Big Data Classification using Machine Learning Algorithms for Intrusion Detection

*** Nilamadhab Mishra[1] , Dr. Sarojananda Mishra[2]**
[1] Research Scholar , Biju Patnaik University of Technology, Odisha, India
[2]Professor Indira Gandhi Intitute of Technology, Odisha, India
[1]E-Mail:nila76.mishra@gmail.com

Abstract:
Any unauthorized action on a computer network is referred to as network intrusion. The purpose of the network access software is to protect the computer network from unauthorized users, including internal ones. We will build an internal network detector, which is a predictive model that can distinguish between " bad 'connections, often referred to as intruders or attacks, and' good 'or normal connections. The purpose was to evaluate and evaluate the findings for access. We also focused on machine learning classification techniques in the KDD CUP 1999 data model for forecasting model. Here, we analyze our approach to existing technologies in order to get the best training and testing possible. Use different machine learning algorithms to create different models.

Keywords: Intrusion Detection, Big Data, Machine Learning, KDDCup data set.

1. Introduction

Data that is difficult to store, manage, and analyze using standard database techniques and software is called Big Data. Big Data [11] is defined by its large volume and speed, as well as the variety of data it contains, which allows for the development of new coping strategies [9]. Intrusion detection systems (IDS) [10] are hardware or software monitors that analyze data to identify any system or network attacks. When working with Big Data, an intrusion detection system makes the system more complex and inefficient because the process of analytics structures is complex and time-consuming. Because it takes longer to check the data, the system is at risk of being damaged for some time before receiving a warning. As a result, in access acquisition systems, Big Data and various methods can be used to analyze data, reduce computer time and training time. [1]

Signature-based detection, uncontrolled discovery, and bastard-based detection are three methods used by the IDS to detect threats. [12] Signature-based detection is used to detect known attacks by analyzing their signatures. It is a powerful tool for detecting known attacks stored in the IDS database. Because of this, it is often thought to be more accurate in detecting an intrusion attempt or known attack. New types of attacks, on the other hand, cannot be identified as their signature does not exist; of information is updated frequently to improve acquisition performance. To solve this problem, Identify disruptive actions that may be disruptive, using uncontrolled, discriminatory detection based on current user and pre-determined profiles. In addition to any system updates, anomaly-based detection is effective against anonymous or zero-day attacks. However, this approach has a high level of false benefits. [1]

Access Login is a software application that uses machine learning skills to identify network access. The IDS detects a network or system of harmful behavior and protects against unwanted access from users, including insiders. The goal of learning an intruder detector is to create a predictive model (e.g. separator) that can distinguish between "bad" (intrusion / attack) and "good" (normal) connections. [2]

There are four different types of attacks:

#DOS: denial-of-service (e.g. syn mood).
#R2L: illegal remote access (e.g. password guessing).
#U2R: unauthorized access to local super user (root) privileges, e.g., various “buffer over flow” attacks.

#probing: surveillance and another probing, e.g., port scanning.

Attacks detection considered as classification problem because the target is to clarify whether the packet either normal or attack packet. Therefore, the model of accepted intrusion detection system can be implemented based on significant machine learning algorithms. In this paper, machine learning algorithms have been Implemented (Gaussian Naive Bayes, Decision Tree, Random Forest, Support Vector Classifier, Logistic Regression, Gradient Descent)[8] to evaluate and accurate the model of intrusion detection system based on a bench market dataset Knowledge Discovery in Databases (KDD) which includes these types of attacks i.e. DOS, R2L, U2R, and PROBE.[3]

2. Data Set Used

KDD Cup 1999 dataset[7] is used to detect intrusion by implementing different machine learning algorithm.

Dataset Description: Data files:

- ✓ kddcup.names : A list of features.
- ✓ kddcup.data.gz : The full data set
- ✓ kddcup.data_10_percent.gz : A 10% subset.
- ✓ kddcup.newtestdata_10_percent_unlabeled.gz
- ✓ kddcup.testdata.unlabeled.gz
- ✓ kddcup.testdata.unlabeled_10_percent.gz
- ✓ corrected.gz : Test data with corrected labels.
- ✓ training_attack_types : A list of intrusion types.
- ✓ typo-correction.txt : A brief note on a typo in the data set that has been corrected.

