iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 10 **Issue:** V **Month of publication:** May 2022

DOI: https://doi.org/10.22214/ijraset.2022.43434

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 10 Issue V May 2022- Available at www.ijraset.com

Building an Efficient Intrusion Detection System using Feature Selection and Machine Learning

Akshay Kaushik[1], Varun Goel[2]
[1]*Department of Information Technology, Maharaja Agrasen Institute of Technology, (Affiliated to GGSIPU), New Delhi, India*
[2]*Assistant Professor, Department of Information Technology, Maharaja Agrasen Institute of Technology, (Affiliated to GGSIPU) New Delhi, India*

Abstract*: With the increase in internet activity and as the world goes digital as the days go, the risk of exposure to malicious activities also increased rapidly. The intruders/hackers use various methods to gain unauthorized access to one's computer or any other device, Network Intrusion is one of the methods by which intruders attack the network of the user, the user can be an individual or an organization based on the intention/agenda of the attackers. Significant Reasons for intrusion are Hacktivism, Steal Money or Data, and Spying. Due to the internet being a vast place, it is challenging to pinpoint a particular way in which Network Intrusion takes place, therefore a Network Intrusion Detection System needs to be put in place in order to deal with the issues regarding Network Intrusions. There are multiple leaks or data extortion that happened previously and, in this paper, the dataset released based on a leak from KDD99 is used. An improved version of KDD99 (NSL-KDD) is used in this study. NSL-KDD datasets have been used for training the Machine Learning Model. Given the number of attributes in the dataset, it was difficult to use all attributes so, feature selection methods were used to get the best attributes to develop an efficient Machine Learning model. In this analysis of Machine Learning algorithms, the algorithms under consideration are Logistic Regression, Support Vector Machine, Decision Tree, Random Forest, and Naive-Bayes. For comparison of the performance of the algorithms metrics like Accuracy Score, Confusion Matrix, and Classification Report were considered to find the best algorithm among them.

Keywords: Machine Learning, Feature Selection, Intrusion Detection, Algorithm, Dataset, NSL-KDD, Attacks,

I. INTRODUCTION

With the increase in the use of the Internet, development in technology, and increasing number of data leak incidents, Network Security has become a vital topic of research. With the availability of data to a larger range of audiences, privacy and integrity of data have to be provided. The intrusion detection system is a tool for detection of abnormal behaviors in a system. An abnormal pattern in general is, unwanted, malicious and misuse activity occurring within the system. The intrusion detection system can be an implementation of software or a hardware that is used to monitor the networks for intrusions or any deviation from the normal activity. Normally, intrusion detection system is a security surveillance system, such as a firewall system that tries to find and if possible, safeguard and prevent the system from harm. The basic functioning of the intrusion detection system is, to behave as a passive alert system, that is, if the intrusion is found on the network IDS produces an alarm and gives the user the relevant information (IP, ports, packets, etc.) which initiated the alarm. The network intrusion or an attack majorly can be classified into four classes:

1) *Denial of Service Attacks (DoS) Attack:* In this attack class, an attacker prevents the legitimate users from using services on the network, by overwhelming or flooding the system with request and consuming the resources by exploiting system's misconfigurations or by aiming the implementations bugs. DoS attacks can be classified on the basis of the services that an attacker exploit. Types of DoS attacks are: Smurf, Neptune, back, teardrop, pod and land.
2) *Probe Attack:* In this class, an attacker scans a network or host to gather known vulnerabilities and information about the host computer. An attacker with the map of machines and services offered to them on the network, uses the information to search for exploits. Probe attack abuses the computer's legitimate features, social engineering techniques. It is the most common class of attacks and demands less technical skills and expertise, e.g., Probe attacks are: ipsweep, Nmap, portsweep and satan.
3) *U2R Attacks:* In this class of attacks, an attacker first hacks into a normal/legitimate user's account on the network and then exploits the vulnerabilities so that he can gain root access of the system. Regular buffer overflows are the most usual exploit in this attack class, and the reasons may be programming mistakes or environment assumptions. Types of Users to Root attacks are: buffer_overflow, rootkit, loadmodule, perl.

4) *R2L Attacks:* It is a class of attacks where an attacker sends packets to a machine on the network, then exploits its vulnerabilities and illegally gains local access as a user. E.g., of Remote to local attacks are: ftp_write, spy, imap, warezclint, guess_passwd, warezmaster, ftp_write, multihop, phf.

