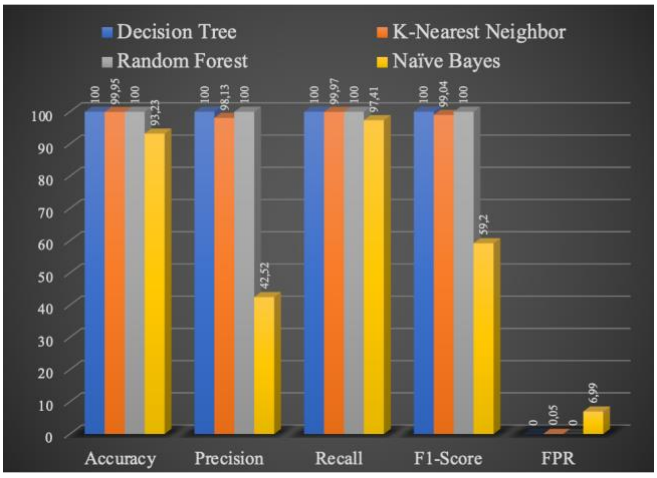

Fig. 5. Results of model performance evaluation in detecting DDoS attacks

B. *Discussion*

Experiments have shown that using the CICIoT2023 dataset to create an intrusion detection model leads to efficient models that find DDoS attacks on IoT networks and have the best performance percentage values.

The decision tree (DT) model showed the best performance from the time measurement results because it required the shortest time to train 23,412,707 data samples. The DT model only needed a running time of 240.06 seconds to qualify more than 20 million instances. Meanwhile, the KNN, RF, and NB models each required 5993.93, 3799.48, and 456 seconds. DT performance was also superior for its ability to test data samples in the shortest time compared to other models. The DT model was able to classify 234,127 instances in 0.08 seconds, or 2,926,588 instances in 1 second. Meanwhile, the KNN model took 53.29 seconds, RF took 2.15 seconds, and NB took 0.57 seconds to identify 234,127 instances.

The performance of accuracy, precision, recall, F1-score, and false positive rate was obtained by measuring the TP, TN, FP, and FN parameters. From the measurement results, it is known that the RF and DT models obtained the best performance. As shown in Table III, the RF and DT models produced the same amount of TP and TN. Interestingly, the RF and DT models had no FP or FN instances detected, which resulted in their FPR values being 0%. Based on the test results shown in Table IV, the RF and DT models' accuracy reached 100%. They produced precision, recall, F1 score, and FPR values with all being 100%. The performance values achieved by the KNN model were also good, with an accuracy score of 99.95%, precision of 98.13%, recall of 99.97%, F1 score of 99.04%, and FPR of 0.05%. The NB model produces accuracy with a percentage of 93.23% recall of 97.41%, but for precision and F1 score values, the NB model did not reach a rate of 90%, the results of which were 42.52% and 59.2%. The findings show that the NB model did not perform effectively when trained or tested with CICIoT2023 dataset. These small percentages were influenced by the true positive (TP) and false positive (FP) values resulting from detection by the NB model, where from the test results, it is known that the FP value obtained is more significant than TP (Table III). Because the smaller the TP value of FP, the smaller the recall value and F1 score. The comparison of performance evaluation of the four models can be seen in Fig. 5.

To objectively evaluate our proposed method on broader effects, we compared our results to previous research papers, as shown in Table V. Application of the CICIoT2023 dataset, which is the latest dataset with more than 30 million instances, in combination with machine learning methods and the Chi-square algorithm for feature selection produced quite excellent performance of intrusion detection models for identifying DDoS attacks.

TABLE V. PERFORMANCE COMPARISON WITH PREVIOUS RESEARCH

Proposed method by	Feature Selection	Classifier Algorithm	Dataset	The best accuracy
Hattarki *et al.* [11]	-	RF	Real traffic	91.18%
Karanfiloska *et al.* [18]	K-means, PCA	XGBoost, RF, Logistic Regression, DT	NF-ToN-IoT-V2	98.8%
Illavarason and Sundaram [19]	Correlation-based, Information Gain Ratio-based, PCA	KNN, Neural Network, DT, NB, SVM	NSL-KDD	99.1%
Khater *et al.* [20]	Mutual Information and PCA	Multi Layer Perceptron	ADFA-LD	95%
This paper	Chi-square	DT, KNN, RF, NB	CICIoT2023	**100%**

V. CONCLUSION

This study focused on developing models of intrusion detection systems for identifying DDoS attacks on IoT networks using machine learning algorithms, such as decision trees, K-nearest neighbors, random forests, and Naïve Bayes. It utilized the latest IoT dataset, the CICIoT2023 dataset, for training and testing stages. The chi-square algorithm was applied in this study as a feature selector that determines valuable features and selects irrelevant features from the dataset. The findings show that intrusion detection models produced high performances for detecting DDoS attacks. The random forest (RF) and decision tree (DT) models' performances outperformed other detection models' performances. They delivered the best accuracy performance with a percentage of 100%. The RF and DT models also exceeded the precision, recall, and F1-score values, with all values being 100%. Regarding time spent in the training and testing process, the decision tree model produced superior performance compared to other models, with a speed of classifying ≈ 2,926,588 instances in 1 second.

