JOURNAL OF NORTHEASTERN UNIVERSITY

Volume 25 Issue 04, 2022 ISSN: 1005-3026 https://dbdxxb.cn/ Original Research Paper

AN EFFICIENT INTRUSION DETECTION SYSTEMS USING MACHINE LEARNING MODELS

S.P.Senthilkumar
Research Scholar, Department of Computer & Information Science, Annamalai University
Email: senthil.sp74@gmail.com

Dr.Aranga.Arivarasan
Assistant Professor/Programmer, Department of Computer & Information Science
Annamalai University, Email: profarivarasan@yahoo.com

ABSTRACT

The global market takes place on the Internet. A computer network will undoubtedly be necessary for any organization to be successful. However, security becomes a major concern when connecting your business to a network, as your data is more vulnerable to attacks from hostile individuals. An intrusion detection system (IDS) comes in handy in this situation. In this article, we employ Random Forest classifier and Naive Bayes classifier to swiftly identify threats in an IDS because to the vastness of the CICIDS2017 dataset. The model test was performed in Python. The CICIDS2017 dataset was used to test our technique and the results show that the Random Forest classification achieves the highest accuracy for training and testing.

Keywords: Random Forest classifier, Naïve Bayes, Intrusion Detection Systems, CICIDS2017

1) Introduction

Networks have become a vital form of contemporary life, and cyber security as a study topic has grown in importance. As attacks become more sophisticated, it becomes increasingly difficult to find attackers on the network. If such an intrusion is not prevented or acknowledged, it can have catastrophic consequences. The Intrusion Detection System (IDS), a key Cybersecurity tool, checks the functionality of network devices and software. Although traditional IDSs have been in development for years, they continue to struggle to improve prediction performance, minimize false positives, and identify threats. Many scientists have worked on building IDS using machine learning methods to solve the above difficulties. Machine learning techniques aim to recognize network connection patterns to distinguish invisible intruders from known intruders, but they also require regular retraining to maintain high performance. Machine learning algorithms can detect big variations between normal and abnormal data automatically and accurately. Because ML algorithms techniques are very generalizable, they can detect previously unknown assaults. Network managers are confronted with extremely complicated software and applications as the number of Internet users grows and web-based services proliferate. Due to their vulnerabilities, networks are becoming increasingly vulnerable to cyber attacks. An IDS monitors malicious network activity and responds to an ever-growing array of network threats (e.g. DDoS attacks,

Submitted: 08/12/2022 **Accepted**: 27/12/2022

ransomware attacks, botnet attacks, etc.). Therefore, cyber security strives to keep networks as secure as possible by using defense mechanisms to detect suspicious activity. However, firewalls and intrusion detection systems (IDS) suffer from frequently updating their threat detection databases. As a result, there lot of new flaws in systems is growing, which leads to an increase in cyber threats, particularly zero-day assaults. Security hazards are among the most serious challenges to the information technology industry. Cybercrime is anticipated to cost approximately $ 3 billion per year. By 2022, this figure is predicted to more than double.

IDS are a well-known network architectural solution for maintaining the integrity and availability of sensitive resources in secure systems. Despite the adoption of numerous computational algorithms to improve IDS performance, existing IDS continue to fail. First, a massive amount of redundant and irrelevant data in large datasets disturbs an IDS's categorization process. Second, it is possible that a single classifier would not be possible to perceive all forms of attacks. Third, many models rely out datasets, making them difficult for threat generation.

Multiple cyber-attacks and security flaws have jeopardized crucial company data. One of these sorts of danger is intrusion. Intrusion attempts are attempts to evade computer systems' standard security features. The IDS is the most significant piece of network security equipment. This document provides an overview of the IDS and assists the reader in comprehending some of the key ideas and methodologies of the IDS. This research also includes several types of IDS network attack methodologies and types, and an extensive literature evaluation.

