A Paradigm for DoS Attack Disclosure using Machine Learning Techniques

Mosleh M. Abualhaj[1], Ahmad Adel Abu-Shareha[2], Mohammad O. Hiari[3]
Yousef Alrabanah[4], Mahran Al-Zyoud[5], Mohammad A. Alsharaiah[6]
Department of Networks and Information Security[1, 3, 5]
Department of Computer Science[2]
Department of Software Engineering[4]
Data Science and Artificial Intelligence[6]
Al Ahliyya Amman University, Amman, Jordan

***Abstract*—Cybersecurity is one of the main concerns of governments, businesses, and even individuals. This is because a vast number of attacks are their core assets. One of the most dangerous attacks is the Denial of Service (DoS) attack, whose primary goal is to make resources unavailable to legitimate users. In general, the Intrusion Detection and Prevention Systems (IDPS) hinder the DoS attack, using advanced techniques. Using machine learning techniques, this study will develop a detection model to detect DoS attacks. Utilizing the NSL-KDD dataset, the suggested DoS attack detection model was investigated using Naive Bayes, K-nearest neighbor, Decision Tree, and Support Vector Machine algorithms. The Accuracy, Recall, Precision, and Matthews Correlation Coefficients (MCC) metrics are used to compare these four techniques. In general, all techniques are performing well with the proposed model. However, The Decision Tree technique has outperformed all the other techniques in all four metrics, while the Naive Bayes technique showed the lowest performance.**

Keywords*—DoS attack; machine learning; NSL-KDD; IDPS systems

I. Introduction

The world is currently living in the digital era. This digital era has produced many services and applications to make life easier. One of the primary concerns of these services and applications is security [1]. Companies and even individuals live a nightmare due to the number of cyberattacks. At the same time, more than 61000 websites attack is blocked every day. In addition, around 24000 malicious mobile applications are blocked every day on the stores of the applications [2]. One of the most dangerous cyberattacks is a Denial of Service (DoS) attack. The main goal of the DoS attack is to make a resource unavailable to the intended users. DoS attack is increasing rapidly; it is expected that the number of worldwide DoS attack will reach 15.4 million by 2023 [3].

Intrusion Detection and Prevention Systems (IDPS) are among the techniques available to counteract a DoS attack. IDPS is software/hardware that observes and inspects system events in order to sense and warn of unauthorized efforts to access system resources in real-time or near real-time. IDPS detects intrusion by either searching for a pre-defined pattern in the traffic or by observing anomalies of what is considered normal traffic for the network or host [4]. IDPS should be equipped with smart and self-learning techniques to detect zero-day DoS attacks. Machine learning is a subfield of artificial intelligence that encompasses a number of techniques for accomplishing this goal [5].

As the name implies, machine learning systems improve automaticity through experience and by using existing data, which makes it suitable to detect zero-day DoS attacks. Supervised, unsupervised, and semi-supervised machine learning are all types of machine learning. Generally, supervised learning algorithms operate on structured and labeled data similar to that used by the IDPS [6] [7]. Hence, the fundamental aim of this research is to suggest a paradigm for identifying suitable supervised machine learning algorithms for detecting DoS attacks via IDPS.

This paper is structured as follows. Section 2 covers the topics fundamental to this work. These topics include NSL-KDD dataset machine learning techniques, min-max scaler, and K-Fold Cross-Validation. Section 3 discusses related works that have employed machine learning approaches to detect DoS attacks. Section 4 discusses the proposed DoS attack detection model. Finally, Section 5 concludes the paper and discusses the scope for future work.

II. Background

This section discusses the basic concepts that are related to this work. This includes a brief description of the NSL-KDD dataset used in this article. The Machine learning techniques used in this article will also be briefed. Finally, the algorithms used in the data pre-processing and to validate the result will be discussed.

A. NSL-KDD Dataset

NSL-KDD dataset is a processed version of the KDD-CUP99, in which the records that adversely impact the systems are removed. NSL-KDD dataset still has some problems; however, it is still considered an adequate benchmark dataset that helps security developers investigate intrusion detection techniques. The number of records in the NSL-KDD dataset is good to run the experiments and evaluate the results of different techniques. Table I shows the number of records in the NSL-KDD dataset according to the attack type. The NSL-KDD dataset has four different attack types. This paper is only interested in the DoS attack, and all records of the other attacks

are deleted during the pre-processing stage, as discussed below. Table II shows the main attributes of the NSL-KDD dataset [7][8][9].