The application of IDS in detecting attacks within the network is not exempt from preventive measures (IPS) against such attacks. However, because in this study we limited the scope of the problem to only IDS, for future research, an IDS integrated with an IPS can be executed for further investigation. In addition, a combination of applying clustering and classifier algorithms can also be carried out in the future, which is expected to reduce training and testing time.

REFERENCES

[1] P. M. Chanal and M. S. Kakkasageri, "Security and privacy in IoT: a survey," *Wireless Personal Communications,* vol. 115, no. 2, pp. 1667-1693, 2020, doi: 10.1007/s11277-020-07649-9.

[2] F. A. Alaba, M. Othman, I. A. T. Hashem, and F. Alotaibi, "Internet of Things security: A survey," *Journal of Network and Computer Applications,* vol. 88, pp. 10-28, 2017, doi: 10.1016/j.jnca.2017.04.002.

[3] K. Somasundaram and K. Selvam, "IOT – Attacks and Challenges," *International Journal of Engineering and Technical Research,* vol. 8, no. 9, 2018/9// 2018, doi: 10.31873/IJETR.8.9.67.

[4] M. F. Elrawy, A. I. Awad, and H. F. Hamed, "Intrusion detection systems for IoT-based smart environments: a survey," *Journal of Cloud Computing,* vol. 7, no. 1, pp. 1-20, 2018, doi: 10.1186/s13677-018-0123-6.

[5] S. Hajiheidari, K. Wakil, M. Badri, and N. J. Navimipour, "Intrusion detection systems in the Internet of things: A comprehensive investigation," *Computer Networks,* vol. 160, pp. 165-191, 2019, doi: 10.1016/j.comnet.2019.05.014.

[6] A. Yastrebova, R. Kirichek, Y. Koucheryavy, A. Borodin, and A. Koucheryavy, "Future Networks 2030: Architecture & Requirements," *2018 10th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT),* pp. 1-8, 2018, doi: 10.1109/ICUMT.2018.8631208.

[7] H. F. Atlam and G. B. Wills, "IoT security, privacy, safety and ethics," *Digital twin technologies and smart cities,* pp. 123-149, 2020, doi: 10.1007/978-3-030-18732-3_8.

[8] N. Islam *et al.*, "Towards Machine Learning Based Intrusion Detection in IoT Networks," *Computers, Materials \& Continua,* vol. 69, no. 2, pp. 1801--1821, 2021, doi: 10.32604/cmc.2021.018466.

[9] R. Chaganti, W. Suliman, V. Ravi, and A. Dua, "Deep Learning Approach for SDN-Enabled Intrusion Detection System in IoT Networks," *Information,* vol. 14, no. 1, p. 41, 2023, doi: 10.3390/info14010041.

[10] J. Kitili and D. Abiero. "Kenya's Digital Infrastructure Under Threat? A Look at Anonymous Sudan's Thwarted Cyberattack Attempt and its Implications for Kenya's Digital Systems." https://cipit.strathmore.edu/kenyas-digital-infrastructure-under-threat-a-look-at-anonymous-sudans-thwarted-cyberattack-attempt-and-its-implications-for-kenyas-digital-systems/ (accessed August 22, 2023).

[11] R. Hattarki, S. Houji, and M. Dhage, "Real Time Intrusion Detection System For IoT Networks," in *2021 6th International Conference for Convergence in Technology (I2CT)*, 2-4 April 2021 2021, pp. 1-5, doi: 10.1109/I2CT51068.2021.9417815.

[12] S. Roy, J. Li, B.-J. Choi, and Y. Bai, "A lightweight supervised intrusion detection mechanism for IoT networks," *Future Generation Computer Systems,* vol. 127, pp. 276-285, 2022, doi: 10.1016/j.future.2021.09.027.

[13] H. Wu, H. Han, X. Wang, and S. Sun, "Research on artificial intelligence enhancing internet of things security: A survey," *Ieee Access,* vol. 8, pp. 153826-153848, 2020, doi: 10.1109/ACCESS.2020.3018170.

[14] R. Elsayed, R. Hamada, M. Hammoudeh, M. Abdalla, and S. A. Elsaid, "A Hierarchical Deep Learning-Based Intrusion Detection Architecture for Clustered Internet of Things," *Journal of Sensor and Actuator Networks,* vol. 12, no. 1, p. 3, 2022, doi: 10.3390/jsan12010003.

[15] R. Zhao *et al.*, "A novel intrusion detection method based on lightweight neural network for internet of things," *IEEE Internet of Things Journal,* vol. 9, no. 12, pp. 9960-9972, 2021, doi: 10.1109/JIOT.2021.3119055.

[16] A. H. H. Kabla *et al.*, "Machine and deep learning techniques for detecting internet protocol version six attacks: a review," *International Journal of Electrical and Computer Engineering (IJECE),* vol. 13, no. 5, pp. 5617-5631, 2023, doi: 10.11591/ijece.v13i5.pp5617-5631.