Due to the internet being a vast place, it is very difficult to pinpoint a particular way in which Network Intrusion takes place. However, the following are some common techniques through which Network Intrusion has taken place:

Multi-Routing- This refers to when the intruders use multiple sources to intrude which helps them avoid detection. This is also known as asymmetric routing;

Buffer Overflow Attacks- The Buffer overflow attack refers to when certain sections of the computer's memory code are rewritten so that they can be used as a part of the intrusion later on;

Traffic Flooding- This type of attacks is when the intruders flood the victim's systems with traffic that they cannot handle in order to cause chaos and confusion. When the systems have too large traffic in order to screen, then they can easily get away undetected;

Trojan Horse Malware- Trojan Horse Malware gives provides a network backdoor to the attackers so that they get an unfettered access to the network;

Worms- This type of virus is most common and effective. Worms usually spread through email or instant messaging and can spread throughout the network.

To minimize and ultimately stop these attacks, we need to know about them as soon as intruders malicious activity hit the network so, that the defense mechanism can be activated and loss of data or anything confidential can be revealed. For that we need an Intrusion Detection System that alerts the owner of the attack. An intrusion detection system (IDS) goes through the activity on the network to find possible intrusions. Intrusion Detection is possible when we have the model to predict the possibility of an attack, for the model should be trained on data of previous attacks. In this study, the NSL-KDD dataset has been used for training different Machine Learning models.

NSL-KDD dataset is the improved version of the KDD99 dataset, from which duplicate values were removed to get rid of biased results of classification.

Machine Learning Algorithms that are considered for this study are Logistic Regression, Support Vector Machine, Decision Tree, Random Forest and Naïve-Bayes. In this study, research is more intended to find the best attributes among all attributes available in the NSL-KDD dataset and also find the best algorithm among above mentioned algorithms to classify the attack. The selection of the features or attributes is the most pivotal part of this study and building the model with the best Machine Learning algorithm to predict the attack. The study gives an idea about which features and machine learning algorithm should be used by the Intrusion Detection System that will best identify the deviation in the network traffic.

The paper is organized as follows:

- Section I gives the introduction about the NIDS and its need.
- Section II named as Literature Review discussed work of other authors on Intrusion Detection System, Feature Selection Methods, Machine Learning Algorithms and Building an Efficient IDS using Feature Selection and Machine Learning Techniques.
- Section III describes dataset used in this study and methodology followed to get desired results.
- Section IV has the discussion about different Metrics to measure the performance of the model.
- Section V explains the results of the implementation of classifiers and shows the results using metrics discussed in Section IV.
- Section VI discussed the conclusion of this study and what future work can be done in this paper to get better results.

II. LITERATURE REVIEW

Network Security is one of the vital research topics, several other authors have worked on this topic and found different insights.

Most of the studies on the Intrusion Detection System use the KDD99 dataset, The KDD99 dataset consists of 41 features obtained by preprocessing from the DARPA dataset in 1999. It consists of almost 5 M and 2 M instances for training and testing respectively.

The author[1] studied the effectiveness of the dataset, reviewed the datasets and performance evaluation methods on these datasets.

Author[2] have also utilized the NSL-KDD dataset and studied a new model that can be used to estimate the intrusion scope threshold degree based on the network transaction data's optimal features that were made available for training

Author[3] have featured a combined 2 data mining algorithms Decision Tree and SVM in their paper and the main target was to combine the advantages of both the algorithms.

Author[4] had an experiment aiming to understand the implications of using supervised machine learning techniques on intrusion detection and results showed that Random Forest Classifier worked best for that dataset. Similar Studies have been done by many other researchers also.

Adetunmbi A.Olusola., Adeola S.Oladele and Daramola O. Abosede [5] have developed signature-based IDS using neural network with the back propagation training algorithm. It was used to determine and predict current and possibly future attacks. For training & testing of classifier KDD Cup (1999) dataset was used.

Mockamole and Sung [7] selected 6 features from 41 using a novel feature selection algorithm and evaluated using SVM model. So selected features improve the classification accuracy by 1%.

III. RESEARCH METHODOLOGY

A. Dataset

The NSL-KDD dataset is collected, NSL-KDD is the improved version of KDD99. The NSL-KDD dataset has various version available on the internet.