TABLE I. NUMBER OF RECORD FOR EACH ATTACK

Attack Type	*Number of records*
DoS	53387
Probe	14077
U2R	119
R2L	3880
Normal	77055

TABLE II. THE FEATURES OF NSL-KDD DOS

No	*Feature Name*	*Data Type*	*Feature Description*	*Lowest Value*	*Highest Value*
1	duration	Numeral	The session's length	Zero	54451
2	protocol_type	Text	Session protocol	N/A	N/A
2	protocol_type	Text	Session protocol	N/A	N/A
3	service	Text	Destination service	N/A	N/A
4	flag	Text	The session's status flag.	N/A	N/A
5	src_bytes	Numeral	Bytes transmitted from sender to receiver	Zero	89581520
6	dst_bytes	Numeral	Bytes transmitted from receiver to sender	Zero	7028652
7	land	Numeral	1 If from/to the same host/port; else 0.	Zero	One
8	wrong_fragment	Numeral	The number of incorrect fragments.	Zero	Three
9	urgent	Numeral	Number of urgent packets	Zero	Three
10	hot	Numeral	Number of hot indicators	Zero	101
11	num_failed_logins	Numeral	Number of unsuccessful login in attempts	Zero	Four
12	logged_in	Numeral	1 If successfully logged in; else 0.	Zero	One
13	num_compromised	Numeral	The number of compromised conditions	Zero	7479
14	root_shell	Numeral	1 If a root shell is attained; else 0.	Zero	One
15	su_attempted	Numeral	1 If (su root) command tried; else 0.	Zero	Two
16	num_root	Numeral	Number of root accesses	Zero	7468
17	num_file_creations	Numeral	The total number of creation operations.	Zero	100
18	num_shells	Numeral	The total number of shell prompts.	Zero	Two
19	num_access_files	Numeral	The total number of operations on access control files.	Zero	Nine
20	num_outbound_cmds	Numeral	The total number of ftp session outbound commands.	Zero	One
21	is_host_login	Numeral	1 If the login belongs to the hot list; else 0.	Zero	One
22	is_guest_login	Numeral	1 If it's a guest login; else 0.	Zero	One
23	Count	Numeral	The number of sessions to the same host as the present session, in the last 2 seconds.	Zero	511
24	srv_count	Numeral	The number of connections to the same service as the current connection, In the last two seconds.	Zero	511
25	serror_rate	Numeral	The ratio of connections in the same host connection that contain "SYN" errors	Zero	One
26	srv_serror_rate	Numeral	The ratio of connections in the same-service connection that have "SYN" errors	Zero	One
27	rerror_rate	Numeral	The percentage of connections in the same-host connection that have "REJ" errors	Zero	One
28	srv_rerror_rate	Numeral	The ratio of contain s in the same-service contain that contain "REJ" errors	Zero	One
29	same_srv_rate	Numeral	The percentage of connections to the same-service connection.	Zero	One
30	diff_srv_rate	Numeral	The percentage of connections to	Zero	One

No	*Feature Name*	*Data Type*	*Feature Description*	*Lowest Value*	*Highest Value*
			different services.		
31	srv_diff_host_rate	Numeral	The percentage of connections to various hosts in the same-service connection.	Zero	One
32	dst_host_count	Numeral	The percentage count of connections that contain the same receiver host.	Zero	255
33	dst_host_srv_count	Numeral	The percentage count of connections that contain the same receiver host and using the identical service	Zero	255
34	dst_host_same_srv_rate	Numeral	The percentage of connections that contain the same receiver host and using the identical service.	Zero	One
35	dst_host_diff_srv_rate	Numeral	The percentage of various services on the present host.	Zero	One
36	dst_host_same_src_port_rate	Numeral	The percentage of connections to the present host that contain the same port.	Zero	One
37	dst_host_srv_diff_host_rate	Numeral	The percentage of connections to the identical service coming from various hosts.	Zero	One
38	dst_host_serror_rate	Numeral	The percentage of connections to the present host that contain an "SO" error	Zero	One
39	dst_host_srv_serror_rate	Numeral	The percentage of connections to the present host and determined service that contain an "SO" error	Zero	One
40	dst_host_rerror_rate	Numeral	The percentage of connections to the present host that contain an "RST" error	Zero	One
41	dst_host_srv_rerror_rate	Numeral	The percentage of connections to the present host and determined service that contain an "RST" error	Zero	One

B. Machine Learning Techniques that are used in this Article

Supervised machine learning deals with data sets that contain both inputs and the corresponding desired outputs. The classification algorithms category is used within supervised learning when the outputs are discrete; restricted to a limited set of values. The most common classification algorithms are Naive Bayes, K-Nearest Neighbors (KNN), Decision Tree, and Support Vector Machines (SVM) [7][10][11][12].

1) Naive bayes: Naive Bayes is a simple technique based on the Bayes theorem and used to handle classification problems. The Naive Bayes assumption is that the features are independent of one another; existing of any feature is unrelated to any other feature. It is known as one of the best classification algorithms and creates fast machine learning models that predict quickly. In Naive Bayes, the features are making independent and equal contributions to the outcome. Equation 1 shows the probabilistic expressions used in Bayes' theorem [7][10].

$$P(X|Y) = \frac{P(Y|X)P(X)}{P(Y)} \quad (1)$$

2) K-NN: One of the most important and extensively used machine learning algorithms is K-NN. As the name implies, K-NN finds the closest K (number of neighbors) nearest neighbor points to the target point. Then, it predicts the output of the target point from these neighbor points. K can be constant or vary based on the local density of points. Typically, k equals the square root of the dataset's record count. Euclidean is one of the algorithms that are used to find the neighbor points by KNN. Equation 2 shows the formula of the Euclidean algorithm [7][11].

