iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 12 **Issue:** III **Month of publication:** March 2024

DOI: https://doi.org/10.22214/ijraset.2024.58920

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue III Mar 2024- Available at www.ijraset.com

Artificial Intelligence-Based Intrusion Detection System for Cloud Computing

Nishtha Singh
Abdul Kalam Technical University (AKTU), India

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. Threats from the internet could be identified using an intrusion detection system (IDS). The utilization of an intrusion detection system, or IDS, is necessary to preserve network security. 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, Machine learning Algorithm, Intrusion Detection System, Signature-Based IDS, Neural Networks.***

I. OUTLINE

Today, the Internet is an essential part of our everyday lives, and it is 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. Probing, Denial of Service, User to Root, and Remote User are the four categories into which assaults fall. 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

In recent years, the proliferation of networked systems and the increasing volume of data generated in these environments have raised significant concerns regarding network security. By keeping an eye on and evaluating network activity to identify and react to possible breaches, intrusion detection systems, or IDS, are essential to the protection of these systems. Traditional IDS approaches are being complemented and enhanced by the integration of machine learning (ML) techniques, taking advantage of their capacity to analyze massive amounts of data and spot intricate patterns. This literature review synthesizes key studies in the field of IDS, focusing on the application of ML techniques for improved detection accuracy and efficiency.

Othman et al. (2018) introduced an IDS model utilizing machine learning algorithms in a Big Data environment. The research underscores the importance of utilizing Big Data analytics to tackle the difficulties presented by the abundance, variety, and speed of network data. By employing the Spark-Chi-SVM architecture, the authors developed a scalable IDS capable of effectively managing massive datasets, thereby enhancing the accuracy of intrusion detection.

Salo et al. (2019) suggested IG-PCA, a dimensionality reduction method, in conjunction with ensemble classifiers for network intrusion assessment. Their approach aimed to address the complexity associated with high-dimensional data by reducing feature space while preserving relevant information. By integrating IG-PCA with ensemble classifiers, the study achieved improved detection performance, demonstrating the efficacy of feature selection techniques in enhancing IDS accuracy.

©IJRASET: All Rights are Reserved

Wagh et al. (2013) highlighted the increasing use of ML-based intrusion detection systems in a survey on the topic of machine learning-based intrusion detection systems. The study underscores the importance of network security in the era of pervasive computing and highlights the role of ML in enhancing IDS capabilities, particularly in distinguishing between normal and abnormal network behavior.

Idhammad et al. (2018) proposed a cloud-based distributed IDS leveraging data mining techniques for enhanced intrusion detection. By preprocessing network data using a time-based sliding window approach and employing Random Forest classifiers, the study achieved effective anomaly detection in cloud environments. Their approach underscores the importance of distributed architectures and scalable algorithms in addressing the evolving threat landscape.

Abdulhammed et al. (2019) investigated methods to reduce the dimensionality of features for machine learning-based network intrusion detection. The study attempted to decrease the dimensionality of feature space while maintaining discriminatory information by utilizing methods like Principal Component Analysis (PCA) and Auto-Encoder (AE). Their findings suggest that dimensionality reduction techniques enhance classifier performance, thereby improving IDS accuracy and efficiency.

Basaveswara Rao & Swathi (2016, 2017, 2019) proposed various approaches for network intrusion detection, including variance-index-based feature selection algorithms and fast kNN classifiers. Their studies emphasize the importance of feature selection and classifier optimization in developing efficient IDS solutions tailored to specific deployment scenarios.

Ali et al. (2018, 2020) investigated hybrid approaches combining optimization algorithms and ML techniques for intrusion detection in cloud computing environments. By integrating Particle Swarm Optimization (PSO) with Extreme Learning Machines (ELM), the studies aimed to enhance IDS performance in dynamic and resource-constrained cloud environments.

Umer et al. (2017) examined methods for flow-based intrusion detection, stressing the difficulties and developments in identifying abnormalities in networks. The study underscores the importance of flow-based analysis in capturing fine-grained network behavior and identifies future research directions for improving IDS effectiveness.