Features:

Different features we are getting from the dataset which are given below:

Table 1: Basic features of individual TCP connections.[4]

Feature Name	Description	Type
duration	Length (number of seconds) of the connection	continuous
protocol_type	Type of the protocol, e.g. tcp, udp, etc.	discrete
service	Network service on the destination, e.g., http, telnet, etc.	discrete
src_bytes	Number of data bytes from source to destination	continuous
dst_bytes	Number of data bytes from destination to source	continuous
Flag	Normal or error status of the connection	discrete
land	1 if connection is from/to the same host/port; 0 otherwise	discrete
wrong_fragment	Number of “wrong” fragments	continuous
urgent	Number of urgent packets	continuous

Table 2: Content features within a connection suggested by domain knowledge.[4]

Feature Name	Description	Type
Hot	number of “hot” indicators	continuous
num_failed_logins	number of failed login attempts	continuous
logged_in	1 if successfully logged in; 0 otherwise	discrete
num_compromised	number of “compromised” conditions	continuous
root_shell	1 if root shell is obtained; 0 otherwise	discrete
su_attempted	1 if “su root” command attempted; 0 otherwise	discrete
num_root	number of “root” accesses	continuous
num_file_creations	number of file creation operations	continuous
num_shells	number of shell prompts	continuous
num_access_files	number of operations on access control files	continuous
num_outbound_cmds	number of outbound commands in an ftp session	continuous
is_hot_login	1 if the login belongs to the “hot” list; 0 otherwise	discrete
is_guest_login	1 if the login is a “guest”login; 0 otherwise	discrete

Table 3: Traffic features computed using a two-second time window.[4]

Feature Name	Description	Type
count	number of connections to the same host as the current connection in the past two seconds	continuous
	Note: The following features refer to these same-host connections.	
serror_rate	% of connections that have “SYN” errors	continuous
rerror_rate	% of connections that have “REJ” errors	continuous
same_srv_rate	% of connections to the same service	continuous
diff_srv_rate	% of connections to different services	continuous
srv_count	number of connections to the same service as the current connection in the past two seconds	continuous
	Note: The following features refer to these same-service connections.	
srv_serror_rate	% of connections that have “SYN” errors	continuous
srv_rerror_rate	% of connections that have “REJ” errors	continuous
srv_diff_host_rate	% of connections to different hosts	continuous

3. Machine Learning

Machine learning (ML) is a sort of artificial intelligence (AI) that allows software applications to improve their prediction accuracy without being expressly designed to do so. In order to forecast new output values, machine learning algorithms[6] use historical data as input.

Machine learning is frequently used in recommendation engines. Fraud detection, spam filtering, malware threat detection, business process automation (BPA), and predictive maintenance are all common applications.[13]

Types of Machine Learning:

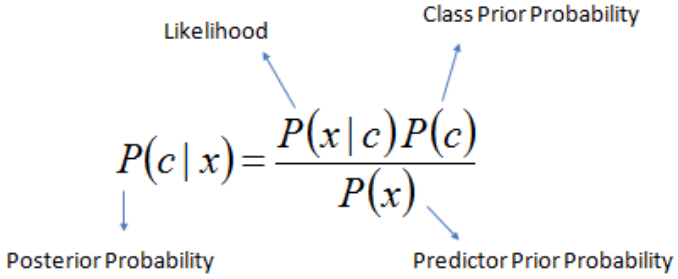
The way an algorithm learns to become more accurate in its predictions is how traditional machine learning is often classified. Supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning are the four primary methodologies. The algorithm that data scientists use is determined by the sort of data they wish to predict.

- **Supervised learning:** In this sort of machine learning, data scientists provide labeled training data to algorithms and specify the variables they want the programme to look for connections between. The algorithm's input and output are both provided.[13]
- **Unsupervised learning:** Algorithms that train on unlabeled data are used in this sort of machine learning. The algorithm looks for relevant connections between data sets. The data used to train algorithms, as well as the forecasts or suggestions they produce, are all predetermined.[13]
- **Semi-supervised learning**: This method of machine learning combines the two previous approaches. Although data scientists may feed an algorithm largely labeled training data, the model is allowed to explore the data and establish its own understanding of the set.
- **Reinforcement learning:** Reinforcement learning is a technique used by data scientists to teach a machine to perform a multi-step procedure with well-defined rules. Data scientists design an algorithm to perform a task and provide it with positive or negative feedback as it figures out how to do so. However, the algorithm, for the most part, selects what actions to take along the road on its own.[13]

Various Classification Algorithms Applied: Guassian Naive Bayes, Decision Tree, Random Forest, Support Vector Machine, Logistic Regression and Gradient Descent.