Euclidean Distance between X and Y =

$$\sqrt{(\boldsymbol{A_2} - \boldsymbol{A_1})^2 + (\boldsymbol{B_2} - \boldsymbol{B_1})^2} \quad (2)$$

3) Decision Tree: The decision tree technique creates an upside-down tree to represent the classification model. It is easy to understand, visualize, and requires little data preparation. The tree consists of nodes that symbolize a dataset's features, branches symbolize the decision rules, and leaves symbolize the class, as shown in Fig. 1. The decision tree is based on the if-else statements (True/False) to move to the next node till reaching the leaf [7][12].

4) SVM: SVM is a widely used supervised learning approach for classification. The SVM technique plots the data items as a space split into categories. Then, it finds the hyperplane that distinctly separates the points in space. The SVM technique should choose the hyperplane with the maximum distance between the target data points. This gives a more accurate classification for any new data points. Fig. 2 clarifies the SVM technique [7][10].

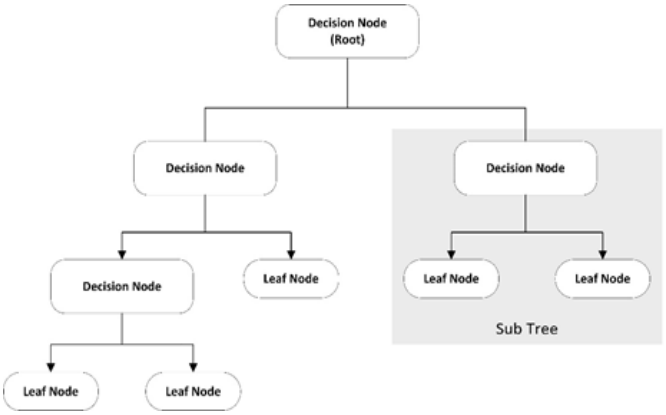

Fig. 1. Decision Tree Technique Scheme.

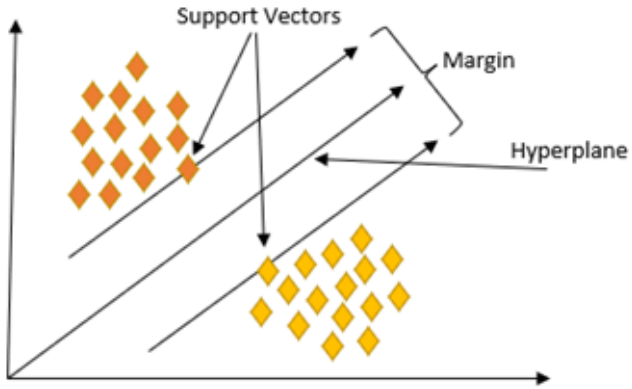

Fig. 2. SVM Technique Scheme.

C. Min-max Scaler

Most machine learning techniques perform better when the data are distributed similarly. In many cases, the data within the dataset is distributed on a wide-scale and, thus, the data should be scaled. Min-max scaler is one of the most used techniques to scale the data within the acceptable range for the machine learning techniques. By default, the Min-max scaler technique returns a value between 0 and 1, using Equation 3.

$$Z_{new} = (Z - Z_{min}) / (Z_{max} - Z_{min}) \quad (3)$$

Where Z_{new} is new derived value, Z is the original value, Z_{min} is the minimum value of the feature, Z_{max} is the maximum value of the feature [13].

D. K-Fold Cross-Validation

When it comes to machine learning, the approach known as K-Fold Cross-Validation is used to validate the results of a model. It is widely used because it is simple, easy to understand, and, more importantly, reduces the validated model's bias. Using the K-Fold Cross-Validation method, the data is split into various groups (k groups). The proposed machine learning is trained on k-1 groups, and the remaining group is used to validate the model [14].

III. Related Work

This section discusses related work on detecting DoS attacks using machine learning approaches.

Peneti S. and Hemalatha E. have proposed a machine learning model to detect Distributed DoS (DDoS) attacks. The authors investigated four different machine learning techniques to design their model: XGBoost, AdaBoost, Random Forests, and Multilayer Perceptron. The CIC IDS 2017 dataset, which cantinas 83 additional features, has been used to evaluate the proposed model. The Recursive Feature Elimination method has been used to shrink the dataset to only the most relevant features to enhance the proposed model performance. The number of features has been set to six and after some experiments the number of features has been finalized to eight. The accuracy, precision, recall, and F1 score measures have been used to evaluate the suitable machine learning techniques for the proposed model. Among the investigated four techniques, Random Forests has outperformed the other techniques in detecting the DDoS attack, while the Multilayer Perceptron has performed less in this particular problem [5].

One of the recent articles that have been used the machine learning techniques for DoS attack detection was proposed by Wankhede S. & Kshirsagar D. Wankhede S. & Kshirsagar D have been used common machine learning techniques to detect DoS attack; namely Random Forest (RF) and Multi-Layer Perceptron (MLP) techniques. The suggested model is aimed at detecting DoS attacks at the application layer. The DoS attack that occurs at the other OSI layers has not been considered. The same CIC IDS 2017 dataset was used to evaluate the RF and MLP techniques for detecting DoS attacks at the application layer. The CIC IDS 2017 dataset is divided into distinct groups, and an appropriate group for each technique is identified. Weka tool has been used to evaluate the RF technique versus MLP technique in the proposed model. The results demonstrated that the RF outperforms the MLP in terms of accuracy [15].