III. INTRUSION DETECTION SYSTEM

Effective security systems with the ability to recognize, stop, and perhaps react to cyberattacks are essential parts of any security architecture. In order to monitor specific sources of activity, security services employ a range of techniques, such as auditing and network traffic data in computer or network systems. All threats must be swiftly and correctly identified by a intrusion detection system (IDS). IDS may help network managers find security flaws objectively. Attempts to gain unauthorized access to the network may come from outside intruders. It is a violation of security objectives to compromise infrastructure security or to render resources unusable for insiders who misuse their system resources.

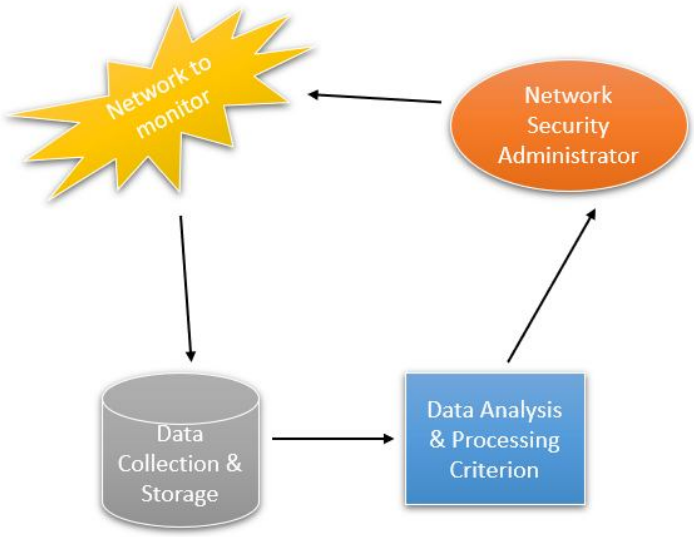


Fig.1. Architecture of Intrusion Detection System

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 in order 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. A signal is generated when an attack is detected. When an intrusion detection system (IDS) detects a vulnerability, the system can decide whether to handle it on its own or notify the network administrator of the problem. A network security administrator may receive a notification of malicious activity or an automatic response to an intrusion as a result of this program's actions. There are numerous 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. Host-based systems can quickly identify buffer overflow attacks.

IDS as they are independent of the operating system. 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.

IV. DESCRIPTIONS OF CICIDS2017 DATASET

Researchers in the field of intrusion detection have already reported accuracy rates as high as 98 percent or higher and false alarm rates as low as 1 percent. This high accuracy rate pushed researchers and manufacturers to invest time and resources in creating valuable products. 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 CICIDS2017[7] collection provides the most up-to-date attack scenarios.

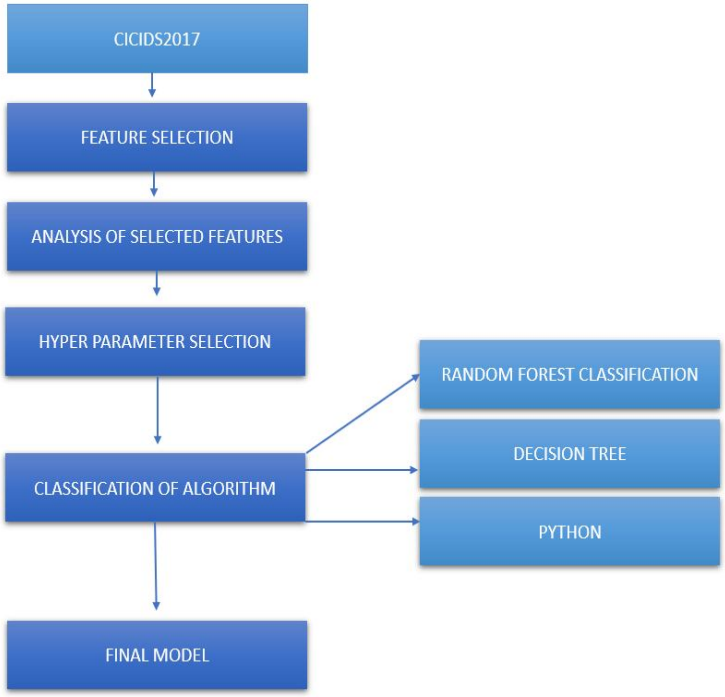


Fig.2. Description of CCIDS2017 Dataset

V. 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.1.
DESCRIPTION OF THE FILE CONTAINING THE CICIDS2017 DATASET

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, Infilteration
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

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 analyze data.

