International Journal of Innovative Research in Computer Science & Technology (IJIRCST)
ISSN: 2347-5552, Volume-12, Special Issue-1, March-2024
https://doi.org/10.55524/CSISTW.2024.12.1.8
Computing for Sustainable Innovation: Shaping Tomorrow's World (CSISTW-2024) on 16th-17th February, 2024
Organized by B.N. College of Engineering and Technology, Lucknow, India

AI-Based Intrusion Detection System in Cloud Computing

Anupam Rathore[1], and Tripti Sahu[2]

[1]M. Tech Scholar, Department of CSE, B.N. College of Engineering & Technology, Lucknow, UP, India
[2]Assistant Professor, Department of CSE, B.N. College of Engineering & Technology, Lucknow, UP, India

Correspondence should be addressed to Anupam Rathore : anupamthakur983@gmail.com

ABSTRACT – It is possible to communicate with others and do business across this global network, which is comprised of hundreds of millions of computers running a variety of hardware and software configurations. This makes it simpler for hackers to abuse resources and conduct Internet attacks since computers are linked to one another. There are significant roadblocks to the development of a security-oriented approach that may be flexible and adaptive in light of the expanding number of Internet assaults. An intrusion detection system is necessary for the identification of online threats (IDS). Maintaining the security of a network requires the use of an intrusion detection system or IDS. Because the cloud platform is continuously expanding and becoming more prevalent in our everyday lives, it is imperative that we develop an effective IDS for it as well. On the other hand, typical intrusion detection systems may encounter difficulties when used in the cloud. A cloud segment may get overburdened by the pre-determined IDS architecture because of the added detection overhead. In the context of a networked system with an adaptable architecture. Using a neural network-based IDS, we show how to make full utilize available resources while not putting undue strain on any single cloud server. The proposed IDS uses a neural network machine learning to identify new threats even more effectively.

KEYWORDS- Artificial Intelligence, Intrusion Detection System, Machine learning Algorithm.

I. INTRODUCTION

Today, the Internet is an essential part of our everyday lives, and it's utilized in everything from commerce to entertainment to education. It has become increasingly common for businesses to use the Internet to get access to information. A computer system may be hacked in a variety of ways because of the wide availability of information on the Internet. Internet assaults and incursions are becoming increasingly commonplace. An incursion or assault may be described as "any combination of actions that seek to breach the security objectives". Some of the most important security objectives are availability, integrity, confidentiality, accountability, and assurance. Assaults may be divided into four categories: Probing, Denial of Service, User to Root, and Remote User. Several anti-intrusion technologies have been developed to stop a huge proportion of Internet assaults. Six anti-intrusion systems have been described by Halma and Bauer (1995), and they include IDS of them. The other five include detection and countermeasures. The most critical of these components is the ability to identify an incursion perfectly.

II. RELATED WORK

Using machine algorithms in a big data environment, Saud Mohammed Othman and Fadl Mutaher Ba-Kiwi [1] presented their work on an intrusion detection model. According to this study, Big Data's ever-increasing volume has shifted the significance placed on data security and analytic technologies. An IDS monitors and analyses data in order to discover any system or network intrusions that may have occurred. Traditional strategies for detecting network assaults have become increasingly complex due to the volume, diversity, and speed of data created in the network. IDS uses Big Data approaches to analyze Big Data in an accurate and effective manner.

By utilizing the Spark-Chi-SVM architecture, they were able to construct an intrusion detection model that is capable of managing enormous amounts of data. The Spark Big Data platform was utilized in the recommended approach in order to facilitate the handling and analysis of data in a timely manner. The large dimensionality of big data adds to the difficulty and length of the categorization process. A Survey of Intrusion Detection Systems utilizing Machine Learning Techniques was provided by Sharmila Wagh and Vinod K. Pachghare [3]. Computers and network-based technologies are becoming more and more commonplace in today's environment, according to the authors. The importance of network security cannot be overstated in today' s computer age. Detection of system assaults and classification of system activity into normal and abnormal forms are the goals of an Intrusion Detection System (IDS). IDS systems that use machine learning to identify intrusions have become increasingly commonplace.