The version we have used have number of instances in the training dataset: 125973, number of instances in the test dataset: 22544

This dataset has the following advantages:

1) It does not consist of recurring instances in the train set, which makes the classifier less biased towards some attacks.

2) There are no null values available in the dataset

3) It does not consist not necessary instances in the training set, so the classifiers will not be partial towards more duplicate records.

The dataset has 41(excluding Label) attributes including label which is the target/ dependent variable. Out of 42, 22 are integer, 4 are object-type and 15 float-type attributes.

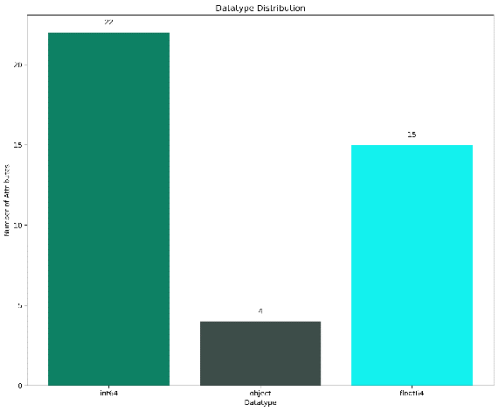


Figure 1: Datatype Distribution

The training dataset have 16 unique attacks in Label attribute while test dataset has 33 unique attacks in the same.

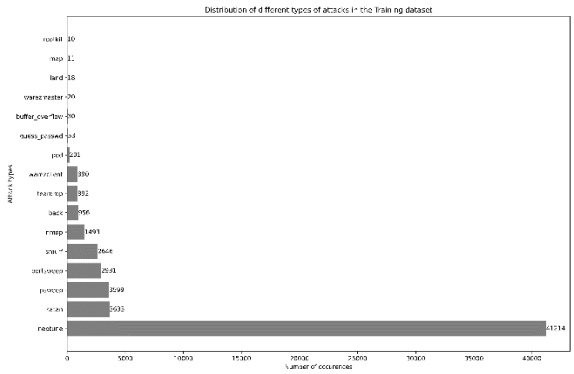


Figure 2:Distribution of Attacks in Training Dataset

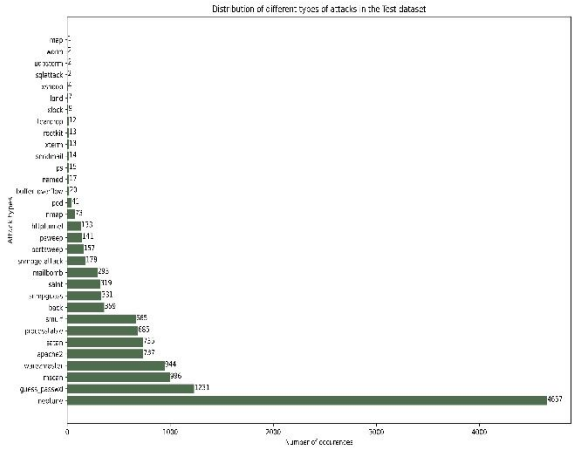


Figure 3:Distribution of Attacks in Test Dataset

B. Methodology

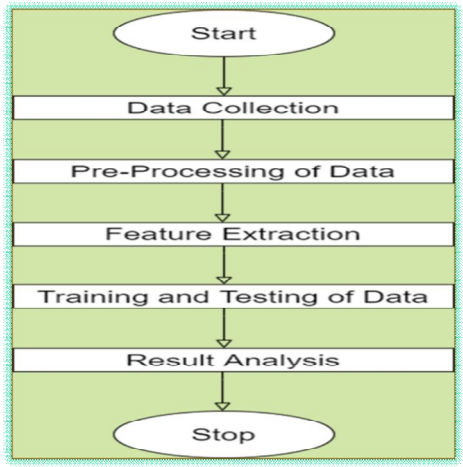


Figure 4 : Flow Chart of the process followed

The process started with the collection of the dataset, after collection pre-processing on the dataset is performed in which data is checked for null values, missing values, out of domain values. There were none of the above anomalies in the dataset. The distribution of different types of attacks was checked and found that attacks like a spy, Perl, phf, multihop, ftp_write, loadmodule, have instances less than 10, so were moved these since there will not be sufficient training data for the Machine Learning Model.

In the dataset, there were three data type attributes int(22 attributes), float(15 attributes), and object(3 attributes). Int, a float is ready for training the model while object (categorical values) type attributes needed encoding to numerical values so it could be used for the training of the model also. For encoding, Label Encoder are used in this study. After conversion, the newly created attributes are concatenated with the rest of the columns.