Missing Values - In addition to 203 cases lacking metadata, the combined CICIDS2017 dataset contains 288602 cases lacking a class designation. 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.

VI. CLASSIFICATION OF ALGORITHM

The results of this study indicate that the most important things to watch out for when choosing a classifier algorithm are its accuracy, learning capacity, scalability, and speed. Eleven distinct classification algorithms—Random Forests, Bayesian Network, Random Trees, Naive Bayes, and J48 classifiers—have been used in research and discoveries to demonstrate the viability of this idea. By applying the Information Gain feature selection, this study shows that random forest trees can learn and perform quite well in terms of attack 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.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

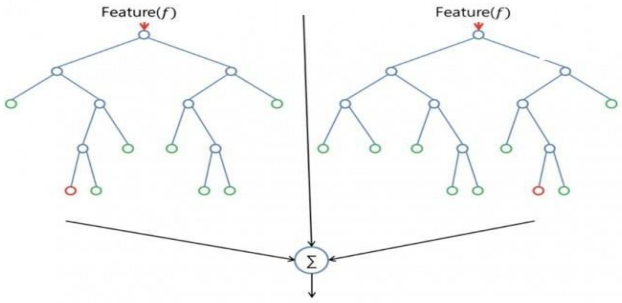


Fig.3. Random Forest classifiers tree

A. 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.

B. Bayes Network (BN)

Probabilistic connections between variables of interest are encoded using a Bayesian Network (BN) modeling technique. The accuracy of this technique is based on assumptions about the target system's model behavior. The accuracy of detection decreases if the assumption is substantially changed.

C. RT (Random Tree)

Any decision tree constructed with a random collection of characteristics is referred to as a "random tree". A decision tree has numerous branches and nodes that can be connected in a number of ways. 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.

D. Naive Bayes (NB)

The probability of falling into a certain class can be statistically predicted, according to the Bayesian classification approach. 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.

E. J48

As a component of the decision tree algorithm, the machine learning algorithm J48 or C4.5 is often utilized. This technique creates a decision tree based on the concept of entropy.

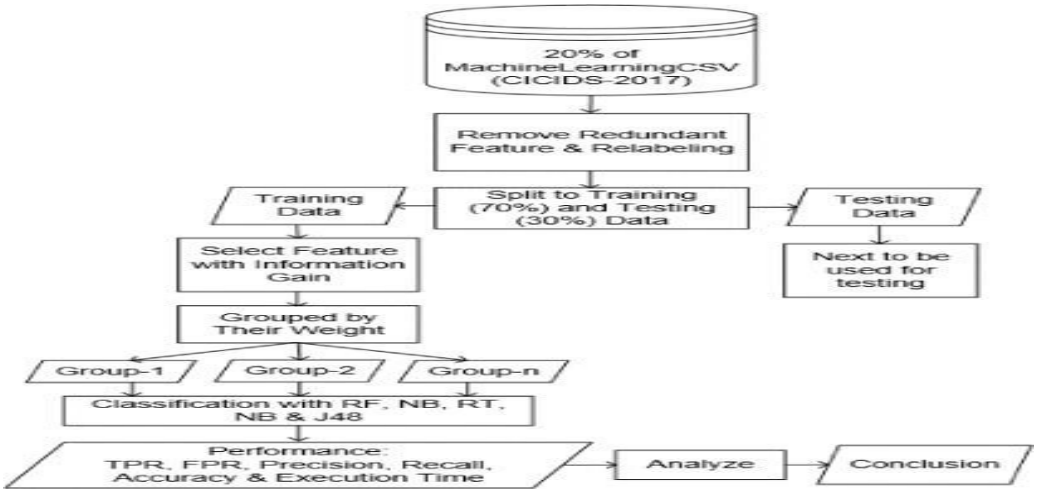


Fig.4. Experimental Design

- The J48, Random Forest (RF), Random Tree (RT), Bayes Net (BN), Naive Bayes (NB), and Random Tree (RT) classifiers are used to categorize each feature group or feature subset. The following factors are taken into account in order to do the analysis: the fraction of incorrectly classified data, the precision, recall, accuracy, accuracy rate, false positive rate, and execution time of the analysis. The True Positive Rate and False Positive Rate are also contrasted. Additionally, there is a comparison between the True Positive Rate and the False Positive Rate. A technique called 10-fold cross-validation is used at this point in the procedure.
- It is essential to analyze and compare each classifier algorithm's TPR, FPR, Accuracy, Precision and Recall, Percentage of Incorrect Categorization, and Execution Time. 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.

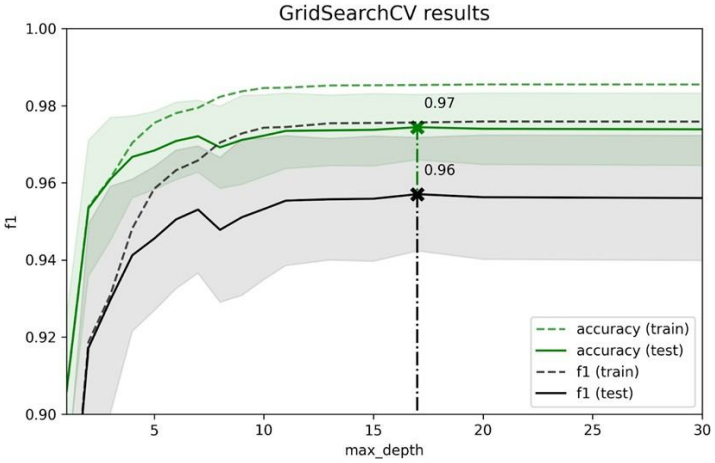


Fig.5. Accuracy Graph

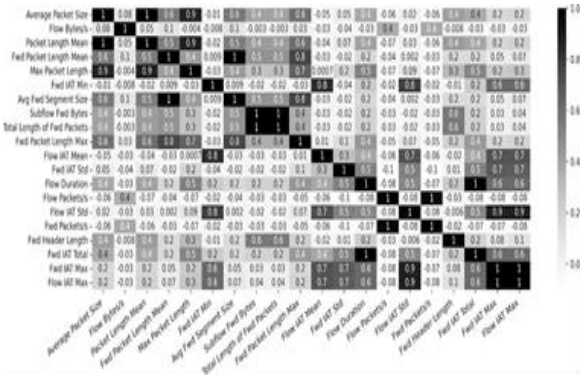


Fig.6. Correlated Heat Map

VII. EXPERIMENTAL RESULT

To assess the effectiveness of Information Gain, metrics such as True Positive Rate (TPR), False Positive Rate (FPR), Precision, Recall, Accuracy, Percentage of incorrectly Classified, and Execution Time are employed alongside the five distinct classifier approaches. 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. Each distinct feature subset is classified in this experiment by combining the RT, BN, RT, NB, and J48 classifiers in a multitude of ways. There are four different ways to represent the random tree: RT, NB, J48, and RT. The effectiveness of categorization algorithms was determined throughout this inquiry using a 10-fold cross-validation method. 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.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

VIII. OVERALL PROCESS

1) Step-by-step procedure ()
2) Feature Ranked data are accepted as input.
3) Features subsets, TPR, FPR, accuracy, recall, and precision are all included in the result.
4) Reduce the feature count from 77 to n, based on the weight of each feature.
5) For every function Fr in the feature-ranked data
6) 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. Group4 is comprised of any feature that has a weight more than or equal to 0.3. All features with a weight of at least 0.2 are included in Group 5. Features with weights greater than or equal to 0.1 make up Group 6. The entire set of traits is represented by Group 7.

Regarding each of the Feature groupings

Give specific features to RF, BN, RT, NB, and J48 using CICIDS-2017-20 percent.

Apply Classifier Accuracy of the Random Forest model, denoted as C1 C2 equals the accuracy of the Bayes Network model C3 equals the accuracy of the Random Tree model C4 = Naïve Bayes model accuracy C5 = J48 model correctness.

The TPR and FPR computations' accuracy, recall, and precision must be ascertained.

Analyze, compare, and assess the accuracy of C1, C2, C3, C4, and C5.

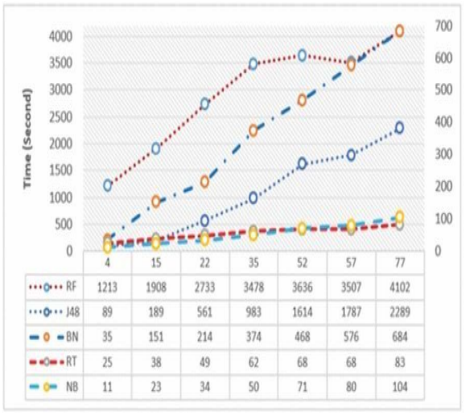
The following list includes classifiers that make use of the four attributes that Information Gain selected. With an accuracy of 96.48 percent, only the RF and RT exceed the other classifiers. Conversely, RF has a value of NaN. "NaN" is an acronym that represents "Not a Number" or "undefined," in that order. Compared to other classifiers, NB has a higher TPR for detecting DoS/DDoS attacks, but a lower TPR for identifying infiltration and regular traffic. In contrast, out of all the companies under investigation, BN has the lowest FPR (0.010). These four (4) features are the only ones that classifiers may use to identify DoS/DDoS, Brute Force, and PortScan attacks. In terms of regular traffic, this just affects NB.

TABLE.2.
PERFORMANCE METRIC USING FOUR FEATURES

DETECTION	RF	BN	RT	NB	J48
Normal	0.960	0.943	0.960	0.174	0.961
DDoS/DoS	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	0.004	0.004	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

For the particular feature method under consideration, this study additionally examines the impact of execution time. The picture below shows a summary of the execution times for each feature subset using RF, J48, BN RT, and NB. A significant impact is seen on the pertinent characteristics procedure's RF, J48, and BN. The run times of RT and NB are incredibly short. As a rule of thumb, the more characteristics to evaluate, the more time it takes to complete.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



	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

Fig.7. Execution Time

IX. CONCLUSIONS & FUTURE WORK

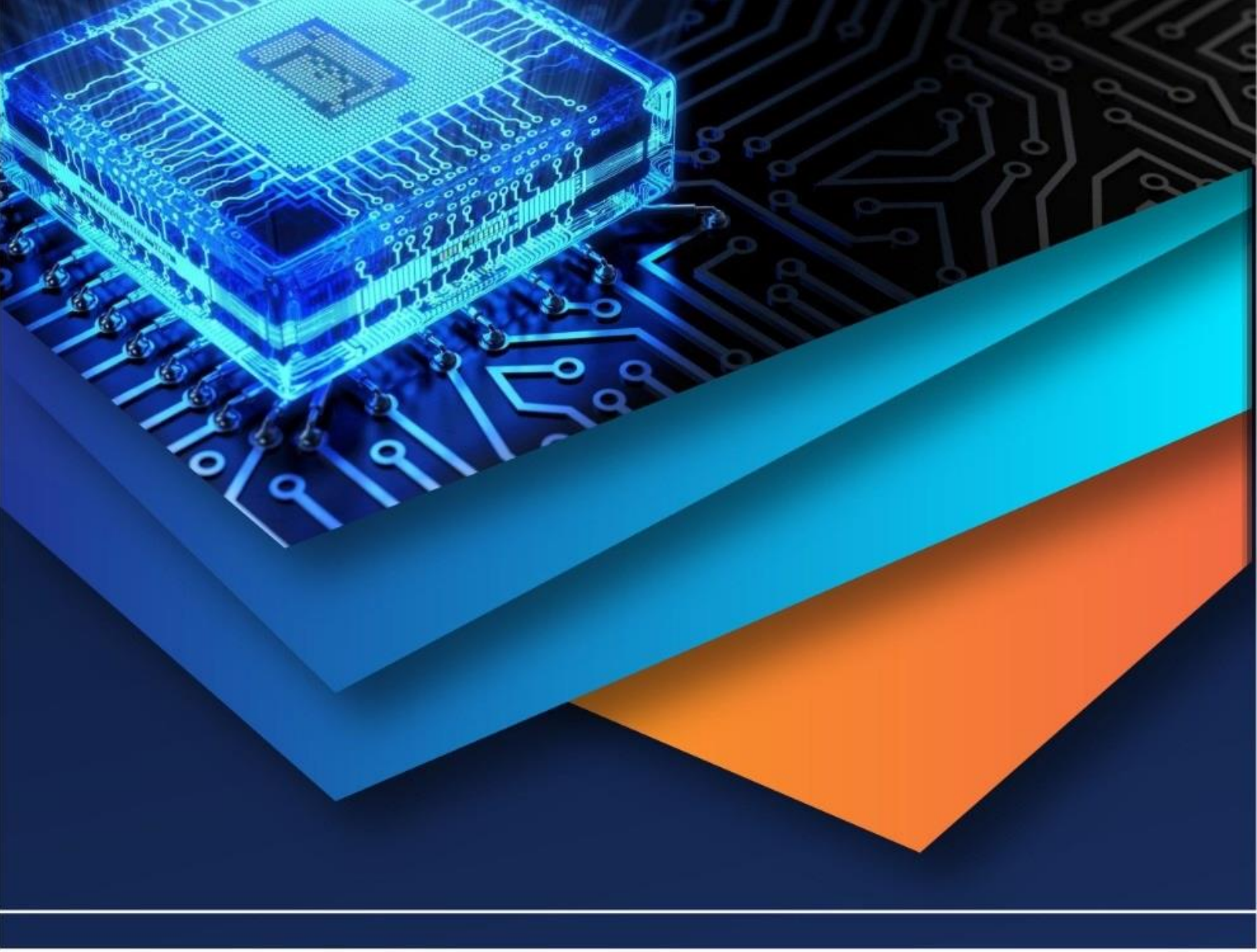
Trials were conducted to show how feature selection affects increasing the accuracy of anomaly detection. Feature sets 15, 22, and 35 were tested, and Information Gain was shown to be the top information classifier due to its accuracy in determining how much data is contained in each feature set. Feature 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. The results of the experiments indicate that a program's execution duration depends on the number of features selected. Ranking characteristics according to weight values is what Information Gain proposes to do. However, the minimal weight value determines how many qualities will be chosen, thus an expert must make this decision. 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.

REFERENCES

[1] Halma, J., & Bauer, S. (1995). An Overview of Intrusion Detection Systems. Proceedings of the 1995 IEEE Symposium on Security and Privacy, 68-76.
[2] Othman, Suad Mohammed, et al. "Intrusion detection model using machine learning algorithm on Big Data environment." Journal of Big Data 5.1 (2018): 34.
[3] 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.
[4] 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.
[5] 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.
[6] Abdulhammed, Razan, et al. "Features Dimensionality Reduction Approaches for Machine Learning Based Network Intrusion Detection." Electronics 8.3 (2019): 322.
[7] 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.
[8] Basaveswara Rao B &Swathi, K. (2017). Fast kNN Classifiers for Network Intrusion Detection System. Indian Journal of Science and Technology, 10(14).
[9] 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.
[10] Ali, Mohammed Hasan, and Mohamad FadliZolkipli. "IntrusionDetection System Based on Fast Learning Network in Cloud Computing." Advanced Science Letters 24.10 (2018): 7360-7363.
[11] Ahmed, H. A. S., Ali, M. H., Kadhum, L. M., Zolkipli, M. F., &Alsariera, Y. A. (2017). A review of challenges and security risks of cloud computing. Journal of Telecommunication, Electronic and Computer Engineering (JTEC), 9(1-2), 87-91.
[12] 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.
[13] Umer, Muhammad Fahad, Muhammad Sher, and Yaxin Bi. "Flow-based intrusion detection: Techniques and challenges." Computers & Security 70 (2017): 238-254.
[14] "Nsl-kdd data set for network-based intrusion detection systems." Available on: http://nsl.cs.unb.ca/KDD/NSLKDD.html, March 2009.
[15] Kyoto 2006+ dataset is available on: http://www.takakura.com/Kyoto_data/
[16] https://www.unb.ca/cic/datasets/ids-2017.html
[17] 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.

[18] 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
[19] Axelson, B. (1999). Intrusion Detection Systems: Common Design and Architecture.
[20] Canadian Institute of Cybersecurity. (2017). CICIDS2017 Dataset.
[21] Breiman, L. (2001). Random Forests. Machine Learning, 45(1), 5-32. DOI: 10.1023/A:1010933404324
[22] Friedman, N., & Koller, D. (2003). Being Bayesian About Network Structure. AISTATS, 1, 201-208.
[23] Breiman, L., Friedman, J., Stone, C. J., & Olshen, R. A. (1984). Classification and Regression Trees. CRC Press.
[24] Mitchell, T. M. (1997). Machine Learning. McGraw-Hill.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)