Another article that used machine learning techniques for DoS attack detection was proposed by Zhe W., Wei C., and Chunlin L. However, the proposed model in this work is designed specifically for smart grid technology. The authors have investigated three different machine learning techniques to protect the smart grid: SVM, Decision Tree, and Naive Bayesian. After examining these three techniques on the KDD99 dataset, it is found that the SVM technique is the best for protecting smart grid technology from DoS attacks. The data is first collected from the network, then certain features are selected from the dataset, and the primary component analysis is used for dimensionality reduction. The accuracy, precision and recall, and F1 score measures have been used to evaluate the suitable machine learning techniques for the proposed model. Among the three techniques tested, SVM outperformed the others in detecting DoS attacks on smart grid technology. [16].

He Z., Zhang T., and Lee, R. B. have advocated the use of machine learning techniques to detect DoS attacks originating in the cloud. The proposed system has investigated four different DoS attack techniques: SSH brute-force, ICMP flooding, DNS reflection, and TCP SYN attacks. This method utilizes statistical data from the hypervisor of the cloud server and the virtual machines to prohibit network packages from being sent out to the external network. The authors have implemented a prototype of the proposed detection system

under natural cloud settings. The cloud is comprised of six servers (labeled S0 to S5), each of which hosts many virtual machines. Several machine learning techniques have been used in the proposed system, including SVM Linear Kernel, SVM RBF Kernel, SVM Poly Kernel, Decision Tree, Naive Bayes, and Random Forest. Among the investigated techniques, SVM Linear Kernel has outperformed other techniques in detecting the DoS attack sourced from the cloud [17].

IV. Proposed DoS Attack Detection Model

This section outlines the suggested model for detecting DoS attacks. First, the NSL-KDD dataset will be processed to be prepared for training and testing the proposed model. Then, the proposed DoS attack detection model will be introduced in detail.

A. Data Preprocessing

Data preprocessing is a set of operations applied to the data to prepare the dataset for machine learning. As discussed below, data transformation and normalization are two of these processes that have been applied to the NSL-KDD dataset in this paper [8][18].

1) Data transformation: NSL-KDD dataset contains numerical and nominal data, as shown in Table II. One of the first steps in data preprocessing is transformation, converting all data to numerical for the machine learning techniques to be applicable. Three nominal features in the NSL-KDD dataset have been transformed to numeric values: protocol type, service, and flag. These features have been converted using the label encoding method [19]. Label encoding changes the values to a number between zero and the number of classes minus one, as shown in Table III. Tables IV and V show samples of the NSL-KDD dataset before and after the transformation operation. Besides, the output column in the NSL-KDD dataset contains four different types of attacks, each of which has several sub-types. All the attack sub-types have been removed except for the DoS sub-types, which is our target in this paper. Then, all DoS sub-types have been replaced to be DoS attack, so that the output column contains only two outputs: DoS attack and normal data. Again, these two outputs have been converted to from nominal into numeric data using the label encoding method. Now, the output column contains 0 representing the DoS attack and 1 representing normal data.

2) Data normalization: An essential step in data preprocessing is normalization operation. Normalization techniques convert the large-scale values into a compatible scale. This enhances the performance of the machine learning techniques and leads to more accurate results. NSL-KDD dataset contains several features distributed at a large scale and needs to be normalized. This study has applied the Min-max scaler technique (as discussed above), which scales the values of a feature between 0 and 1 [7][13]. Table VI shows a sample of the NSL-KDD dataset after normalization. Fig. 3 illustrates the NSL-KDD dataset data preprocessing steps.

TABLE III. Transformation

Feature Name	***Old Value***	***New Value***
Protocol Type	Icmp	One
	Tcp	Two
	Udp	Three
Service	auth,bgp ……. , X11, Z39_50	0-64
Flag	OTH	Zero
	REJ	One
	RSTO	Two
	RSTOS0	Three
	RSTR	Four
	S0	Five
	S1	Six
	S2	Seven
	S3	Eight
	SF	Nine
	SH	Ten

TABLE IV. Before Transformation

No	***Instances***	***Output***
1	0,tcp,ftp_data,SF,491,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,2,0,0,0,0,1,0,0,150,25,0.17,0.03,0.17,0,0,0,0.05,0	normal
2	0,udp,other,SF,146,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,13,1,0,0,0,0,0.08,0.15,0,255,1,0,0.6,0.88,0,0,0,0,0	normal
3	0,tcp,private,S0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,123,6,1,1,0,0,0.05,0.07,0,255,26,0.1,0.05,0,0,1,1,0,0	DoS
4	0,tcp,private,REJ,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,121,19,0,0,1,1,0.16,0.06,0,255,19,0.07,0.07,0,0,0,0,1,1	DoS
5	0,tcp,private,S0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,166,9,1,1,0,0,0.05,0.06,0,255,9,0.04,0.05,0,0,1,1,0,0	DoS

TABLE V. After Transformation

No	***Instances***	***Output***
1	0,1,19,9,491,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,2,0,0,0,0,1,0,0,150,25,0.17,0.03,0.17,0,0,0,0.05,0	1
2	0,2,40,9,146,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,13,1,0,0,0,0,0.08,0.15,0,255,1,0,0.6,0.88,0,0,0,0,0	1
3	0,1,44,5,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,123,6,1,1,0,0,0.05,0.07,0,255,26,0.1,0.05,0,0,1,1,0,0	0
4	0,1,44,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,121,19,0,0,1,1,0.16,0.06,0,255,19,0.07,0.07,0,0,0,0,1,1	0
5	0,1,44,5,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,166,9,1,1,0,0,0.05,0.06,0,255,9,0.04,0.05,0,0,1,1,0,0	0

TABLE VI. AFTER NORMALIZATION

No	*Instances*	*Output*
1	0,0.5,0.296875,0.9,5.48E06,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0.003913894,0.003913894,0,0,0,0,1,0,0,,0.588235294,0.098039216,0.17,0.03,0.17,0,0,0,0.05,0	1
2	0,1,0.625,0.9,1.63E06,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0.025440313,0.001956947,0,0,0,0,0.08,0.15,0,1,0.003921569,0,0.6,0.88,0,0,0,0,0	1
3	0,0.5,0.6875,0.5,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0.240704501,0.011741683,1,1,0,0,0.05,0.07,0,1,0.101960784,0.1,0.05,0,0,1,1,0,0	0
4	0,0.5,0.6875,0.1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0.236790607,0.037181996,0,0,1,1,0.16,0.06,0,1,0.074509804,0.07,0.07,0,0,0,0,1,1	0
5	0,0.5,0.6875,0.5,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0.324853229,0.017612524,1,1,0,0,0.05,0.06,0,1,0.035294118,0.04,0.05,0,0,1,1,0,0	0

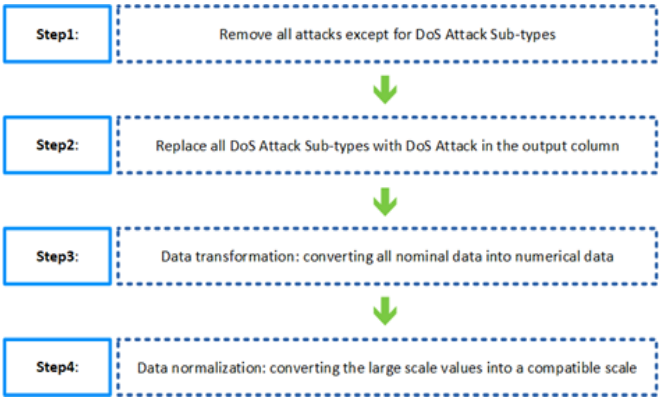

Fig. 3. NSL-KDD Dataset Data Preprocessing Steps.

B. DoS Attack Detection

This section contains a detailed discussion of our detection model of DoS attacks. As discussed earlier, the NSL-KDD dataset has been preprocessed to be prepared for the machine learning techniques. At first, besides the normal traffic, the NSL-KDD dataset has been filtered to contain only the sub-attack types that cause the DoS attack. These sub-attack types include: Back, land, Neptune, pod, smurf, teardrop, mailbomb, processtable, udpstorm, apache2, and worm. Then, all these sub-attack types have been labeled as DoS attack in the output column. Table VII shows the number of records of each sub-attack type and, eventually, the DoS attack. As such, now the NSL-KDD dataset contains only DoS attack type and normal traffic data. Then, the nominal features have been transformed using the label encoding technique including the output column. After that, the NSL-KDD dataset was normalized using the Min-max scaler technique (as discussed above). At this point, the NSL-KDD dataset is preprocessed and ready for the machine learning techniques to be applied. The generated NSL-KDD dataset was utilized to train and test the suggested DoS attack detection model.

The resulted NSL-KDD dataset (after data preprocessing) contains well well "labeled" data. In addition, the output variable is categorical; DoS attack and normal data. Therefore, the classification algorithms within the supervised machine learning are used in the proposed DoS attack detection model. Accordingly, the machine learning techniques used in the proposed model are Naive Bayes, KNN, Decision Tree, and SVM. The technique with the best performance measures, as shown below, will be determined for the proposed system. The K-Fold Cross-Validation technique has been used to validate the proposed model. In which, the NSL-KDD dataset has been divided into five groups. Four groups are used to train the used machine learning technique in each iteration, and the remaining one is used to test the used technique. In this way, each group is used to test the entire dataset. After testing and training, the suitable machine learning technique to detect DoS attacks was determined. Consequently, the traffic is analyzed using a high-performance machine learning technique that distinguishes between normal traffic and DoS attack traffic. Fig. 4 clarifies the proposed DoS attack detection model.

TABLE VII. NUMBER OF RECORD FOR DOS TYPES

Attack Type	*Number of records*
back attack	1315
land attack	25
neptune attack	45871
pod attack	242
smurf attack	3311
teardrop attack	904
mailbomb attack	293
processtable attack	685
udpstorm attack	2
apache2 attack	737
worm attack	2
Total	**53387**

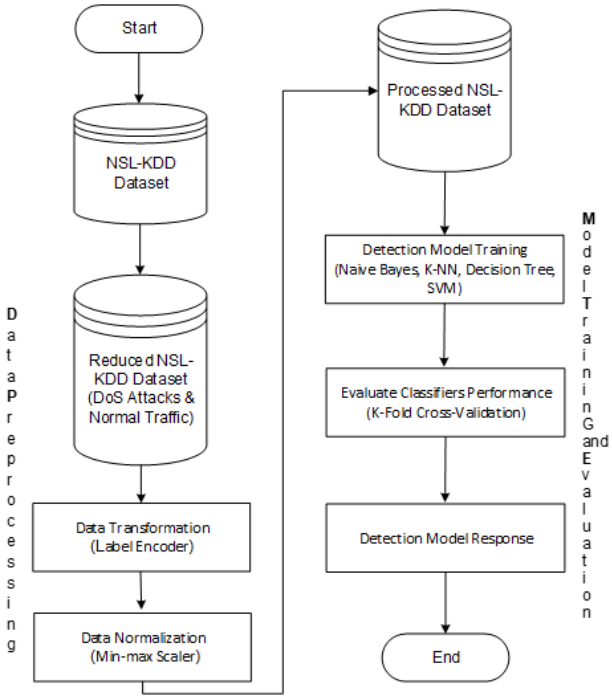

Fig. 4. DoS Attack Detection Model.

V. Performance Evaluation

This section examines the suggested DoS attack detection model's performance. The proposed model was designed using the Python programming language. Python is easy to use and widely used with machine learning. It provides several built-in tools specifically for machine learning that simplify complex tasks. The device used for testing has Intel Core i7-9750H processor and 32GB RAM with 64 bit MS-Windows.

The confusion matrix contains four elements [20][21] that summarize the performance of a proposed machine learning model:

1) True Positive (TP): indicates an attack and that the detection model successfully predicted this attack.

2) True Negative (TN): indicates no attack and the detection model successfully predicted no attack.

3) False Positive (FP): indicates no attack and the detection model wrongly predicted an attack.

4) False Negative (FN): indicates an attack and the detection model wrongly predicted no attack.

Fig. 5 elaborates the confusion matrix. The target of the proposed model is to increase the TP and TN and decrease the FP and FN.

Four measures have been employed to evaluate the proposed system based on the elements of the confusion matrix. These measures are Accuracy, Recall, Precision, and Matthews Correlation Coefficients (MCC). Accuracy is the ratio of properly forecasted attacks to the total number of forecasted attacks. Accuracy can be calculated using Equation 4. The Recall is the number of samples in the attack class that is successfully predicted to the total number of the prediction of the attack class. Recall can be calculated using Equation 5. Precision is the number of attacks that are correctly predicted as an attack to the number of attacks that are predicted as an attack. Precision can be calculated using Equation 6. MCC is a measure of the quality of classification with two classes. The closer the value to 1 indicates a more accurate classification. MCC can be calculated using Equation 7 [7][9][20][21].

Fig. 6, 7, 8, and 9 show the Accuracy, Recall, Precision, and MCC of the proposed model with the four tested techniques: Naive Bayes, KNN, Decision Tree, and SVM. Fig. 6, 7, 8, and 9 show that the Decision Tree technique achieved the highest performance with all four metrics: Accuracy (99.891%), Recall (99.904%), Precision (99.912%), and MCC (99.964%). On the other hand, the Naive Bayes technique achieved the lowest performance with all four metrics: Accuracy (94.472%), Recall (98.114%), Precision (92.923%), and MCC (88.643%). In general, all techniques perform well with the proposed model, except for the Naive Bayes technique. However, the Decision Tree technique could be considered as the best among the four techniques because it outperforms the other techniques in all four metrics.

$$Accuracy = \frac{(TP+TN)}{(TP+TN+FP+FN)} \quad (4)$$

$$Recall = \frac{TP}{(TP+FN)} \quad (5)$$

$$Precision = \frac{TP}{(TP+FP)} \quad (6)$$

$$MCC = \frac{((TP*TN)-(FP*FN))}{\sqrt{(TP+FP)*(TP+FN)*(TN+FP)*(TN+FN)}} \quad (7)$$

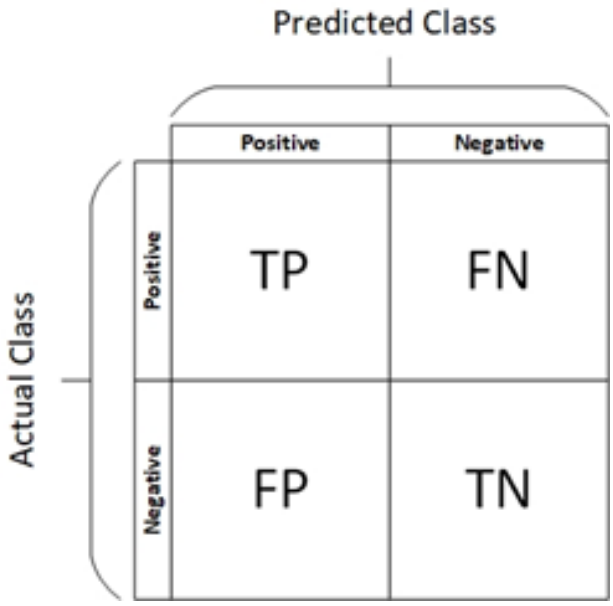

Fig. 5. Confusion Matrix.

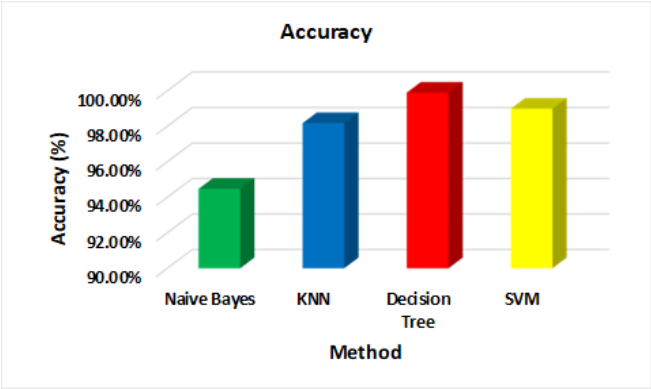

Fig. 6. Accuracy of the Proposed Model.

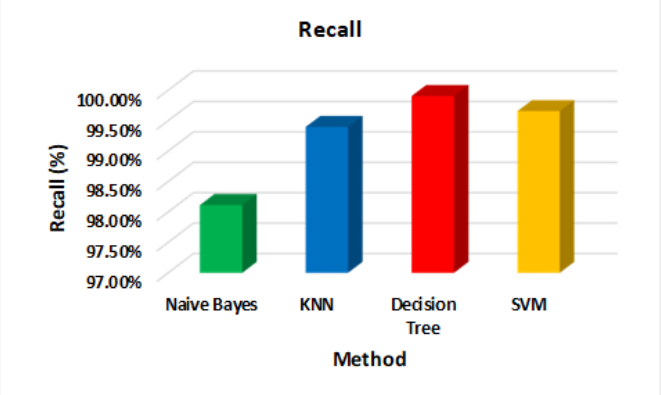

Fig. 7. Recall of the Proposed Model.

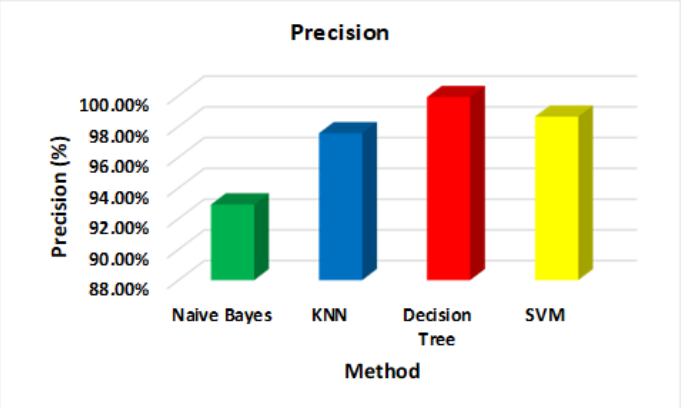

Fig. 8. Precision of the Proposed Model.

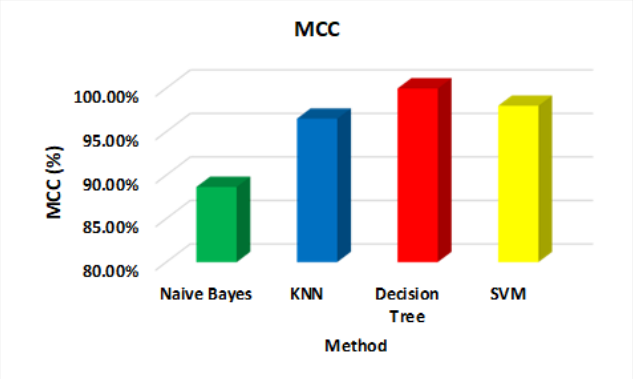

Fig. 9. MCC of the Proposed Model.

VI. Conclusion

DoS is a hazardous attack that threatens governments, businesses, and individuals. New techniques to launch DoS attacks emerge continuously. These techniques required an adaptive system to mitigate them. This paper developed a new paradigm for disclosing DoS attacks using machine learning approaches. The proposed model's primary objective is to mitigate existing and newly discovered DoS attack types. Several machine learning techniques were Naive investigated with the proposed model. Among these techniques, the Decision Tree technique has shown the highest performance. Whereas the Accuracy, Recall, Precision, and MCC, of the Decision Tree technique with the proposed model is 99.891%, 99.904%. 99.912%, and 99.964%, respectively. Therefore, the proposed detection model is promising for mitigating the newly emerged DoS attack types.

References

[1] M. Bang and H. Saraswat, "Building an effective and efficient continuous web application security program," 2016 International Conference On Cyber Situational Awareness, Data Analytics And Assessment (CyberSA), 2016, pp. 1-4, DOI: 10.1109/CyberSA.2016.7503287.

[2] Symantec internet security threat report 2018 Volume 23, Symantec, 2018.

[3] Cisco Annual Internet Report (2018–2023) White Paper, March 9, 2020, Cisco.

[4] P. R. Chandre, P. N. Mahalle and G. R. Shinde, "Machine Learning Based Novel Approach for Intrusion Detection and Prevention System: A Tool Based Verification," IEEE Global Conference on Wireless Computing and Networking (GCWCN), 2018, pp. 135-140, DOI: 10.1109/GCWCN.2018.8668618.

[5] S. Peneti and H. E, "DDOS Attack Identification using Machine Learning Techniques," International Conference on Computer Communication and Informatics (ICCCI), 2021, pp. 1-5, DOI: 10.1109/ICCCI50826.2021.9402441.

[6] K. Hara and K. Shiomoto, "Intrusion Detection System using Semi-Supervised Learning with Adversarial Auto-encoder," NOMS 2020 - 2020 IEEE/IFIP Network Operations and Management Symposium, 2020, pp. 1-8, DOI: 10.1109/NOMS47738.2020.9110343.

[7] Çavuşoğlu, Ü. (2019). A new hybrid approach for intrusion detection using machine learning methods. Applied Intelligence, 49(7), 2735-2761.

[8] I. Abrar, Z. Ayub, F. Masoodi and A. M. Bamhdi, "A Machine Learning Approach for Intrusion Detection System on NSL-KDD Dataset," International Conference on Smart Electronics and Communication (ICOSEC), 2020, pp. 919-924, DOI: 10.1109/ICOSEC49089.2020.9215232.

[9] Ravipati, Rama Devi, and Munther Abualkibash. "Intrusion detection system classification using different machine learning algorithms on KDD-99 and NSL-KDD datasets-a review paper." International Journal of Computer Science & Information Technology (IJCSIT) Vol 11 (2019).

[10] T. M. Ma, K. YAMAMORI and A. Thida, "A Comparative Approach to Naïve Bayes Classifier and Support Vector Machine for Email Spam Classification," 2020 IEEE 9th Global Conference on Consumer Electronics (GCCE), pp. 324-326, DOI: 10.1109/GCCE50665.2020.9291921.

[11] P. Wang, Y. Zhang and W. Jiang, "Application of K-Nearest Neighbor (KNN) Algorithm for Human Action Recognition," IEEE 4th Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC), 2021, pp. 492-496, DOI: 10.1109/IMCEC51613.2021.9482165.

[12] H. Elaidi, Y. Elhaddar, Z. Benabbou and H. Abbar, "An idea of a clustering algorithm using support vector machines based on binary decision tree," International Conference on Intelligent Systems and Computer Vision (ISCV), 2018, pp. 1-5, DOI: 10.1109/ISACV.2018.8354024.

[13] Ahsan, Md Manjurul, et al. "Effect of data scaling methods on machine learning algorithms and model performance." Technologies 9.3 (2021): 52.

[14] T. Wong and N. Yang, "Dependency Analysis of Accuracy Estimates in k-Fold Cross Validation," in IEEE Transactions on Knowledge and Data Engineering, vol. 29, no. 11, pp. 2417-2427, 1 Nov. 2017, DOI: 10.1109/TKDE.2017.2740926.

[15] Wankhede, Shreekh, and Deepak Kshirsagar. "DoS attack detection using machine learning and neural network." Fourth International Conference on Computing Communication Control and Automation (ICCUBEA). IEEE, 2018.

[16] Zhe, Wang, Cheng Wei, and Li Chunlin. "DoS attack detection model of smart grid based on machine learning method." IEEE International Conference on Power, Intelligent Computing and Systems (ICPICS). IEEE, 2020.

[17] He, Zecheng, Tianwei Zhang, and Ruby B. Lee. "Machine learning based DDoS attack detection from source side in cloud." IEEE 4th International Conference on Cyber Security and Cloud Computing (CSCloud). IEEE, 2017.

[18] A. K. B and M. M. Kodabagi, "Efficient Data Preprocessing approach for Imbalanced Data in Email Classification System," 2020 International Conference on Smart Technologies in Computing, Electrical and Electronics (ICSTCEE), 2020, pp. 338-341, DOI: 10.1109/ICSTCEE49637.2020.9277221.

[19] B. -B. Jia and M. -L. Zhang, "Multi-Dimensional Classification via Decomposed Label Encoding," in IEEE Transactions on Knowledge and Data Engineering, DOI: 10.1109/TKDE.2021.3100436.

[20] M. M. S. Pangaliman, F. R. G. Cruz and T. M. Amado, "Machine Learning Predictive Models for Improved Acoustic Disdrometer," IEEE 10th International Conference on Humanoid, Nanotechnology, Information Technology,Communication and Control, Environment and

Management (HNICEM), 2018, pp. 1-5, DOI: 10.1109/HNICEM. 2018.8666256.

[21] N. Ajithkumar, P. Aswathi and R. R. Bhavani, "Identification of an effective learning approach to landmine detection," 1st International Conference on Electronics, Materials Engineering and Nano-Technology (IEMENTech), 2017, pp. 1-5, DOI: 10.1109/IEMENTECH.2017.8077018.