Now dataset was free from anomalies and the attributes were either integer or float, ready to be normalized for training the models. Min-Max and Standard Scaler were used to normalize the data in two different instances and models were trained; it was found that the dataset normalized with Standard Scaler produced output with more accuracy score.

$$Standard\ \ Scaler\ (z) = \frac{(x-u)}{s}$$

x=current value
u=mean value
s=standard deviation

As our dataset is normalized.

Now the label attribute in train dataset has imbalance class distribution which needed to be handled before moving to modeling of the Machine Learning Algorithms.

For this SMOTE(Synthetic Minority Oversampling Technique) is used to make all the labels equal so that the model does not have majority bias and produces result equally for each individual class.

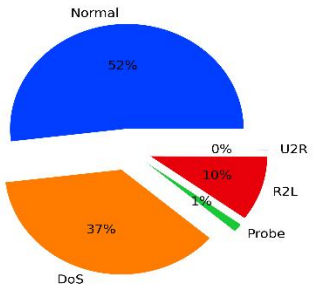


Figure 5: Class Distribution before SMOTE

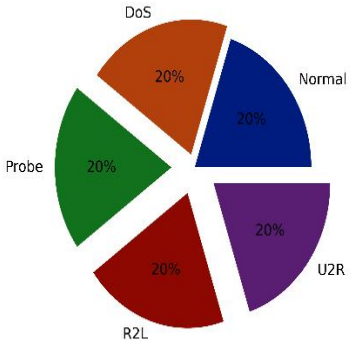


Figure 6: Class Distribution after SMOTE

After SMOTE and Standardization of the dataset, it was ready for training, the Implementation of the Machine Learning algorithms mentioned above is done using python library scikit-learn.

1) Logistic Regression

```
clf = LogisticRegression()
clf.fit(X_train, Y_train)
pred = clf.predict(X_test)
```

Figure 7: Implementation of Logistic Regression

2) *Support Vector Machine(SVM):*

```
clf1 = LinearSVC(random_state=0)
clf1.fit(X_train, Y_train)
predsvm = clf1.predict(X_test)
```

Figure 8:Implementation of Support Vector Machine

3) Decision Tree Classifier

```
clf2=DecisionTreeClassifier(random_stat
e=0)
clf2.fit(X_train, Y_train)
predDT = clf2.predict(X_test)
```

Figure 9: Implementation of Decision Tree

4) Random Forest Classifier

```
clf3 =
RandomForestClassifier(random_state=40,
n_estimators=300,n_jobs=-1)
clf3.fit(X_train, Y_train)
predRF = clf3.predict(X_test)
```

Figure 10: Implementation of Random Forest

5) Naive-Bayes

```
BNB_Classifier = BernoulliNB()
BNB_Classifier.fit(X_train, Y_train)
predNB = BNB_Classifier.predict(X_test)
```

Figure 11: Implementation of Naïve-Bayes

After implementation, the performance of all the classifiers was measured using various metrics like accuracy score, confusion matrix, and classification report.

IV. METRICS AND PERFORMANCE EVALUATION

Before moving to the measure of the performance of model we need to know a few terms, terms are:

1) *True Positive(tp):* True Positive means when model predicted the instance positive and it was positive in y_true also.
2) *True Negative(tn):* True negative is when model correctly predicts the negative class of the dataset.
3) *False Positive(fp):* False positive is when model incorrectly predicts the positive class.
4) *False Negative(fn):* False negative is when model incorrectly predicts the negative class.

Confusion Matrix is a table that is used to measure the performance of the model(classification model)

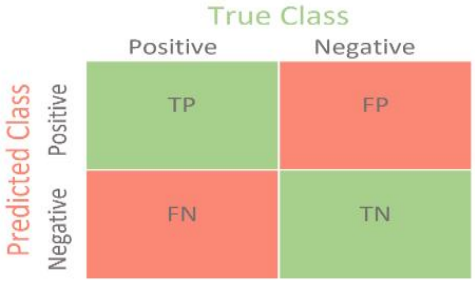


Figure 12: Confusion Matrix

After implementation, the performance of the model was measured using the following metrics:

a) Accuracy Score: It is defined as the ratio of the sum of true positive and true negative to all predictions. The Calculation formula is given below:

$$Accuracy = \frac{(tp+tn)}{tp+tn+fp+fn}$$

b) Precision: Precision is the ratio of true positive to the sum of true positive and false positive. Formula is given below:

$$Precision = \frac{tp}{tp+fp}$$

c) Recall: Recall is the ratio of true positive to the sum of true positive and false negative. Formula is given below:

$$Recall = \frac{tp}{tp+fn}$$

d) Support: The support is the number of occurrences of each class in y_true.