An intrusion detection system based on distributed machine learning and the cloud is described in this study [4]. The suggested system on the cloud will be accompanied by edge network components from cloud providers. As a result, the edge network routers on the physical layer may intercept incoming network traffic. Before being transmitted to a module that employs the NaiveBayesclassifierr to identify anomalies, the network data collected by each Cloud router is preprocessed using a time-based sliding window technique. When there is a buildup of network congestion, server nodes that are powered by Hadoop and MapReduce are made accessible to each anomaly detection module. A

server for synchronizing anomalous network traffic data is assigned to each time frame. This server collects anomalous network traffic data from each router in the system. Following this, Random Forest classifiers are applied to each attack in order to establish the kind of assault that was committed.

Hassan Musafer and Ali Alessa' s study, titled " Machine Learning- Based Network Intrusion Detection Detection: Dimensionality Reduction Approaches, " was recently published. [5]. They claim that all parties, including consumers, businesses, and governments, are worried about the safety of computer networks. As it becomes increasingly difficult to safeguard networked systems against assaults, the strategies that attackers use to carry out such assaults also evolve. A portion of the solution is in making the existing intrusion detection systems more effective. The use of machine learning to create intrusion detection systems is an approach that is becoming more popular because of its efficiency (IDS). When improvements are made to IDS qualities like as discrimination and representation, there is a considerable increase in the system' s overall performance. Through the use of Deep Learning' s Auto-Encoder (AE) and Principal Component Using Principal Components Analysis (PCA), the dimensionality of the features were reduced for the purpose of this inquiry (PCA). It is possible that the combination of these two approaches will result in the acquisition of low-dimensional attributes that can be utilised in the building of classifiers such as Random Forest (RF), Bayesian Network, Linear Discriminant Analysis (LDA), and Quadratic Discriminant Analysis (QDA).

A. Intrusion Detection System-

One of the fundamental components of security infrastructure is an effective security system that can detect, prevent, and maybe respond to computer threats. FoSecurityervices, it uses a variety of methods, including audit and network traffic data in computer or network systems, to monitor target sources of activity. All threats must be swiftly and correctly identified by an intrusion detection system (IDS). IDS may help network managers find security flaws objectively. Intruders from the outside may attempt to get access to the network illegally. security infrastructure or make resources inaccessible to insiders who abuse their system resources, all of which are breaches of security objectives.

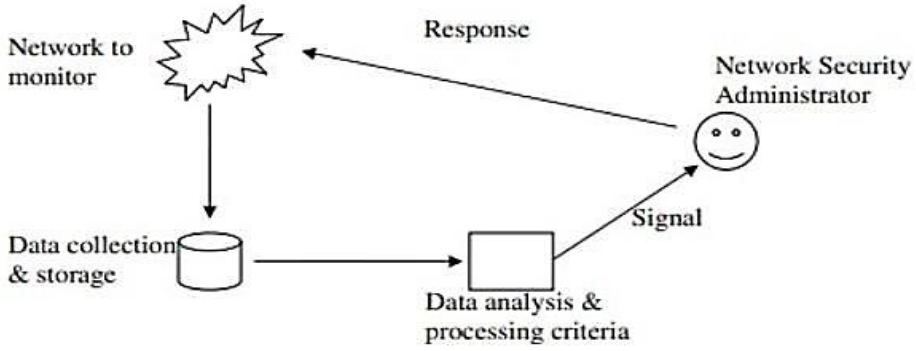


Figure 1: Architecture of Intrusion Detection System

As the number of computer attacks has risen, many IDS designs have been proposed. Axelson (1999) suggested a common design for IDS in Figure 1. The following are the most common IDS components, according to Axelson (1999): The network to be monitored must be identified to keep tabs on any unauthorized intrusions. Alternatively, the whole network might be utilized for this. A disc-based event data collection and storage device is in charge of collecting and storing data from various sources. The IDS's data processing and analysis unit is its brain. All the functions essential for detecting aberrant traffic patterns are included in this device. The detection of an assault results in the generation of a signal. The system may choose to act independently or send a signal to the network administrator informing them of an issue that needs to be fixed when an intrusion detection system (IDS) finds one. The actions of this software may cause an automatic intrusion response system or notify a network security administrator of potentially harmful activity. There are several methods to classify an IDS's modules. Two types of IDS can be distinguished based on the collection and archiving of data: Those IDSs that collect data from a host are called host-based IDSs. System calls, operating system logs (such as NT events and CPU utilization logs), and application logs are additional sources of information. Since buffer overflow attacks do not depend on the operating system, host-based IDS can detect them with ease. These techniques don't work with encrypted data or switched networks. If an intrusion detection system (IDS) gathers data from the network in the form of packets, it is referred to as network-based IDS. You may set up these IDS on nearly any system and they work with practically every platform.

B. Descriptions of CICIDS2017 dataset

As much as 98 percent+ accuracy and less than 1 percent false alarms are already claimed by researchers in the field of intrusion detection. Researchers and manufacturers were compelled to devote money and effort to the development of useful goods because of this higher rate of accuracy. In reality, only a few models have been recognized the industry to design an IDS. By analyzing temporary IDS models and training and testing datasets, the dataset not only includes the most recent network assaults, but it also meets all of the criteria for attacks that really occur in the real world. We noticed just a few flaws in this dataset when we investigated its properties. An obvious flaw is a large dataset, which was compiled from five days' worth of

Canadian Institute of Cybersecurity traffic data spread over eight files. An IDS might be designed from a single dataset. There are a lot of redundant entries in the dataset, making it unsuitable for training any IDS. Even if the dataset comprises contemporary assault scenarios, we also discovered that the dataset has a substantial class imbalance. Class imbalance datasets can mislead the classifier, biassing it towards the majority class. The research community was given a subset of the CICIDS2017 dataset to work with in developing and testing detection algorithms in an attempt to address these issues. Fig below shows the description of the CICIDS2017 dataset on which I have worked. we are able to identify the root cause of this issue. The Canadian Institute of Cybersecurity's CICIDS2017[7] collection provides the most up-to-date attack scenarios

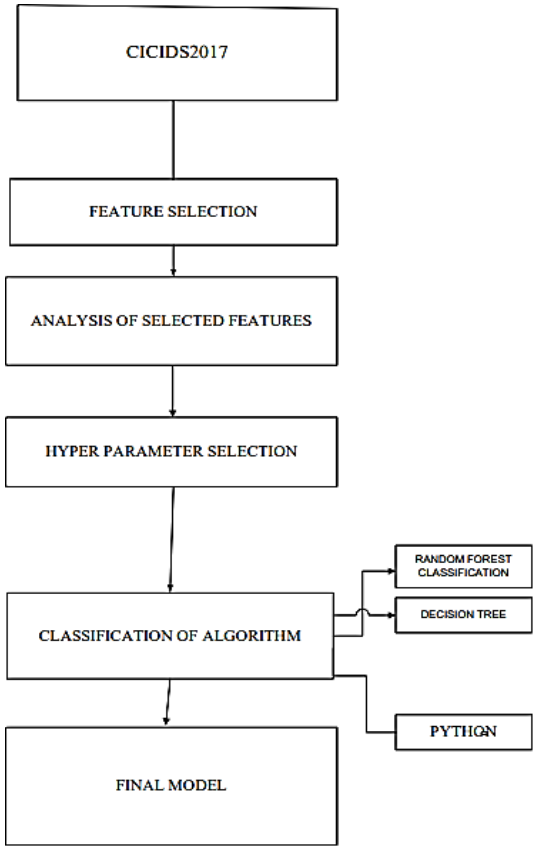


Figure 1: Description of the file containing the CICIDS2017 dataset

C. Shortcomings of the CICIDS2017

CICIDS2017 dataset has several flaws, as we previously noted, and the purpose of this work is to remedy those weaknesses so that future researchers may better understand them.

Scattered Presence- We can see in table 1 that there are now eight files containing the CICIDS2017 dataset's data. It's time-consuming to deal with individual files. As a result, we created a single file that had 3119345 instances of each of the files in question.

Table 2: Attack found in a day wise

Name of Files	Day Activity	Attacks Found
Monday-WorkingHours.pcap_ISCX.csv	Monday	Benign (Normal human activities)
Tuesday-WorkingHours.pcap_ISCX.csv	Tuesday	Benign, FTP-Patator, SSH-Patator
Wednesday-workingHours.pcap_ISCX.csv	Wednesday	Benign, DoS GoldenEye, DoS Hulk, DoS Slowhttptest, DoS slowloris, Heartbleed
Thursday-WorkingHours-Morning-WebAttacks.pcap_ISCX.csv	Thursday	Benign, Web Attack – Brute Force, Web Attack – Sql Injection, Web Attack – XSS
Thursday-WorkingHours-Afternoon-Infilteration.pcap_ISCX.csv	Thursday	Benign, Infiltration
Friday-WorkingHours-Morning.pcap_ISCX.csv	Friday	Benign, Bot
Friday-WorkingHours-Afternoon-PortScan.pcap_ISCX.csv	Friday	Benign, PortScan
Friday-WorkingHours-Afternoon-DDos.pcap_ISCX.csv	Friday	Benign, DDoS

D. Huge Volume of Data

All of the potential recent assault labels may be found in one location after integrating all of the data files. However, the combined dataset grows enormously in size. The sheer amount of information available becomes a problem in and of itself. It has a drawback in that it takes more time to load and analyse data.

E. Missing Values

The combined CICIDS2017 dataset has 288602 cases with a missing class label and 203 instances with missing metadata, as well. We found this to be a problem. To create a dataset with 2830540 unique occurrences, all but a few of the original data points were eliminated.

F. Classification of Algorithm

As per the results of this study, the most crucial aspects to examine during the selection process are the accuracy, scalability, speed, and learning capacity of a classifier algorithm. Five distinct classification algorithms—Random Forests, Bayesian Network, Random Trees, Naive Bayes, and J48 classifiers—have been used to demonstrate research and conclusions that back up this idea. Through the use of the Information Gain feature selection, this study shows that random forest trees may learn and perform magnificently in terms of assault detection. The Bayesian Network surpasses other algorithms when it comes to categorizing assaults. Random Tree is a scalable and efficient method. Since Naive Bayes has a low model complexity, it is a better choice for classifying data than other algorithms.

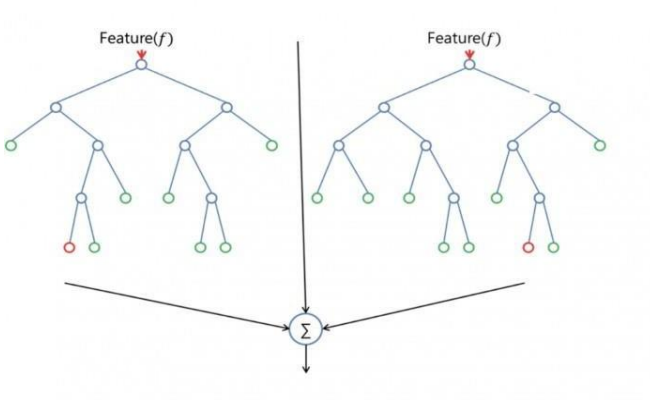


Figure 2: Random Forest classifiers tree.

1. Random Forest (RF)

Ensemble classifier approaches include Random Forest. A "forest" of classifiers is a decision tree classifier ensemble. To generate each decision tree, qualities are randomly selected at each node. In 2001, Breich introduced the random forest algorithm.

2. Bayes Network (BN)

Probabilistic connections between variables of interest are encoded using a Bayesian Network (BN) modeling technique. Assumptions about the model behavior of the target system are used to determine how accurate this technique is. If the assumption is significantly altered, then detection accuracy is reduced.

3. Random Tree (RT)

The term "random tree" refers to a decision tree that is built using a random set of attributes (random). There are many nodes and branches that can be linked together in a variety of ways in a decision tree. A node is used to represent an attribute being tested, while branches are used to indicate the findings. In the form of class albethey e, decision leaves display the final choice made after the computation of all attributes.

4. Naive Bayes (NB)

According to the Bayesian categorization system, the likelihood of belonging to a given class may be predicted statistically. Based on the Bayes theorem, we may classify data in a Bayesian fashion. Like the Nave Bayes classification, the Bayesian classification is better recognized by its more formal name. Ignoring other attribute values, Nave Bayes considers that attribute values have no effect on the class they belong to.

5. J48

Machine learning algorithm J48 or C4.5 is frequently used and is part of the decision tree algorithm. The entropy idea is used to form a decision tree in this technique.

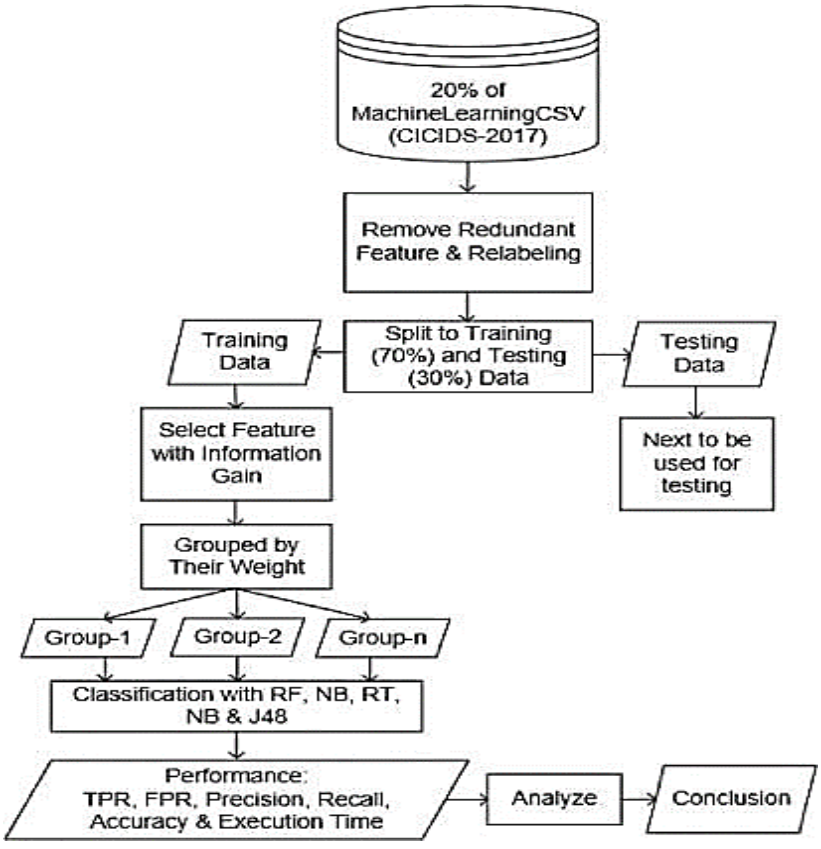


Figure 3: Experimental design

Random Forest (RF), Bayes Net (BN), Random Tree (RT), Naive Bayes (NB), and J48 classifiers are used to categorize each feature group or feature subset, respectively. The True Positive Rate, False Positive Rate, Precision Rate, Recall Rate, Accuracy Rate, Proportion of Incorrectly Categorized Data, and Analysis Execution Time are all taken into account when doing the analysis. Furthermore, there is a comparison between the True Positive Rate and the False Positive Rate. Furthermore, there is a comparison between the True Positive Rate and

the False Positive Rate. 10-fold cross-validation is the mechanism used at this point in the process. Analyzing and comparing the TPR, FPR, Precision and Recall, Accuracy, Percentage of Incorrect Categorization, and Execution Time of each classifier algorithm is an imperative requirement.at each and every level of the learning and testing process, a ten-fold cross-validation is conducted. It is essential that you arrive at some inferences or conclusions at this juncture.

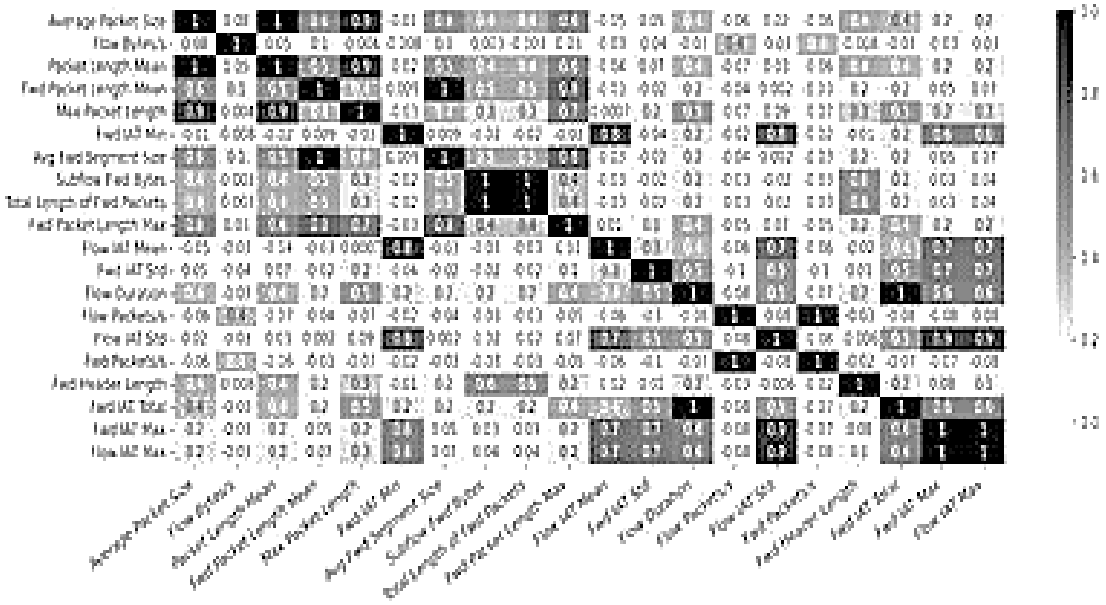


Figure 4: Accuracy graph

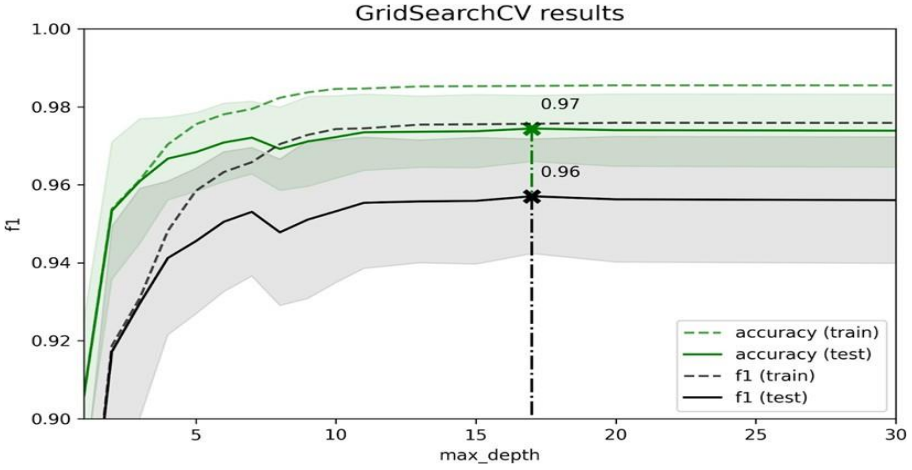


Figure 5: Correlated Heat Map

III. EXPERIMENTAL RESULT

In addition to the five distinct classifier approaches, the True Positive Rate (TPR), the False Positive Rate (FPR), the Precision, the Recall, the Accuracy, the Percentage of incorrectly Classified, and the Execution Time are the metrics that are utilised to evaluate the efficacy of Information Gain. These measurements make up what is known as the metric set when considered as a whole. The actual execution is simulated at a number of different periods in time throughout the course of the training, with the goal of further refining and improving upon it. During this experiment, the RT, BN, RT, NB, and J48 classifiers are combined in a wide variety of different ways to classify each unique feature subset. RT stands for the random tree; BN stands for the random tree; RT, NB, and J48 stand for the random tree; and RT. A 10-fold cross-validation method was used during the course of this investigation in order to ascertain whether or not categorization algorithms are successful. The 10-fold cross-validation is utilized since it cuts down on the overall amount of time spent calculating while also maintaining the reliability of the classification strategies. As a direct and immediate result of this, 10 random folds of the input dataset of the same size will be constructed from it. During the process of cross-validation, nine of the ten-fold data sets will be employed for training,

while just one of the ten-fold data sets will be utilized for testing. The ultimate outcome is a test fold, which is produced after ten repetitions of this procedure.

IV. OVERALL PROCESS

- Step-by-step procedure ()
- Feature Ranked data are accepted as input.
- The output includes the following: features subsets, TPR, FPR, accuracy, recall, and precision
- Decrease the number of features from 77 to n depending on the feature weight.
- For each and every function Fr in the data for Feature Ranked
- Begin to choose features using the Feature Weight, and then save them on Feature Groups

Group1 is comprised of any characteristic that has a weight more than or equal to 0.6 Any characteristics that have a weight that is more than or equal to 0.5 are included in Group2. Group3 is comprised of all characteristics with weights more than or equal to 0.4%. Every feature with a weight of at least 0.3 belongs to Group 4. Features with weights higher than or equal to 0.2 make up Group 5. Features with weights higher than or equal to 0.1 make up Group 6. The whole set of traits is represented by Group 7.

1) With respect to every grouping of features

2) Provide RF, BN, RT, NB, and J48 with specific characteristics using CICIDS-2017-20 percent

Detection	RF	BN	RT	NB	J48
Normal	0.960	0.943	0.960	0.174	0.961
DoS/ DDoS	0.992	0.996	0.992	0.999	0.991
Port Scan	0.995	0.992	0.995	0.983	0.995
Bot	0.438	0.642	0.430	0.687	0.381
Web Attack	0.072	0.031	0.072	0.000	0.072
Infiltration	0.000	0.000	0.400	0.400	0.000
Brute Force	0.792	0.991	0.792	1.000	0.790
Recall	0.965	0.962	0.970	0.903	NaN
Precision	NaN	0.953	0.965	0.335	0.965
FPR	0.016	0.010	0.016	0.026	0.016

Figure 6: Performance Metric Using Four Features

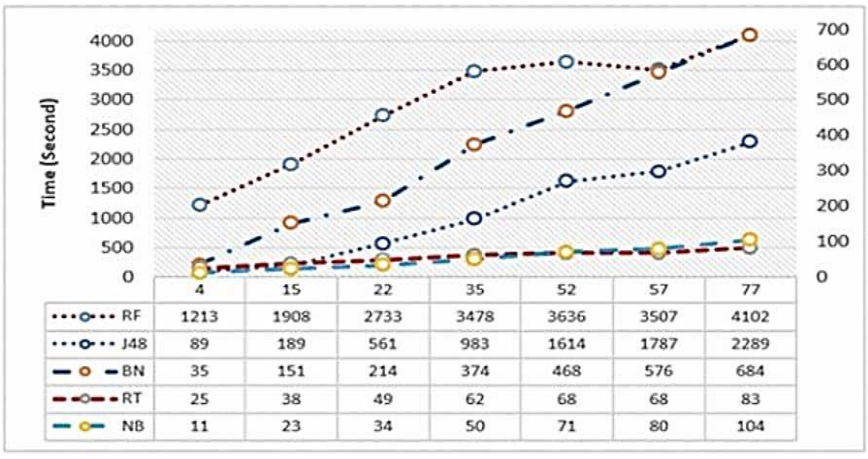
3) Use the Classifier The Random Forest model's accuracy is represented by C1. The accuracy of the Bayes Network model is equivalent to C2. The Random Tree model's accuracy is equal to C3. Accuracy of the Naïve Bayes model (C4) C5 = Correctness of J48 model

4) TPR and FPR computation accuracy, recall, and precision must be ascertained.

5) Compare and Examine the Accuracy of C1, C2, C3, C4, and C5.

The following list includes classifiers that make use of the four attributes that Information Gain selected. With an accuracy of 96.48 percent, only the RF and RT exceed the other classifiers. Conversely, RF has a value of NaN. "NaN" is an acronym that represents "Not a Number" or "undefined," in that order. Compared to other classifiers, NB has a higher TPR for detecting DoS/DDoS assaults, but a lower TPR for identifying infiltration and regular traffic. In contrast, out of all the firms under investigation, BN has the lowest FPR (0.010). Only DoS/DDoS, PortScan, and Brute Force attacks may be recognized by classifiers utilizing these four characteristics. In terms of normal traffic only NB is affected and using these four (4) characteristics. Only NB is affected by this in terms of normal traffic.

For the particular feature method under consideration, this study additionally examines the impact of execution time. An overview of the execution time for each feature subset employing RF, J48, BN RT and NB can be seen in the image below. There is a major influence on the RF, J48, and BN of the pertinent features procedure. RT and NB have extremely short run times. As a rule of thumb, the more characteristics to evaluate, the more time it takes to complete.



	4	15	22	35	52	57	77
RF	1213	1908	2733	3478	3636	3507	4102
J48	89	189	561	983	1614	1787	2289
BN	35	151	214	374	468	576	684
RT	25	38	49	62	68	68	83
NB	11	23	34	50	71	80	104

Figure 7: Execution time

V. CONCLUSIONS AND FUTURE WORK

Experiments were undertaken to demonstrate the impact of feature selection on improving anomaly detection accuracy. Feature sets 15, 22, and 35 were tested, and Information Gain was shown to be the top information classifier due to its accuracy in determining how much data is contained in each feature set. Feature sets 52, 57, and 77, on the other hand, are the best for J48. It is possible to detect all communication using feature sets 52, 57, and 77.5, even though BN's precision is lower than RF and J48, despite its lower level of accuracy. Experiments have also shown that the chosen traits reduce FPR, particularly for BN. Experimental results show that the number of features picked has an impact on the execution time of a program. Ranking characteristics according to weight values is what Information Gain proposes to do. It is, nevertheless, necessary for an expert to decide the minimum weight value, which influences the number of characteristics that will be picked. We intend to experiment with a variety of feature selection strategies in order to come up with the best possible mechanism. Each feature subset that impacts an assault will be analyzed as part of future research.

CONFLICTS OF INTEREST

The authors declare that they have no conflicts of interest

REFERENCES

1. Othman, Suad Mohammed, et al. "Intrusion detection model using machine learning algorithm on Big Data environment." Journal of Big Data 5.1 (2018): 34.
2. Salo, Fadi, Ali Bou Nassif, and Aleksander Essex. "Dimensionality reduction with IG-PCA and ensemble classifier for network intrusion detection." Computer Networks 148 (2019): 164-175.
3. Wagh, SharmilaKishor, Vinod K. Pachghare, and Satish R. Kolhe. "Survey on intrusion detection system using machine learning techniques." International Journal of Computer Applications 78.16 (2013). Chiba, Z., Abghour, N., Moussaid, K., El Omri, A., &Rida, M. (2019, June). An Efficient Network IDS for Cloud Environments Based on a combination of Deep Learning and an Optimized Self-adaptive Heuristic Search Algorithm. In International Conference on Networked Systems (pp. 235-249). Springer, Cham.
4. Idhammad, Mohamed, Karim Adel, and Mustapha Belouch. "Distributed intrusion detection system for cloud environments based on data mining techniques." Procedia Computer Science 127 (2018): 35-41.
5. Abdulhammed, Razan, et al. "Features Dimensionality Reduction Approaches for Machine Learning Based Network Intrusion Detection." Electronics 8.3 (2019): 322.
6. Basaveswara Rao B &Swathi K (2016) Variance-Index Based Feature Selection Algorithm for Network Intrusion Detection, IOSR Journal of Computer Engineering (IOSR-JCE) e-ISSN: 2278-0661,p-ISSN: 2278- 8727, Volume 18, Issue 4, Ver. V (Jul.-Aug. 2016), PP 01-11.
7. Basaveswara Rao B &Swathi, K. (2017). Fast kNN Classifiers for Network Intrusion Detection System. Indian Journal of Science and Technology, 10(14).
8. Swathi, Kailas am, and BobbaBasaveswara Rao. "Impact of PDS Based kNN Classifiers on Kyoto Dataset." International Journal of Rough Sets and Data Analysis (IJRSDA) 6.2 (2019): 61-72.
9. Ali, Mohammed Hasan, and Mohamad FadliZolkipli. "IntrusionDetection System Based on Fast Learning Network in Cloud Computing." Advanced Science Letters 24.10 (2018): 7360-7363.
10. Ahmed, H. A. S., Ali, M. H., Kadhum, L. M., Zolkipli, M. F., &Alsariera, Y. (2017). A review of challenges and security risks of cloud computing. Journal of Telecommunication, Electronic and Computer Engineering (JTEC), 9(1-2), 87-91.
11. Ali, Mohammed Hasan, et al. "A hybrid Particle swarm optimizationExtreme Learning Machine approach for Intrusion Detection System." 2018 IEEE Student Conference on Research and Development (SCOReD). IEEE, 2018.
12. Umer, Muhammad Fahad, Muhammad Sher, and Yaxin Bi. "Flow-based intrusion detection: Techniques and challenges." Computers & Security 70 (2017): 238-254.
13. "Nsl-kdd data set for network-based intrusion detection systems." Available on: http://nsl.cs.unb.ca/KDD/NSLKDD.html, March 2009.
14. Kyoto 2006+ dataset is available on: http://www.takakura.com/Kyoto_data/ [https://www.unb.ca/cic/datasets/ids-2017.html
15. Basaveswara Rao B, et al., "A Fast KNN Based Intrusion Detection System for Cloud Environment", Jour of Adv Research in Dynamical & Control Systems, Vol. 10, Issue 7, 2018 PP 1509 -1515.
16. Ring, Markus, Sarah Wunderlich, DenizScheuring, Dieter Landes, and Andreas Hotho. "A Survey of Network-based Intrusion Detection Data Sets." Computers & Security (2019). 18. Ahmim, A., Maglaras, L., Ferrag, M