2. Literature review

Sneh Lata Pundir et al. proposed a [1] random forest to select important features from the kddcup'99 dataset. Ansam Khraisat et al. provided a survey of IDS [2] methodologies and types, as well as their benefits and drawbacks. A few machine learning techniques to detect novel attacks are explored. Paulo Angelo et al. have stated that, a survey of [3] Random Forest-based techniques used in this frame of reference, taking into consideration the particularities of these models. Hamed Alqahtani, et al expressed that, they employ [4] ML algorithms, like videlicet Bayesian Network, Naive Bayes classifier, Decision Tree, Random Decision Forest, Random Tree, Decision Table, and Artificial Neural Network, to descry intrusions. Heitor Scalco Neto, Wilian Soares Lacerda, and Rafael Vero Françozo carried out [5] the findings acquired with the Random Forests technique reveal precision rates from around 98 percent, going to bring huge progress in the area of Intrusion Detection. The classifiers have [6] been invented by Salim Qadir Mohammed et al. using the Python library Scikit-Learn. KNN, Support Vector Machine, Naive Bayes, Decision Tree, Random Forest, Stochastic Gradient Descent, Gradient Boosting, and Ada Boosting. With an accuracy of only 99.96 percent and a failure rate of 0.038 percent, the RF classification outperforms well. Each ML algorithm [7] has its own set of strengths and weaknesses, and they should be used with caution to achieve the security system's objectives. Nabila Farnaaz et al. invented [8] an IDS model using a RF classifier. They did run experiments on the NSL-KDD data set to monitor the efficiency of their framework. Finding revealed that their suggested method is effective. Kapil Sachan, et al.

presented [9] the random forest classification algorithm and principal component analysis to create effective intrusion detection systems. The presented method surpassed previously utilized algorithms including such SVM, Naive Bayes, and Decision Tree. Perala Karishma et al. resume [10] that PCA will help organize the dataset by reducing its dimensionality. The outcomes show that the enhanced method beats competing techniques like SVM, Naive Bayes, and Decision Tree in terms of high accuracy. Kritika Singh and Bharti Nagpal presented a review of the [11] RF algorithm and an overview of the various survey techniques currently in use has also been compiled.(Nagpal, 2018). Ashwini and Sakshi Pathak claim to have [12] selected the best attributes for each type of attack using the ANOVA f-test of the dataset. The results show that the decision tree method outperforms the ANN in terms of overall efficiency. According to B.Yogesha et al. the classification [13] techniques are used for evaluating NSL Dataset with features. This project's classification methods include SVM, RFC, K Neighbors Classifier, Logistic Regression, and Naive Bayes. According to the results, the RFC is more efficient and more accurate than the other classifications. According to Zhiqiang Liu and Yucheng Shi, an IDS based [14] on the RF classifier. The primary benefit of our suggested framework is that it enhances the accuracy results of the classic RF by concentrating on essential features and lessening training time. Tao Wuet et al.convey a NIDS [15] algorithm that is based on a robust Random Forest Synthetic Minority Oversampling (SMOTE) technique. The performance has been measured in this paper using the NSL-KDD dataset, with an accuracy rate of 99.72 % on the training set and 78.47 % on the test set. Zhewei Chen et al. said, [16] the Adaptive Synthetic Sampling (ADASYN) method was suggested in this study to balance the datasets. In a contrasting intrusion detection investigation using the CICIDS2017 dataset, ADASYN outperformed Random Forest.

3. Proposed Work

The intrusion detection model developed in this article uses classical machine learning algorithms including Random Forest and Naive Bayes, which are extensively used in similar studies. In Figure 1, the potential intrusion detection models are displayed. The CICIDS2017 dataset includes information on new malware attacks such as Brute Force FTP, Brute Force SSH, DoS, Heartbleed, Web Attack, Infiltration, Botnet and DDoS, as well as details on benign behaviors [17] . The timestamp, source and destination IP addresses, source and destination ports, protocols, and attacks are used to label this dataset. This dataset was gathered using a fully configured network architecture, which includes nodes running Linux, Apple's macOS iOS, Microsoft Windows (including Windows 10, Windows 8, Windows 7, and Windows XP), as well as modem, firewall, switches, and routers. 80 network flow features from the recorded network traffic may be found in this collection.