Classification Report:
The classification report shows the values of precision, recall, F1-score, and support scores of the classifier.

V. RESULTS

After implementation and performance measurement of the four classifiers used in this paper, their results are as follows:

A. Logistic Regression

```
Accuracy of Logistic Regression is 0.8076.
Confusion Matrix is:
 [[8101  401  279  508  422]
 [ 616 6203  158  473   10]
 [  65   77 2255   24    0]
 [ 970    3   16 1573  300]
 [   5    2    0    4   52]]
Classification Report is:
              precision    recall  f1-score   support

           0       0.83      0.83      0.83      9711
           1       0.93      0.83      0.88      7460
           2       0.83      0.93      0.88      2421
           3       0.61      0.55      0.58      2862
           4       0.07      0.83      0.12        63

    accuracy                           0.81     22517
   macro avg       0.65      0.79      0.66     22517
weighted avg       0.83      0.81      0.82     22517
```

Figure 13: Result of Logistic Regression

B. Support Vector Machine

```
Accuracy of SVM is 0.82.
Confusion Matrix is:
 [[8427   63  314  488  419]
 [ 708 6094  192  458    8]
 [  39   96 2252   24   10]
 [ 873    1   26 1638  324]
 [   3    3    0    4   53]]
Classification Report is:
              precision    recall  f1-score   support

           0       0.84      0.87      0.85      9711
           1       0.97      0.82      0.89      7460
           2       0.81      0.93      0.87      2421
           3       0.63      0.57      0.60      2862
           4       0.07      0.84      0.12        63

    accuracy                           0.82     22517
   macro avg       0.66      0.81      0.67     22517
weighted avg       0.85      0.82      0.83     22517
```

Figure 14: Result of Support Vector Machine

C. Decision Tree Classifier

```
Accuracy of DT is 0.8735.
Confusion Matrix is:
 [[9180   98  212  215    6]
 [ 164 7261   32    3    0]
 [ 191   20 2210    0    0]
 [1820    5   42  984   11]
 [  20    0    1    8   34]]
Classification Report is:
              precision    recall  f1-score   support

           0       0.81      0.95      0.87      9711
           1       0.98      0.97      0.98      7460
           2       0.89      0.91      0.90      2421
           3       0.81      0.34      0.48      2862
           4       0.67      0.54      0.60        63

    accuracy                           0.87     22517
   macro avg       0.83      0.74      0.77     22517
weighted avg       0.87      0.87      0.86     22517
```

Figure 15: Result of Decision Tree

D. Random Forest Classifier

```
Accuracy of RF is 0.8815.
Confusion Matrix is:
 [[9216   63  212  215    5]
 [  72 7283  104    1    0]
 [  39    1 2381    0    0]
 [1922    1    2  932    5]
 [  17    0    0    9   37]]
Classification Report is:
              precision    recall  f1-score   support

           0       0.82      0.95      0.88      9711
           1       0.99      0.98      0.98      7460
           2       0.88      0.98      0.93      2421
           3       0.81      0.33      0.46      2862
           4       0.79      0.59      0.67        63

    accuracy                           0.88     22517
   macro avg       0.86      0.76      0.79     22517
weighted avg       0.88      0.88      0.87     22517
```

Figure 16: Result of Random Forest

E. Naive-Bayes

```
Accuracy of NB is 0.6238.
Confusion Matrix is:
 [[7665   22  458 1541   25]
 [  67 2687 3213 1251  242]
 [ 235  265 1908   13    0]
 [ 594   73  430 1745   20]
 [   8    0    0   15   40]]
Classification Report is:
              precision    recall  f1-score   support

           0       0.89      0.79      0.84      9711
           1       0.88      0.36      0.51      7460
           2       0.32      0.79      0.45      2421
           3       0.38      0.61      0.47      2862
           4       0.12      0.63      0.21        63

    accuracy                           0.62     22517
   macro avg       0.52      0.64      0.50     22517
weighted avg       0.76      0.62      0.64     22517
```