[17] E. C. P. Neto, S. Dadkhah, R. Ferreira, A. Zohourian, R. Lu, and A. A. Ghorbani, "CICIoT2023: A Real-Time Dataset and Benchmark for Large-Scale Attacks in IoT Environment," *Sensors,* vol. 23, no. 13, p. 5941, 2023. [Online]. Available: https://www.mdpi.com/1424-8220/23/13/5941.

[18] M. Karanfilovska, T. Kochovska, Z. Todorov, A. Cholakoska, G. Jakimovski, and D. Efnusheva, "Analysis and modelling of a ML-based NIDS for IoT networks," *Procedia Computer Science,* vol. 204, pp. 187-195, 2022/01/01/ 2022, doi: https://doi.org/10.1016/j.procs.2022.08.023.

[19] P. Illavarason and B. Kamachi Sundaram, "A Study of Intrusion Detection System using Machine Learning Classification Algorithm based on different feature selection approach," *2019 Third International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC),* pp. 295-299, 2019.

[20] B. Sudqi Khater, A. W. B. Abdul Wahab, M. Y. I. B. Idris, M. Abdulla Hussain, and A. Ahmed Ibrahim, "A lightweight perceptron-based intrusion detection system for fog computing," *applied sciences,* vol. 9, no. 1, p. 178, 2019.

[21] T. Wisanwanichthan and M. Thammawichai, "A double-layered hybrid approach for network intrusion detection system using combined naive bayes and SVM," *IEEE Access,* vol. 9, pp. 138432-138450, 2021, doi: 10.1109/ACCESS.2021.3118573.

[22] R. A. Disha and S. Waheed, "Performance analysis of machine learning models for intrusion detection system using Gini Impurity-based Weighted Random Forest (GIWRF) feature selection technique," *Cybersecurity,* vol. 5, no. 1, pp. 1-22, 2022, doi: 10.1186/s42400-021-00103-8.

[23] J. Simon, N. Kapileswar, P. K. Polasi, and M. A. Elaveini, "Hybrid intrusion detection system for wireless IoT networks using deep learning algorithm," *Computers and Electrical Engineering,* vol. 102, p. 108190, 2022/09/01/ 2022, doi: 10.1016/j.compeleceng.2022.108190.

[24] J. R. Quinlan, "Induction of decision trees," *Machine Learning,* vol. 1, no. 1, pp. 81-106, 1986/03/01 1986, doi: 10.1007/BF00116251.

[25] S. R. Safavian and D. Landgrebe, "A survey of decision tree classifier methodology," *IEEE Transactions on Systems, Man, and Cybernetics,* vol. 21, no. 3, pp. 660-674, 1991, doi: 10.1109/21.97458.

[26] M. Nene and J. Singh, "A survey on machine learning techniques for intrusion detection systems," *International Journal of Advanced Research in Computer and Communication Engineering,* vol. 11, p. 4349, 12/10 2013.

[27] J. Han and M. Kamber, *Data Mining: Concepts and Techniques Third Edition*. Elsevier, 2012.

[28] S. B. Kotsiantis, "Supervised Machine Learning: A Review of Classification Techniques," presented at the Proceedings of the 2007 conference on Emerging Artificial Intelligence Applications in Computer Engineering: Real Word AI Systems with Applications in eHealth, HCI, Information Retrieval and Pervasive Technologies, 2007.

[29] G. Serpen and E. Aghaei, "Host-based misuse intrusion detection using PCA feature extraction and kNN classification algorithms," *Intelligent Data Analysis,* vol. 22, pp. 1101-1114, 2018, doi: 10.3233/IDA-173493.

[30] Y. K. Saheed, A. I. Abiodun, S. Misra, M. K. Holone, and R. Colomo-Palacios, "A machine learning-based intrusion detection for detecting internet of things network attacks," *Alexandria Engineering Journal,* vol. 61, no. 12, pp. 9395-9409, 2022, doi: 10.1016/j.aej.2022.02.063.

[31] S. Zhang, "Challenges in KNN Classification," *IEEE Transactions on Knowledge and Data Engineering,* vol. 34, no. 10, pp. 4663-4675, 2022, doi: 10.1109/TKDE.2021.3049250.

[32] H. M. Alshahrani, "CoLL-IoT: A Collaborative Intruder Detection System for Internet of Things Devices," *Electronics,* vol. 10, no. 7, p. 848, 2021, doi: 10.3390/electronics10070848.

[33] S. Learn. Scikit Learn. https://scikit-learn.org/stable/modules/ensemble.html#forests-of-randomized-trees (accessed 25 December, 2022).

[34] T. Saranya, S. Sridevi, C. Deisy, T. D. Chung, and M. A. Khan, "Performance analysis of machine learning algorithms in intrusion detection system: a review," *Procedia Computer Science,* vol. 171, pp. 1251-1260, 2020, doi: 10.1016/j.procs.2020.04.133.

[35] R. D. Ta and S. Badugub, "Network Intrusion Detection System Using KNN and Naive Bayes Classifiers," *system,* vol. 4, no. 5, p. 6.