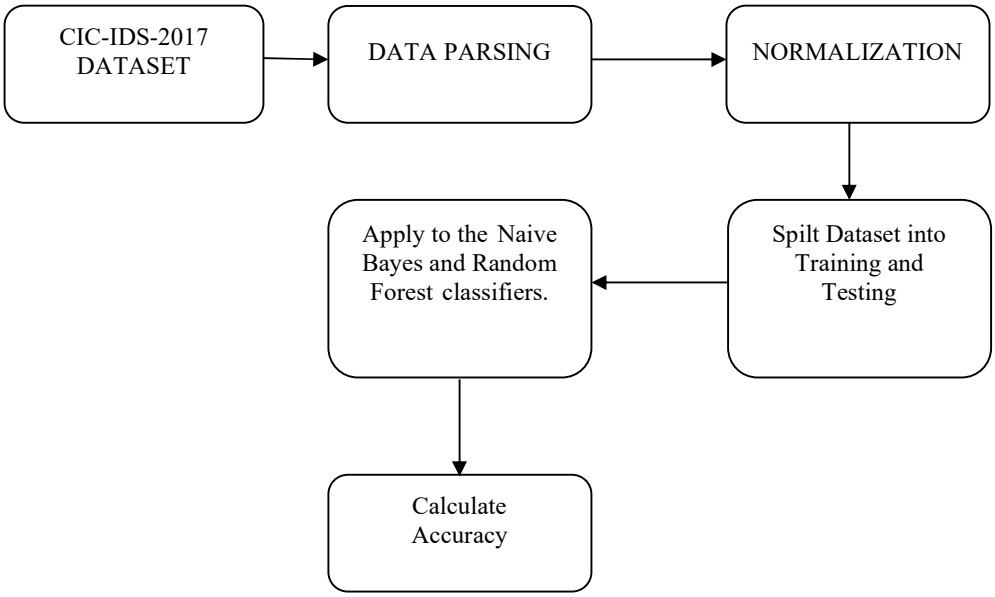


Fig.1: Proposed work

4. Dataset

Getting reliable data is the most important and tedious step to start using machine learning models. We use the CIS-IDS-2017 dataset from the Canadian Institute of Cybersecurity. According to the eleven datasets available since 1998, most are outdated and unreliable. These datasets range from covering a limited spectrum of known attacks to anonymizing packet payload data, making it impossible to reflect current trends. The dataset available from CIC contained 2,830,743 records of IDS related data organized in 8 different files. 79 parameters were available for each record, of which 78 columns were used as input parameters and the 79th column was used as the prediction parameter. Around 5000 records contained values unsuitable for unsupervised machine learning (like NaN) and the same were removed. Column number 79 of the CIC dataset represented the threat level associated with the input parameters presented in columns 1 to 78. Since the target values were available in the form of strings in the CIC dataset, the following numeric codes were used to replace the strings (as depicted in Table 1).

Table 1: Attack levels, Numeric codes and Number of records

Numeric code.	Attack level	Number of records
1	BENIGN	24,01,124
2	Bot	1,966
3	DoS GoldenEye	10,293
4	DoS Hulk	2,31,073
5	DoS Slowhttptest	5,499
6	DoS slowloris	5,796

7	FTP-Patator	7,938
8	Heartbleed	11
9	Infiltration	36
10	PortScan	1,58,930
11	SSH-Patator	5,897
12	Web Attack - Brute Force	1,507
13	Web Attack - Sql Injection	21
14	Web Attack - XSS	652
Total records		**28,30,743**

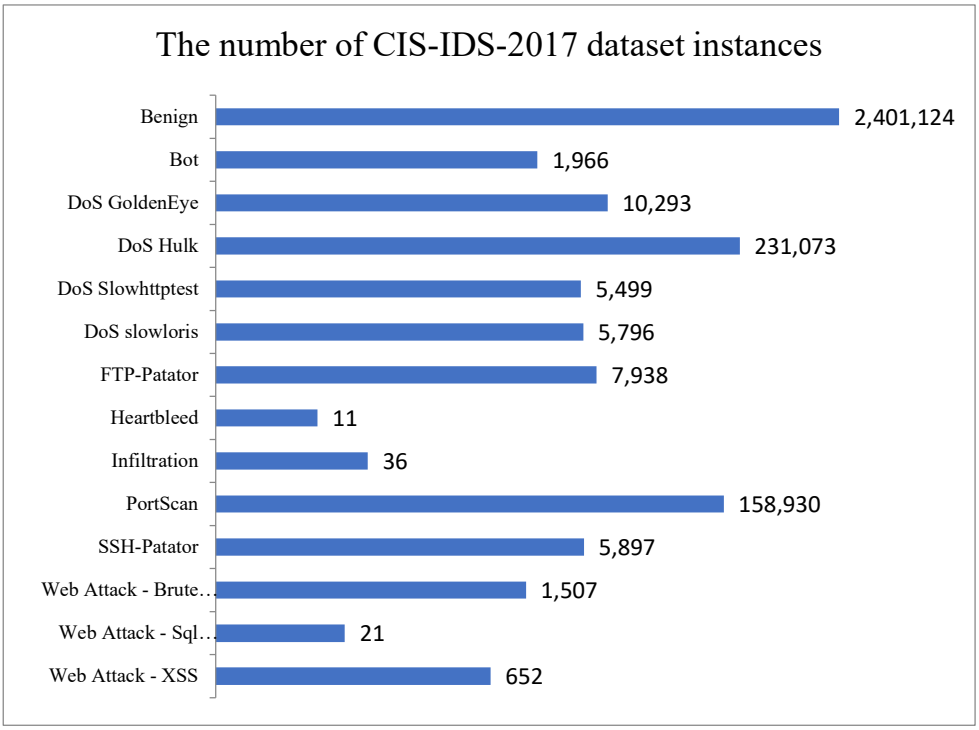


Fig.2: Graphical representation of Instances

Destination port, flow time, total forwarded packets, etc. are the column titles. The prediction parameter, called Label, is found in column 79. The information is collected in one .csv file. The pandas package read csv function was used to read the previously prepared data. Using the split train test function in Scikit Learn's sklearn.model selection module, the data read in this way was split into training and test sets. A selection of 70% of the data was used for training, while the remaining 30% was used for testing.

5. Machine Learning algorithms

Random Forest Classifier

As an ensemble classifier, Random Forest integrates a variety of classification techniques. On a randomly selected piece of data they generate several decision trees. The test class is then determined by adding up all the grades from each tree or by assigning a certain weight to each tree's grade.

The bagging algorithm forms the basis of Random Forest, which uses ensemble learning. The output of all trees is combined once as many trees as possible are created in the subset of the data. This reduces the problem of over fitting decision trees, as well as variance and precision. Classification and regression problems can be solved with Random Forest.

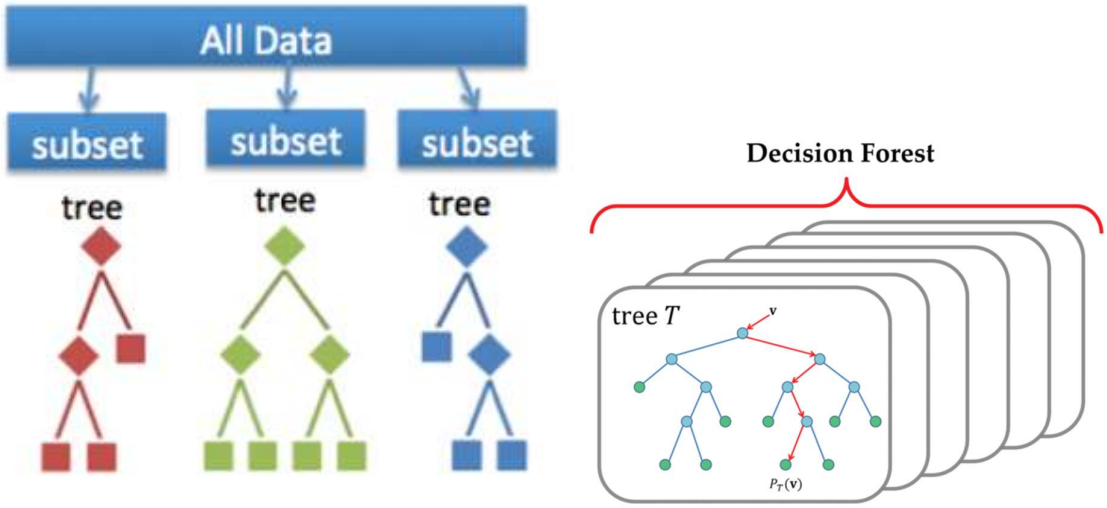


Fig. 3. Random forest classification.

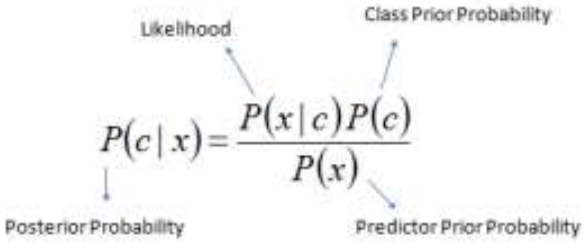
Python implementation strategy

```
import numpy as np
import pandas as pd
from sklearn import datasets
from sklearn.model_selection import train_test_split
from mlxtend.plotting import plot_decision_regions
from sklearn.metrics import accuracy_score
from sklearn.ensemble import RandomForestClassifier
from sklearn.datasets import load_iris
import pickle
import os
import datetime

modelFile = 'rfccicds.obj'

readStart = datetime.datetime.now()

ids = pd.read_csv(r'dt.csv')
print(ids)
X = ids.iloc[:,:-1]
y = ids.iloc[:,-1]
X
y

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=1, stratify=y)
print(X_train, X_test, y_train, y_test)

re = datetime.datetime.now()
print("Data has been read from CSV file. Time taken: ", (re-readStart))
```

```
if(os.path.exists(modelFile)):
    forest = pickle.load(open(modelFile,'rb'))
    me = datetime.datetime.now();
    print("Random Forest classifier loaded from file. Time taken for reading: ",(me-re))
else:
    forest = RandomForestClassifier(criterion='gini',n_estimators=5, random_state=1, n_jobs=1)
    forest.fit(X_train, y_train)
    pickle.dump(forest,open(modelFile,'wb'))
    ce = datetime.datetime.now()
    print("New Random Forest Classifier has been created and saved to file ", modelFile, "Time taken: ",(ce-re))

ps = datetime.datetime.now()
y_pred = forest.predict(X_test)
acc = accuracy_score(y_test, y_pred)*100
print('Accuracy: %.3f%%' % acc)
pe = datetime.datetime.now()

print("Prediction completed and accuracy ascertained. Time taken: ", (pe-ps))

print("Total time taken: ",(pe-readStart));
```

Naive Bayes algorithm

Under the premise of predictor independence, it is a classification strategy based on Bayes' theorem. Simply put, a Naive Bayes classification assumes that the existence of a trait in a class is unrelated to the presence of any other trait. For example, fruits that are red, round, and about 3 inches in diameter can be classified as apples. Even if these traits are interdependent or based on the presence of other traits, each of these traits increases the likelihood that this fruit is an apple, which is why it is called "naive."



$$P(c \mid X) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

Above,

P(c|x) is the posterior probability of class (c, target) given predictor (x, attributes).

P(c) is the prior probability of class.

P(x|c) is the likelihood which is the probability of predictor given class.

P(x) is the prior probability of predictor.

A similar approach is used by Naive Bayes to predict the probability of different classes based on different attributes. When there are problems with multiple classes, this approach is mostly used in text classification.

Python implementation strategy

Scikit learn (python library) will help here to build a Naive Bayes model in Python. There are three types of Naive Bayes model under scikit learn library:

```
model = GaussianNB()
model.fit(train_X, train_y)
prediction = model.predict(test_X) #prediction based on testing_set

print("The accuracy of NB is: ", metrics.accuracy_score(prediction, test_y))

expected = train_y #expected result stored in training_set

predicted = model.predict(train_X)

print( "Classification report:")

print(metrics.classification_report(expected, predicted)) #calculating classification report
print( "Confusion matrix:")

print(metrics.confusion_matrix(expected, predicted)) #calculating confusion matrix
```

Classification report

The classification report is a conceptual visualization that shows the four basic parameters of a classification model: accuracy, recall, F1 score, and support, to provide the level of accuracy as a result of model customization.

Accuracy

The most important indicator of the effectiveness of a classification model (classifier) is accuracy. It refers to how well the algorithm can anticipate data that has not yet been observed while learning the data patterns in the dataset.

Precision

Accuracy should be considered an important performance metric. This ratio measures the ratio of closely observed positives to all observed positives.

Recall

Recall is the ratio of closely observed positives to all other observations in a class. The ratio of positive observations indicates how the result is presented.

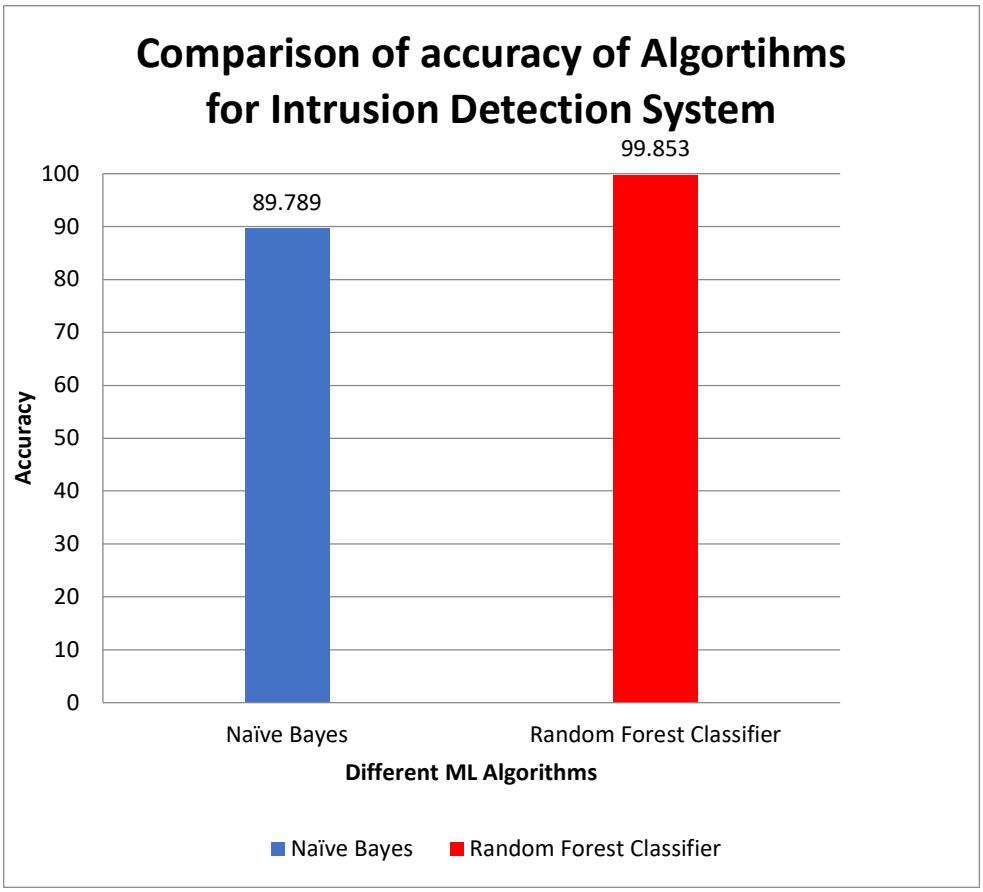
F1 Score

The F1 score should be considered as a crucial performance metric. F1 score may be more important than accuracy in some situations. The costs of false positives and false negatives can sometimes differ in a large data set. Accuracy is preferable if they are identical. However, we have to look at the F1 score if they are not the same.

Support

The set of actual results observed in a class is called support. It shows the proportion of valid results where the results obtained were identical to those predicted.

	Accuracy	Precision	Recall	F – Measure
Naïve Bayes	89.789	0.964	0.963	0.963
Random Forest	99.853	0.997	0.997	0.997



Results and Discussion

The results of running the models through Python program on a computer with an Intel Celeron processor, 4 GB of RAM and 256 GB of SSD. Naive Bayes and Random Forest Classifier both provide accuracy results of 99.853% and 89.789%, respectively, against test data. To apply the required algorithms and other crucial library packages to the classification model, we import all the necessary components. To import all required functions and algorithms, we use the scikit-learn library package. The dataset is split into two sets, a training set and a test set, using a 70:30 ratio, with 70% of the data used for training and the remaining 30% used for testing or validation the classification model. After the completion of the training and testing phases, we used the two algorithms sequentially for the same training set and verified our results using the test set to prepare the classification report.

Conclusion

We look at the many patterns and forms of malicious intrusion attempts in the CIS dataset in this research, and then we suggest and present a Random Forest Classifier and Nave Bayes-based model for intrusion detection systems. The most recent CIC-IDS-2017 dataset, which accurately depicts current traffic levels, is used in this study. The dataset was processed using the SCIKIT Python package; CIC-IDS-2017 dataset was trained on 70% of the records,

testing on the remaining 30% of the records showed the accuracy of the results to be 99.853% and the Naive Bayes classifier returned an accuracy of 89.789%. The generated models surpass previous research in this area in terms of accuracy and have a higher attack detection rate. In this study, supervised learning methods were employed to identify attacks using the CIS-IDS-2017 dataset. Naive Bayes is the fastest, whereas Random Forest has the highest accuracy (99.853%). In our upcoming study, we will train and evaluate deep learning models like Convolutional neural networks and recurrent neural networks against conventional machine learning techniques in terms of accuracy and speed. To classify computer security threats in real time, the Random Forest Classifier object developed using the CIC dataset is useful. Deep learning and neural networks will be used in the follow-up research for this study to improve training of the data set. Use technologies like Tensorflow to quickly and accurately build a better learning model. Additionally, efforts can be made to improve the models to better detect and defend against known attacks.

References

1) Amrita, S. L. (2013)., "Feature Selection using Random Forest in Intrusion Detection System", *International Journal of Advances in Engineering & Technology* , 1319-1324.
2) Ansam Khraisat, I. G. (2019) , "Survey of intrusion detection systems:techniques, datasets and challenges", *Cybersecurity* , 1-22.
3) Drummond, P. A. (2019), "A Survey of Random Forest Based Methods for Intrusion Detection Systems", *ACM Journals* , 1-36.
4) Hamed Alqahtani, I. S. (2020) "Cyber Intrusion Detection Using Machine Learning Classification Techniques", *International Conference on Computing Science, Communication and Security* (pp. 121-131). Gujarat: Springer Nature Singapore Pte Ltd.
5) Heitor Scalco Neto, W. S. (2021)," Random Forests for Online Intrusion Detection in Computer Networks", *Journal o f Computer Sci ence* , 906-914.
6) Hussein, S. Q. (2022), "Performance Analysis of different Machine Learning Models for Intrusion Detection Systems", *Journal of Engineering* , 61-91.
7) S.P.Senthilkumar , Dr.Aranga.Arivarasan (2022), "Empirical Analysis of Machine Learning Models towards Adaptive Network Intrusion Detection Systems", Proceedings of the Fourth International Conference on Smart Systems and Inventive Technology (ICSSIT-2022) , published by IEEE, pp.198 -206.
8) Jabbar, N. F. (2016) "Random Forest Modeling for Network Intrusion Detection System", *Twelfth International Multi-Conference on Information Processing-2016 (IMCIP-2016)* (pp. 213-217). Bangalore: ScienceDirect.
9) Kapil Sachan, A. P. (2022),"Network Intrusion Detection System Using Random Forest and PCA", *Journal of Advanced Research in Computer and Communication Engineering* , 521-525.

10) Lakshmi, P. K. (July 2021), "Intrusion Detection System Using PCA with Random Forest Approach", *Journal of Engineering Sciences* , 92-99.
11) Nagpal, K. S. (2018) , "Random Forest Algorithm in Intrusion Detection System : A Survey", *International Journal of Scientific Research in Computer Science, Engineering and Information Technology* , 673-676.
12) Pathak, A. P. (2020), "Study on Decision Tree and KNN Algorithm for Intrusion Detection System", *International Journal of Engineering Research & Technology* , 376-381.
13) Reddy, B. a. (2022), "Intrusion detection System using Random Forest Approach", *Turkish Journal of Computer and Mathematics Education* , 725-733.
14) Shi, Z. L. (2022), "A Hybrid IDS Using GA Based Feature Selection Method and Random Forest", *International Journal of Machine Learning and Computing,* , 43-50.
15) Tao Wu, H. F. (2022), "Intrusion detection system combined enhanced random forest with SMOTE algorithm", *EURASIP Journal on Advances in Signal Processing* , 1-20.
16) Zhewei Chen, W. Y. (2021), "ADASYN－Random Forest Based Intrusion Detection Model. *SPML 2021", 4th International Conference on Signal Processing and Machine Learning Beijing China August 18 - 20, 2021* (pp. 152-159). Association for Computing Machinery ,New YorkNY, United States.
17) Iman Sharafaldin, Arash Habibi Lashkari and Ali A. Ghorbani, "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization*", Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP 2018)*, pages 108-116