Figure 17: Result of Naïve-Bayes

VI. CONCLUSIONS AND FUTURE WORK

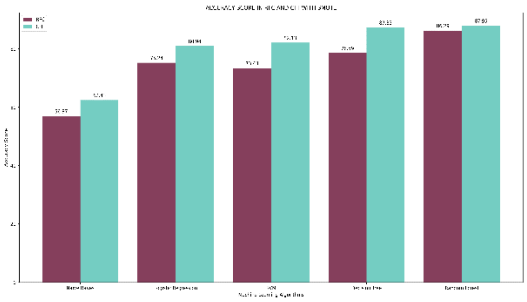


Figure 18: Bar graph showing Accuracy Score for RFE and CHI

The analysis of multiple classification models like Support Vector Machine, Logistic Regression, Decision Tree, Random Forest and Naïve-bayes for anomaly intrusion detection system is done. The performance of these models has been observed and studied on the basis of their accuracy and precision on the test data. The experiments proved that the classifiers are capable of handling high-dimensional data and still produce accurate results. The results indicate that the ability and accuracy of the Random Forest classifier outperform that of Others. The Random Forest and SVM took higher time in training and testing of the dataset as compared to Logistic Regression and Decision Tree. The accuracy in the results produced using Naïve-Bayes is lowest amongst all classifiers. It is also clear that When using Chi-square test over RFE accuracy of each model have increased, which shows better ability of chi-square test on such datasets. Hence, use of Chi-Square is preferred over RFE and Genetic Algorithm(IT TOOK OVER 2 Hours for feature selection).

	ALL_FEATURES	CHI_SMOTE	CHI_WITHOUT_SMOTE	RFE_SMOTE	RFE_WITHOUT_SMOTE	GA_SMOTE
Naive Bayes	65.32	62.375094	64.83	56.97	67.50	60.25
Logistic Regression	78.99	80.756762	78.84	75.24	70.71	69.56
SVM	77.49	82.000266	77.87	73.43	68.70	65.30
Decision Tree	83.66	87.351779	86.81	78.59	72.80	72.35
Random Forest	86.73	88.151175	87.69	86.29	73.90	74.05

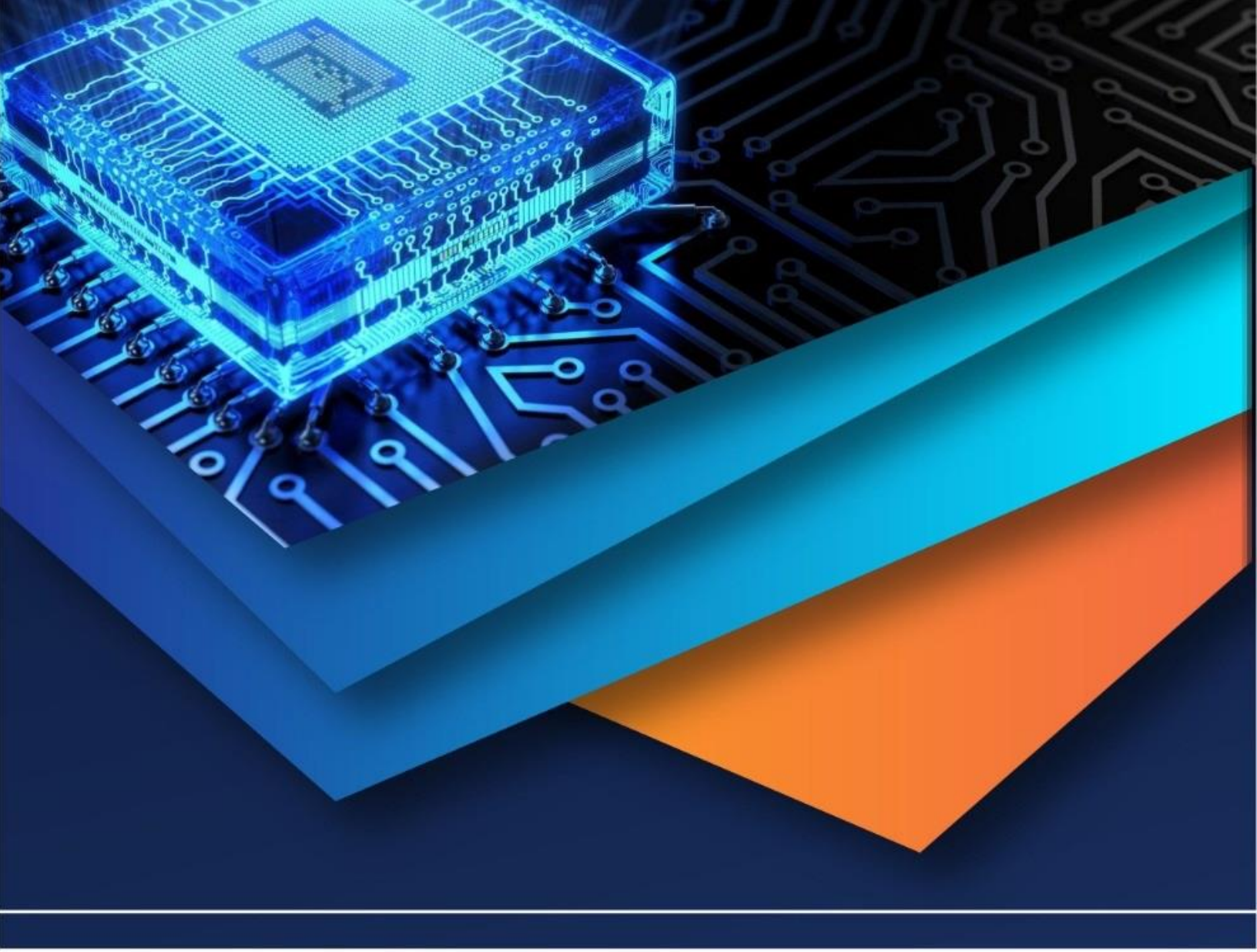
Figure 19: Figure showing Accuracy score of different models when used under different scenarios