Guassian Naive Bayes (NB):

The Bayes theorem is used to create the naive bayes classification model. It's mostly utilised to deal with numeric values. This technique is mostly used to classify text and documents. This algorithm is simple to use and train, and it can easily predict classes. It is presumptively true that features are unaffected by class. The naive bayes algorithm is used in a variety of applications, including sentiment analysis, recommendation systems, and spam filtering. Gaussian naive bayes, Multinomial naive bayes, Complement naïve bayes, Bernoulli naive bayes, Categorical naive bayes, and out-of-core naive bayes are the six approaches of Naive Bayes.[5] The likelihood of a Naive-bayes classification model can be expressed as:



Decision Tree(DT):

DT is a basic supervised machine learning technique that is used for both classification and regression of a dataset using a series of decisions (rules). The model is structured like a tree, with nodes, branches, and leaves. Each node represents a feature or an attribute. Each leaf represents a possible outcome or class name, while the branch represents a decision or rule. To minimize over-fitting, the DT algorithm automatically selects the optimal attributes for building a tree and then prunes the tree to remove extraneous branches. CART, C4.5, and ID3 are the most prevalent DT models. Multiple decision trees are used in several advanced learning algorithms, such as Random Forest (RF) and XGBoost.[6]

Random Forest(RF):

Random Forest is a well-known machine learning algorithm that uses the supervised learning method. In machine learning, it can be utilized for both classification and regression issues. It is based on ensemble learning, which is a method of integrating several classifiers to solve a complex problem and increase the model's performance. "Random Forest is a classifier that contains a number of decision trees on various subsets of a given dataset and takes the average to enhance the predicted accuracy of that dataset," according to the name. The random forest takes the forecasts from each tree and predicts the final output based on the majority votes of predictions.

Support Vector Machine (SVM):

SVM is a supervised machine learning technique based on the concept of a hyper plane with maximum margin separation in n-dimensional feature space. It can be used to solve both linear and nonlinear problems. Kernel functions are used to solve nonlinear issues. The goal is to use the kernel function to transfer a low-dimensional input vector into a high-dimensional feature space. Then, utilizing the support vectors, an optimal maximum marginal hyper-plane is found, which serves as a decision boundary. By properly predicting the normal and harmful classes, the SVM method can be utilized to improve the efficiency and accuracy of NIDS. [6]

Logistic Regression (LR):

The supervised classification algorithm logistic regression uses only discrete values as input. It produces a regression-based model that predicts whether a given data has a chance of 1 or 0. Any of the categories used to classify data might be referred to as these values.

Gradient Descent (GD):

In machine learning and deep learning, gradient descent is the most used optimization approach. It's a first-order optimization procedure. This means that while updating the parameters, it only takes into consideration the first derivative. We update the parameters in the opposite direction of the gradient of the objective function J(w) with respect to the parameters on each iteration, where the gradient indicates the steepest ascending direction. The learning rate determines the amount of the step we take each iteration to reach the local minimum. As a result, we descend in the direction of the slope until we hit a local minimum.

Approach Used: I have applied various classification algorithms that are mentioned above on the KDD dataset and compare their results to build a predictive model.

Tool used for simulation is **Annaconda3 (32bits) of Python 3.8.5 .**

4. Structure of the simulation and details result:

In this paper there are two methodologies, we need to use:

A) Data Preprocessing
B) Modelling

Data Preprocessing :

i) Importing library and reading the list of features from “Kddcup.names” file
ii) Appending column to the dataset and adding a new column ‘target’ to the dataset. We are getting 42 features.
iii) Reading the ‘attack_types’ file which is given below:

Table-4 : List of attack _types

Attack_type	Class
back	dos
buffer_overflow	u2r
ftp_write	r2l
guess_passwd	r2l
imap	r2l
ipsweep	probe
land	dos
loadmodule	u2r
multihop	r2l
Neptune	dos
nmap	probe
perl	u2r
phf	r2l
pod	dos
portsweep	probe
rootkit	u2r
satan	probe
smurf	dos
spy	r2l
teardrop	dos
warezclient	r2l
warezmaster	r2l

iv) Creating a dictionary of the attack types.

v) Reading the dataset ('kddcup.data_10_percent.gz') and adding Attack Type feature in the training dataset where attack type feature has 5 distinct values i.e. dos, normal, probe, r2l, u2r.

vi) Shape of data frame and getting data type of each feature as (494021,43)

vii) Find out the missing values of all features as follows:

Table-5: Missing values of all features

Features	Missing Value
duration	0

protocol_type	0
service	0
flag	0
src_bytes	0
dst_bytes	0
land	0
wrong_fragment	0
urgent	0
hot	0
num_failed_logins	0
logged_in	0
num_compromised	0
root_shell	0
su_attempted	0
num_root	0
num_file_creations	0
num_shells	0
num_access_files	0
num_outbound_cmds	0
is_host_login	0
is_guest_login	0
count	0
srv_count	0
serror_rate	0
srv_serror_rate	0
rerror_rate	0
srv_rerror_rate	0
same_srv_rate	0
diff_srv_rate	0
srv_diff_host_rate	0
dst_host_count	0
dst_host_srv_count	0

dst_host_same_srv_rate	0
dst_host_diff_srv_rate	0
dst_host_same_src_port_rate	0
dst_host_srv_diff_host_rate	0
dst_host_serror_rate	0
dst_host_srv_serror_rate	0
dst_host_rerror_rate	0
dst_host_srv_rerror_rate	0
target	0
Attack Type	0
dtype: int64	

No missing value found, so we can further proceed to our next step.

viii) Finding Categorical Features which are given below:
['service', 'flag', 'protocol_type']

ix) Data Correlation – Find out the highly correlated variables using heatmap and exclude them for analysis.

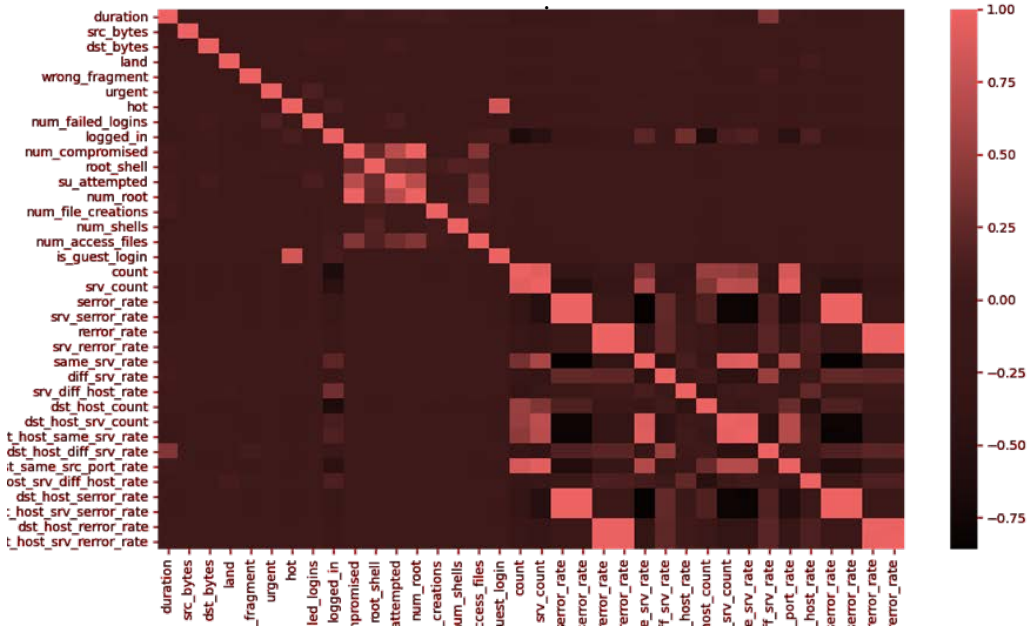


Figure-1 : Display the highly correlated variables using heatmap

x) Feature Mapping – Apply feature mapping on features such as : 'protocol_type' & 'flag'.
xi) Remove irrelevant features such as 'service' before modelling.

Modelling:

i) Importing libraries and splitting the dataset
   a) Splitting the dataset and get the result as : (494021, 31).
   b) Split test and train data and get the following result:

Table-6: Test and Train data

X_train	**X_test**
(330994, 30)	(163027, 30)
y_train	**y_test**
(330994, 1)	(163027, 1)

ii) Apply various machine learning classification algorithms such as Naive Bayes(NB), Decision Tree(DT), Random Forest(RF), Support Vector Classifier(SVC), Logistic Regression(LR) and Gradient Descent(GD) , we get the following training and testing score result which are given below:

Table-7: List of Training and Test Score

Algorithm	**Training Score**	**Test Score**
NB	87.951	87.903
DT	99.058	99.052
RF	99.997	99.964
SVC	99.875	99.879
LR	99.352	99.352
GD	99.793	99.771

iii) Analyse the training and testing accuracy of each model by using above table and get the following result graphically :

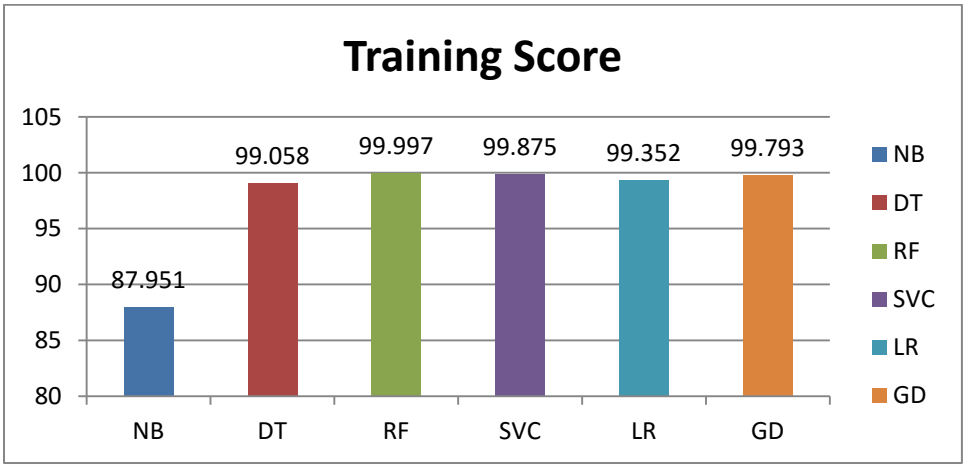


Figure-2: Analyse the training accuracy of each model.

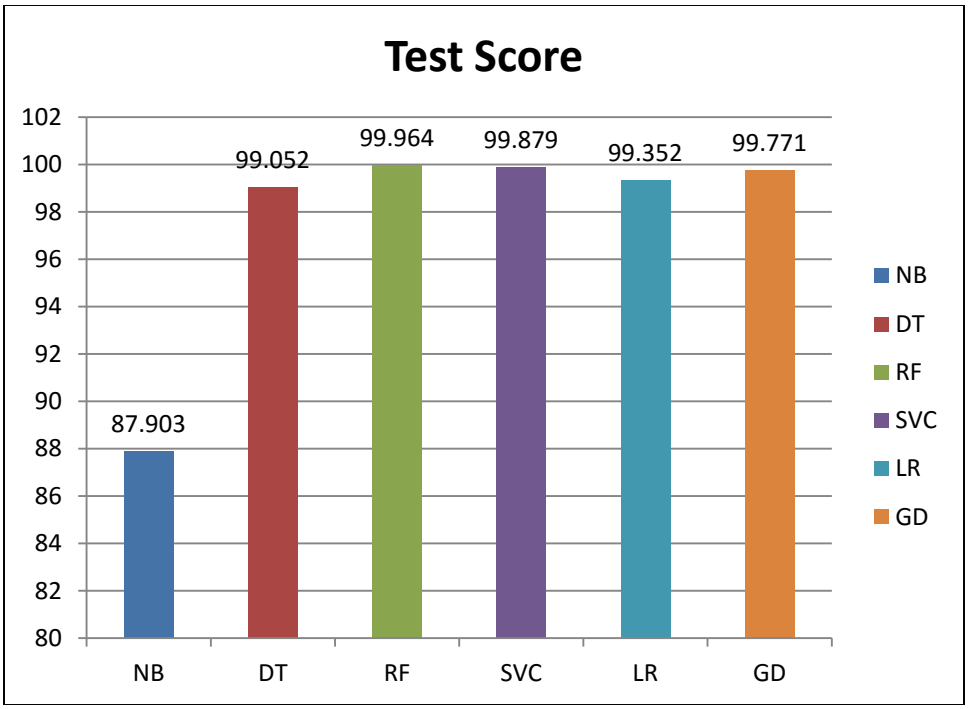


Figure-3: Analyse the testing accuracy of each model.

iv) Analyse the training and testing time of each model :

By applying the different algorithm , we get the above training and testing score but the time taken for training and testing, we get the following data results which are given below:

Table-8: Training and Testing Time

Algorithim	Training Time	Testing Time
NB	2.39	3.56
DT	6.03	0.50
RF	52.58	4.64
SVC	485.20	262.57
LR	386.35	0.51
GD	1945.99	17.54

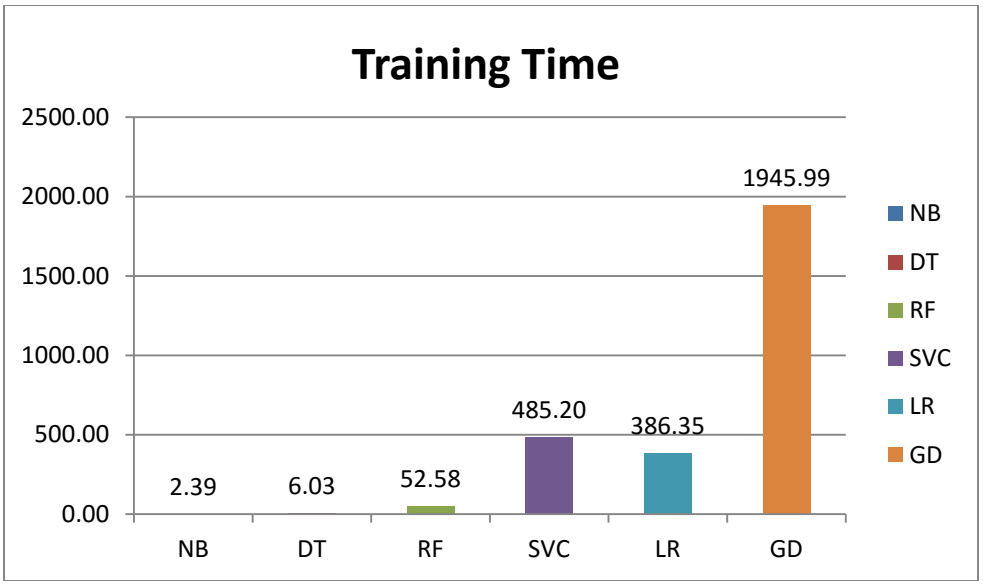


Figure-4: Analyse the training time of each model.

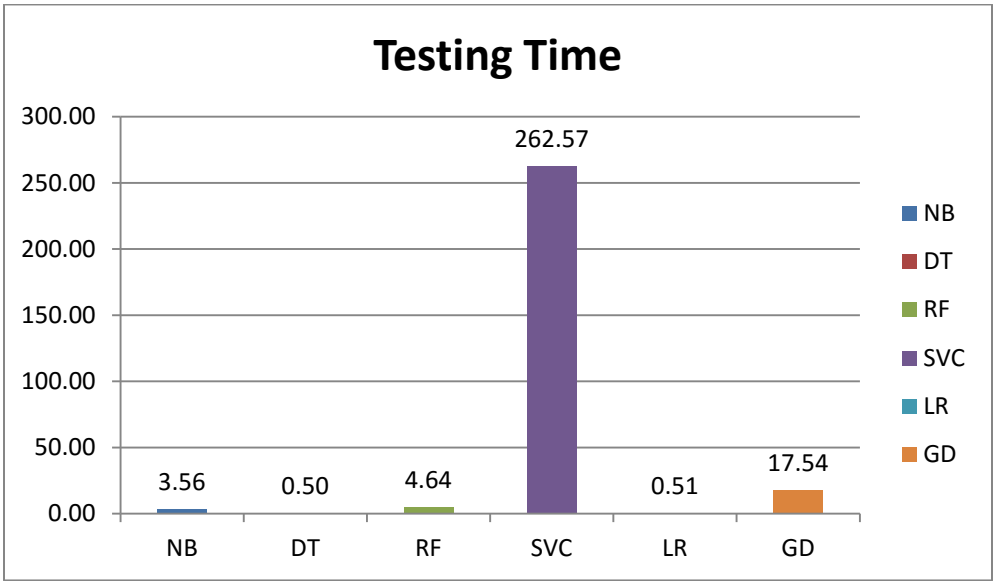


Figure-5: Analyse the testing time of each model.

5. Conclusion

In this analysis, we have to use six different Machine learning algorithm of classification in KddCup data set. Then analyze in terms of training score, testing score, training time and testing time of different models. From the analysis ,it is cleared that, Decision Tree model is the best fits of our data considering both accuracy and time complexity.

Reference

1. Suad Mohammed Othman, Fadl Mutaher Ba‐Alwi, Nabeel T. Alsohybe and Amal Y. Al‐Hashida, “Intrusion detection model using machine learning algorithm on Big Data environment”, Journal of Big Data, Othman *et al. J Big Data (2018) 5:34* , https://doi.org/10.1186/s40537-018-0145-4
2. Ansam Khraisat* , Iqbal Gondal, Peter Vamplew and Joarder Kamruzzaman, “Survey of intrusion detection systems: techniques, datasets and challenges”, Khraisat et al. Cybersecurity (2019) 2:20 https://doi.org/10.1186/s42400-019-0038-7.
3. Mohammad Almseidin, Maen Alzubi, Szilveszter Kovacs and Mouhammd Alkasassbeh, “Evaluation of Machine Learning Algorithms forIntrusion Detection System”, SISY 2017 • IEEE 15th International Symposium on Intelligent Systems and Informatics • September 14-16, 2017 • Subotica, Serbia.
4. Salvatore J. Stolfo, Wenke Lee, Andreas Prodromidis, Philip K. Chan, “Cost-Based modeling and evaluation for data mining with application to fraud and intrusion detection: Results from the JAM project”, online: https://www.researchgate.net/publication/228808098.
5. Ch. Aishwarya, N. Venkateswaran, T. Supriya, M. Sreekar, V. Sreeja, “Intrusion Detection System using KDD Cup 99 Dataset” , International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-9 Issue-4, February 2020.
6. Nilamadhab Mishra , Dr. Sarojananda Mishra, “On the NSL-KDD Dataset, a Survey of Machine Learning-based Intrusion Detection Systems”, Journal of Huazhong University of Science and Technology, ISSN-1671-4512, vol 50, issue 4.
7. Preeti Aggarwal, Sudhir Kumar Sharma, “Analysis of KDD Dataset Attributes - Class wise For Intrusion Detection”, 3rd International Conference on Recent Trends in Computing 2015 (ICRTC-2015), Procedia Computer Science 57 (2015) 842 – 851.
8. Zeeshan Ahmad, Adnan Shahid Khan, CheahWai Shiang, Johari Abdullah, Farhan Ahmad, “Network intrusion detection system: A systematic study of machine learning and deep learning approaches”, WILEY, DOI: 10.1002/ett.4150.
9. Nilamadhab Mishra, Sarojananda Mishra, “A Review on Big Data Classification: using Machine Learning Technique to Classify Intrusion” , IJRAR March 2020, Volume 7, Issue 1, E-ISSN 2348-1269, P- ISSN 2349-5138.
10. Nilamadhab Mishra et al, “ Intrusion Detection using IoT”,International Journal of Computer Science and Mobile Applications, ISSN: 2321-8363.
11. Neelamani Samal, Nilamadhab Mishra, “Big Data Processing: Big Challenges and Opportuni”, Journal of Computer Sciences and Applications, 2015, Vol. 3, No. 6, 177-180, DOI:10.12691/jcsa-3-6-13.
12. Yuyang Zhou , Guang Cheng , Shanqing Jiang , Mian Dai, “Building an efficient intrusion detection system based on feature selection and ensemble classifier”, Computer Networks 174 (2020) 107247.
13. Shagan Sah, “Machine Learning: A Review of Learning Types”, doi:10.20944/preprints202007.0230.v1 .