This work can be continued by finding the new data instances which includes newer attacks which happened in recent times as the dataset used is not recent. Also, the analysis can be continued on newer datasets that have cutting-edge attacks information and get the best classifier to predict the possibility of a network intrusion.

REFERENCES

[1] A. A. Aburomman and M. B. I. Reaz, “A survey of intrusion detection systems based on ensemble and hybrid classifiers,” Comput. Secur., vol. 65, pp. 135–152, 2017
[2] Aljawarneh, M. Aldwairi, and M. B. Yassein, “Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model,” J. Comput. Sci., vol. 25, pp. 152– 160, 2018.
[3] P.Sangkatsanee, N. Wattanapongsakorn and C. Charnsripinyo,, “Practical Real-Time Intrusion Detection Using Machine Learning Approaches, Computer Communications” , vol. 34, no. 18, pp. 2227–2235, (2011).
[4] J. Manjula C. Belavagi and Balachandra Muniyal, “Performance Evaluation of Supervised Machine Learning Algorithms for Intrusion Detection” Twelfth International Multi-Conference on Information Processing- 2016.
[5] Adetunmbi A.Olusola., Adeola S.Oladele and Daramola O. Abosede, "Analysis of KDD 99 Intrusion Detection Dataset for Selection of Relevance Features", Proceedings of the World Congress on Engineering and Computer Science 2010,Vol I WCECS 2010, San Francisco, USA, October 20-22 2010.
[6] Modi, U., Jain, A.: An improved method to detect intrusion using machine learning algorithms. Inf. Eng. Int. J. (IEIJ) 4(2), 17–29 (2016). https://doi.org/10.5121/ieij.2016.4203

[7] S. Mukkamala, A. H. Sung, "Significant feature selection using computational intelligent techniques for intrusion detection", Advanced Methods for Knowledge Discovery from Complex Data, Springer, 2005, pp. 285–306.
[8] Prakash Kalavadekar, Dr. Shirish Sane "Effective Intrusion Detection Systems using Genetic Algorithm",International Journal on Emerging Trends in Technology, Volume 4, Special Issue July-2017, pp.8315-8319. Al Tobi, A.M.; Duncan, I. KDD 1999 generation faults: A review and analysis. J. Cyber Secur. Technol. 2018, 2, 164–200.
[9] Nadiammai, G.V.; Hemalatha, M. Effective approach toward Intrusion Detection System using data mining techniques. Egypt. Inf. J. 2014, 15, 37–50
[10] Patcha, A., Park, J.M.: An overview of anomaly detection techniques: existing solutions and latest technological trends. Comput. Netw. 51(12), 3448–3470 (2007)



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY