International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

A Novel Intrusion Detection Techniques of the Computer Networks Using Machine Learning

Nilamadhab Mishra[1]*, Sarojananda Mishra[2]

Submitted: 29/01/2023 **Accepted:** 04/04/2023

Abstract
Network intrusion is an unauthorized work of a computer network. Protecting the computer network against unauthorized users, even internal ones, is the goal of the network access programme. We will create a local network detection, which is a prophecy mold to discriminate amid "good quality" or typical associations and "appalling" associations, which sometimes referenced to as intruders or attacks. Evaluation of the findings for accessibility was the goal. In the Knowledge discovery Cups 1999 dataset for predicting, we also concentrated on machine learning-based classification to facilitate acquire greatest training and testing, to access our strategy for using currently available technologies. To generate various classification models, used varieties machine-learning based techniques and comparing each other for detecting best fit model for the computer networks with respect to time and accuracy.

Keywords: *Intrusion Detection Techniques, Machine -Learning, Computer Networks, Classification Approach.*

1. Introduction

Intrusion detection techniques (IDT) are a brand of either physical or program control technique which examine and identified data in the network or structure for intrusions[1]. The IDS uses three techniques to identify threats: bastard-based detection, uncontrolled discovery, and signature-based detection. By examine- ng their signatures, signature-based detection is used to identify known assaults. It is an effective technique for finding known assaults that have been recorded at the database of IDS. As a result, it also frequently believed an effective identifying attempted for imposition or known assault. On the other hand, new forms of assaults cannot be recognized since does not exist their signature; data-information is frequently modernized to enhance attainment concert. The issue is resolved by, utilizing uncontrolled, selective recognition based on the recent client and pre-well-known profile; identify acts that may be disruptive. Abnormality-based revealing is effectual next to unidentified 0'day threats in addition to any system updates. This strategy, though, offers a lot of fictitious advantages[2].

1Ph.D Scholar, Biju Patnaik University of Technology, Rourkela, Odisha, India,
nilamadhab76@gmail.com
2Professor, Indira Gandhi Institute of Technology, Sarang , Dhenkanal, Odisha, India

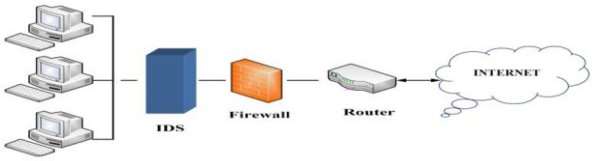

Figure 1. Network based Intrusion Detection System[3]

A computer programme called Access Login recognizes network access by means of machine learning. The IDS guards against unauthorized access from users, including insiders, and recognizes a network or system exhibiting dangerous activity.
The objective of the study is to develop an intruder prediction form which can differentiate the connections is normal or attack connections.

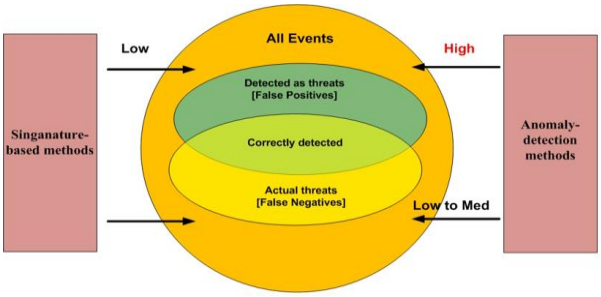

Figure 2. Signature-based-vs-anomaly-based-methods [4]

Four different attacks shown in the following table:
sarose.mishra@gmail.com

Table 1. Attacks-Types

Attacks- Types	Description	Examples
dos	Security incident to interrupt the network services.	Disconnect of the network service(pod)
r2l	Instead of having login into the system the packets are send to the remote system by connecting to the network.	Guessing of password(imap)
u2r	Due to different vulnerabilities of the system , it will exploit the unauthorised access to the local user.	Over flow of the buffer (loadmodule)
Probing	To gather server data packets, it will identify the Internet Protocol address by scanning the network.	Scanning of port(satan)

Classification problem is an attack detection problem to clarify the packet of data is either attack or normal type. Therefore, the IDTs have been implemented with different machine- learning (ML) methodologies. Here the author, implemented different ML algorithms to examine the intrusion detection based upon the approved-dataset, Knowledge-Discovery (KDD) included the attacks-types like Daniel-of-Service (dos), r2l, u2r, and Probe[5].

1.1 Literature

IDTs based literature is abounding with recent machine learming methods. Many proposed IDTs models found in classical machine learning methods which give low accuracy as well as the depends on the manual process to design the traffic features. In Bigdata [6], such type of the exercise is purely impossible for the man and hence the method can consider as inactive.

In the above situation, the authors[7] have proposed, no algorithm can handle itself all attack types efficiently. The lowest false negative value was achieved by the rule based classifier, but it did not have the highest detection accuracy rate. In case Bayes-network classifier, it has highest correct detection capability of the normal packets. On the other hand, Random forest classifier has the rate of accuracy high with less RMSE-value and rate of false-positive. It has been predicted that the Random-forest classifier present adequate performance-parameters without the parame- ter of false negative.

The authors[8] gives the importance to improve KNN classifier in available intrusion detection work which joins both K-MEANS cluster algorithm and KNN classification model to enhance the functionality of the model.

The authors have proposed [9] the classification of traffic based upon IDTs with convolutional neural network considering the importance of the classification of traffic before anomaly detection.

The problem of high dimensional data have proposed by the author[10], which typical to traffic of network and implement the deep belief network(DBN) to reduce the dataset into stunted dimension.

IDTs proposed in [11] feature size reduction working using PSO, ACO, and GA in all dataset of NSL-knowledge-discovery and UNSW-NB15 followed by classifier REPT (“Reduced-error pruning-Tree”) for feature selection. At the end achieved the infringement-type classification by collection of classifiers, like, bagging and forest rotation methods.

The authors in [12] enhanced the detection mechanism by implementing random forest model(multi-level) to predict anomalous behaviour of the network.

In the process of selecting useful features, authors have proposed an SVM-based IDT in [13] in which parameters and weights are optimized by utilizing characteristics of both the genetic algorithm and SVM.

However, the IDT that was proposed by the authors in [14] also makes use of CNN and the entire NSL-KDD dataset, which helps it perform better than traditional machine learning techniques like random forest (RF) and support vector machines (SVM), as well as certain deep learning algorithms like Long-short-Term-memory (LSTM) and DBN. Particle swarm optimiza-tion, in which the number of nodes per hidden layer is optimized, also improves the performance of the projected IDT form. In conclusion, classification of low-dimensional data makes use of PNN.

In [15][16] the authors propose an IDT model that combines the mechanism of attention with Bidirectional long short-term memory (BLSTM), which uses the BLSTM method to automatically extract traffic data from network flow. The adopted artificial intelligence classifier uses unprocessed data as its input, not features that were manually designed. The authors of this learning strategy did not address the tuning of CNN's parameters. Additionally, the ML method used was not tested for its capability. Because it is compressed, the proposed method does not validate unknown malware traffic, which indicates the scope of subsequent work.

In order to detect all types of attacks, including user2root (u2r) and remote2local (r2l) attacks, the authors of [5] acknowledged that a single ML classifier is not helpful. Instead, they suggested using signature-based IDTs to detect these attacks. As a result, the proposed IDT employs a two-layered hybrid strategy in which Naive-Bayes identify Daniel-of-Source and PROBE in layer one and SVM detection of u2r and r2l in layer two accomplish the desired objective.

Table 2. Snapshot of Literature Survey

References	Year	Algorithm	Main Contribution	Field
Mohammad Almseidin et al. [7]	2017	Machine Learning Classifier	Effectiveness and performance evaluated	Machine–Learning(ML)
Shapoorifard et al. [8]	2017	K-MEANS, KNN	Combination of algorithms to increase the detection accuracy using K-MEANS and KNN	Machine –Learning(ML)
Wang et. al. [9]	2017	CNN	Improved the detection of traffic data Using CNN	Deep-Learning(DL)
Zhao et. al. [10]	2017	PNN,DBN	The DBN and PNN models and PNN are used for classification in order to limit the scope of the data.	Deep –Learning(DL)
K.Vengatesan et. al.[17]	2018	DL	Stacked Auto encoder procedure is superior to that in RBM	Deep –Learning(DL)
Suad Mohammed Othman et al.[1]	2018	Chi-Logistic Regression with Chi-SVM	To select related features used ChiSqSelector and SVM With SGD to differentiate data . This model have high performance and speed	Machine –Learning(ML)
Tao et al. [13]	2018	GA, SVM	Optimized by genetic algorithm using weights of SVM.	Machine- Learning(ML)
Yale Ding et al.[14]	2018	LSTM and DBN	For better accuracy and type of intrusion proposed CNN-based IDS.	Deep Learning and Machine Learning
Jiadong et al. [12]	2019	Random-Forest	Through the multilevel random forest model, abnormal network behaviour have been detected.	Machine Learning(DL)
Bayu Adhi tama et al.[11]	2019	ACA, PSO and GA	2-stage meta-classifier and hybrid-feature-selection implemented.	Features selection(Hybrid)
Ch. Aishwarya et al. [2]	2020	Naive-bayes, J-48, and Random -forest	For the design and implementation of Random-forest, work well in IDT.	Machine Learning(ML)
Zeeshan Ahmad et al. [18]	2020	ML and DL	AI-based NIDS	Machine Learning(ML)/ Deep –Learning(DL)
Su et. al. [15]	2020	LSTM , CNN	To detect each attack type LSTM and CNN models are proposed.	Deep Learning(DL)
Treepop W. et al.[5]	2021	Naive-Bayes and SVM classifier	A double-layered hybrid approach (DLHA) was proposed by the authors.	Machine- Learning(ML)
Yanfang Fu. et. al.[16]	2022	ADASYN and CNN	For better performance classification proposed a model DLNID.	Deep Learning(DL)

2. Background

As machine learning is the kind of artificial intelligence which accepts programming application for enhancing the future forecast exactness without premeditated to do so. For forecasting of new output values, Machine-Learning-algorithms (MLA) [19] utilize chronological input facts.

The machine-learning-techniques are commonly used in recommendation engines. Also different techniques are used for scam-recognition; spamming-filtration; detection of malware risk; Business-process-automation (BPA); as well as analytical maintenances are general relevances. [20]

2.1 Machine-Learning-Types:

Conventional machine-learning is frequently categorized as the practice, through this algorithm, increase the accurateness of its predictions. Four main approaches are supervised, unsupervised, semi supervised, and reinforcement learning methods. By utilizing the data, data analysts want to foresee the algorithm that they choose.

- **Supervised learning:** As the name suggests the supervision i.e. called supervised-machine-learning. This means, we have trained "labeled" data, and hence the machine predicted the output, depending upon the training. In this context, the labeled data represents which inputs mapped with which output. More accurately, we may state that after the training of the data the machine work with input and have given related output, and then it will use to predict the outcome using test-dataset [20].

- **Unsupervised learning:** This type of Machine-learning employs algorithms which train on unlabeled dataset. The programme find out the link between datasets that are pertinent. The forecasts or recommendations generated by algorithms as well as the data utilized to train them are predefined [20].

- **Semi supervised learning:** Semi-supervised learning is a type of machine-learning algorithm that lies in between supervised and unsupervised learning**.** This is a supervised algorithm that uses labeled data and an unsupervised algorithm that uses unlabeled data. During the training process, the grouping of labeled and unlabeled data sets was used.

- **Reinforcement-Learning:** Artificially intelligent software components automatically explore their surroundings by hitting and trailing, take action, learn from experience, and improve their performance. This is the basis of the reinforcement method. Maximize rewards is the mantra of the reinforcement method [20].

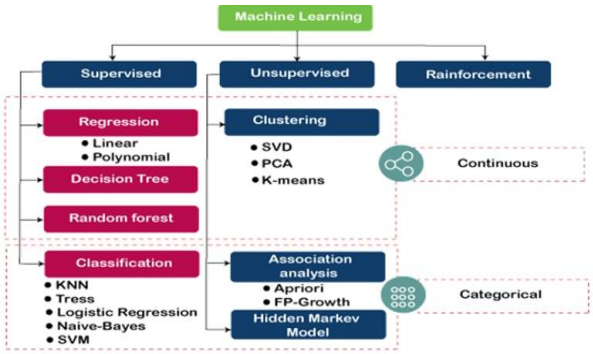

Figure 3. Types of Machine Learning[21]

2.1 Various Classification Algorithms:

Guassian-Naïve-Bayes Algorithm(NBA):

The method is implemented to establish a classification model with only numerical values and classify both documents and text. It is very simple to train and use also can easily predict classes. It is a given that class has no bearing on features. Applications for the naive bayes algorithm include reaction study, recommender system, and spam filter. [21] A Naive-bayes classifier is written like below:

$$P(c|x) = \frac{P(x|C)P(c)}{P(x)} \qquad -- \quad (1)$$

where c and x both are events and P(x) ≠ 0.

- Given that the event "x" occurs, determine P(c). Evidence is another word for the event "x."
- P(c) is 'c's priori [Seen the probability of event, before evidence]. Attribute value of an unidentified case (in this case, "x" event).
- p(c|x) represents the likelihood of an event "x" a posteriori, or after evidence has been observed.

Decision Tree Algorithm(DTA):

DTA is the fundamental of supervised-learning method which uses a series of decisions, for classification and prediction of data (rules). The model is organized into nodes, branches, and leaves like a tree. Every node stands for a property or an attribute. The branch stands for a choice or a protocol, where each leaf denotes a potential outcome or a name of the class. The DTA method automatic chooses the good qualities for constructing tree, and henceforth prunes

the tree to get rid of needless branches in order to reduce over-fitting. Common DTA models are the CART, C4.5, and IDS3 types. Several sophisticated learning-algorithms which are available like random-forest-algorithm (RFA) and XG Boost-Algorithim, employ multiple decision trees [22] .

Below diagram Figure 4 explains the decision tree general structure:

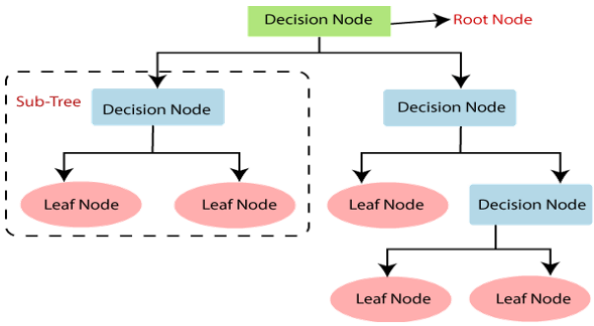

Figure 4. Decision tree general structure[23]

Random Forest Algorithm (RFA)

RFA is an established machine learning algorithm that falls under the supervised approach category. It can be used to solve problems related to classification and regression. Adapted the model's performance capability by combining various classifiers to tackle a problem using ensemble learning. "Random-forest(RF) algorithm incorporates multiple decision-trees on different subsets of the given dataset and takes the average to boost the projected accuracy of that dataset," as the name suggests, is the function of the "random-forest" algorithm. The forecast from each decision tree and the majority prediction of votes are then used by the random forest algorithm to predict the final outcome.

In the below figure (Fig. 5) explained the Random-forest algorithm:

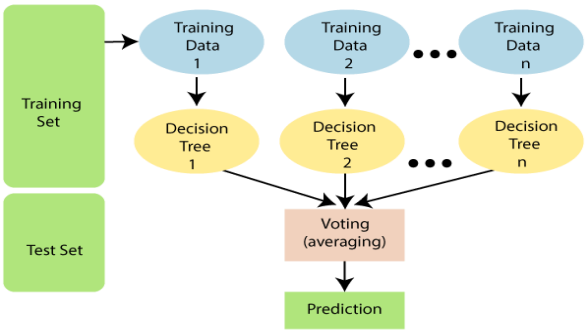

Figure 5. Model for the Random Forest Algorithm in use.[24]

Algorithm for Support Vector Machines (SVMA):

The idea of the hyper plane along with greatest partitioned of margin in n^{th}-dimensional attributes spaces serve as the foundation for the supervised machine learning method known as SVM. It is capable of handling both linear and nonlinear issues. Nonlinear problems are resolved with the function of kernel. The objective is to convert a vector of low dimensional inputs into a high dimensional features by implementing the kernel function. The ideal maximize marginal-hyper-plane is then discovered using the support vectors and acts as a decision boundary. The accuracy and efficiency of NIDS can be increased by using the SVM algorithm to accurately forecast the normal and dangerous classifications. [5]

Extreme vectors and points can be selected by SVM that help to create hyper-plane. Support-vector, which are symbolize these excessive instances on the basis of the support vector -method. The Fig-6 is given below represents the hyper-planed which classified into two different parts:

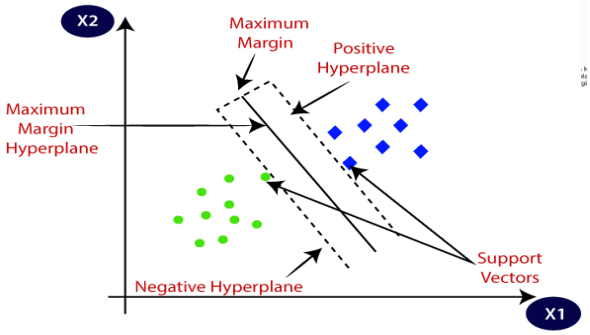

Figure 6. Classification using decision boundary or hyper-plane.[21]

Logistic Regression (LR):

Logical regression, a supervised classification algorithm, only accepts distinct value as input and generates a regression-based-model that foretells whether known pieces of information have a likelihood of being 1 or zero (0). These values can refer to any of the classifications used to group data.

Logistic regression can be used rapidly to identify the factors that will work well when classifying observations using various sources of data. The logistic-function is depicted in the graphical representation which is given below (Figure 7):

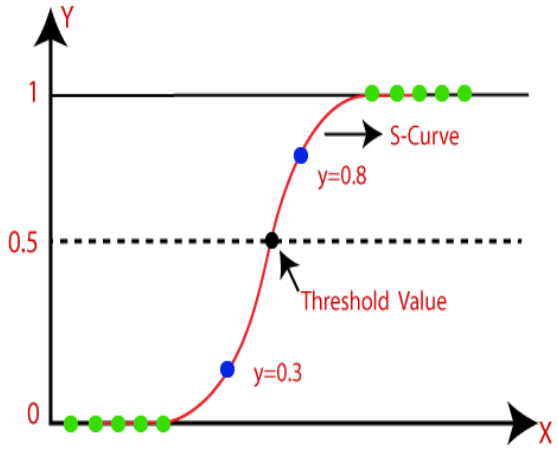

Figure 7. Representation of the Logistic Function[25]

Gradient Descent Algorithm (GDA):

The most frequent optimization method is gradient-descent, which is utilized in deep-learning and machine-learning algorithm. It is a forwarded optimization technique which is used to consider the first derivative when changing the parameters. In every repetition, we have to change the parameter in the reverse path of the goal function J (w) gradient, where the gradient denotes the sharpest ascending direction. To achieve the local-minimum, take the size of each step depending on the rate of learning. As a result of which, need to continue or to move downward until reach to a local minimum.

As shown below in Figure 8, the ideal gradient-descent representation of the local minimum or local maximum of a function is as follows:

The local minimum of that function will be revealed if we walk in the direction of a negative gradient or away from the present point's gradient.

The local maximum of a function is obtained whenever move in the direction of a +ve gradient or the Gradient of the function at the current point.

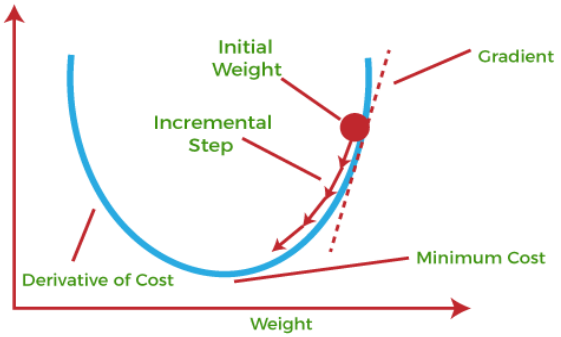

Figure 8. Using gradient-descent represents the local minimum and maximum[26]

Gradient Ascent, or steepest decline, is the term used to describe the entire process. The important purpose of a Gradient-descent method is to repeat minimizing cost-function. It carries out the following steps repeatedly to reach the goal:

- Determines the gradient function or slope by computing the function's first-order derivative.
- Moving to the reverse way of the gradient, which indicates that the slope has increased from the present location by an amount equal to alpha times, where alpha is the learning rate. The length of the stages is determined by this tuning parameter used in the optimization process.

3. Simulation and Result

3.1 Workflow Diagram:

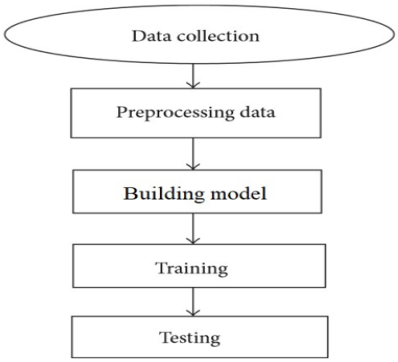

Figure 9. Workflow diagram of the data analysis

3.2 Preprocessing of Data :

i) List of features reading from “Kddcup. names” file by importing concern file.

ii) Adding new column to the dataset as ‘target’ through which find out 42-features.

iii) Reading of ‘Attack_Types’ file shown in the table-3:

Table 3. List of attack _types

CLASS	ATTACK TYPE
dos	Disconnect of the network service(pod, Neptune, Smurf)
r2l	Guessing of password(multihop, Phf, Warezclient)
u2r	Over flow of the buffer (loadmodule, Rootkit, Perl)
Probing	Scanning of port (portsweep, Nmap, Satan)

iv) Creating of the dictionary using attack types.

v) The reading and features of the [attack-type] dataset (["kddcup.data_10_percent.gz") have been added to the training_dataset. This dataset contains five distinct attack type features: dos, normal, Probe, u2r, and r2l.

vi) Determining the data type of each feature and shaping the data frame.

vii) Finding missing values which as follows:

Table 4. Missing values of all the features

Features	Value of Missing
Attack Type	'0'
count	'0'
diff_srv_rate	'0'
dst_bytes	'0'
dst-host-count	'0'
dst_host-diff_srv-rate	'0'
dst_host_rerror_rate	'0'
dst_host-same_src-port_rate	'0'
dst_host_same_srv-rate	'0'
dst_host_serror_rate	'0'
dst_host_srv_count	'0'
dst_host_srv_diff_host_rate	'0'
dst_host_srv_rerror-rate	'0'
dst_host-srv_serror_rate	'0'
duration	'0'
flag	'0'
hot	'0'
is_guest_login	'0'
is_host_login	'0'
land	'0'
logged-in	'0'
num_access_files	'0'
num_compromised	'0'
num_failed_logins	'0'
num_file_creations	'0'
num_outbound_cmds	'0'
num_root	'0'
num_shells	'0'
protocol_type	'0'
rerror_rate	'0'
root_shell	'0'
same_srv_rate	'0'
serror_rate	'0'
service	'0'
src_bytes	'0'
srv_count	'0'
srv_diff_host_rate	'0'
srv_rerror_rate	'0'
srv_serror_rate	'0'
su_attempted	'0'
target	'0'
urgent	'0'
wrong_fragment	'0'

Nothing missing value have found, then we go for further step.

viii) Categorical Features have been found out:
['service-', 'flag-', 'protocol-type']

ix) Finding correlated variables by the implementation of heat-map and exempted them for scrutiny.

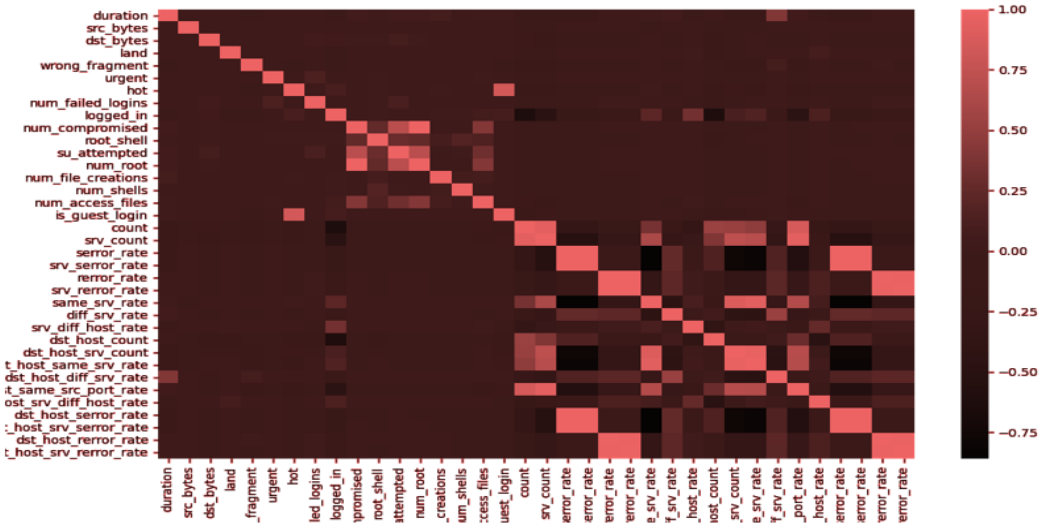

Figure 9. Displaying highly-correlated variables using heat-map

x) Mapping of feature – Applied feature mapping on 'protocol-type' & 'flag-'.

xi) Removed unrelated facial appearance for instance 'service' before modeling.

3.3 Modeling:

i) Libraries importing and dataset splitting

a) Dataset divided as [494021, 31].

b) Training and Testing data splitting which available in table-5.

Table 5. Training and Testing data Splitting

X_train_data	X_test_data
[(330994), (30)]	[(163027), (30)]
y_train_data	**y_test_data**
[((330994), (1))]	[((163027),(1))]

ii) Using various machine-learning classification algorithms like: We obtain the following trained and tested results from the Naive Bayes Algorithm (NBA), Decision Tree Algorithm (DTA), Random Forest Algorithm (RFA), Support Vector Classifier Algorithm (SVCA), Logistic Regression Algorithm (LRA), and Gradient Descent Algorithm (GDA) represent in Table-6:

Table 6. List of Score Training and Testing

Algorithim	Training	Testing
NBA	87.951	87.903
DTA	99.058	99.052
RFA	99.997	99.964
SVCA	99.875	99.879
LRA	99.352	99.352
GDA	99.793	99.771

4. Datasets

The dataset of KDDCup1999 [27] implemented to identify interruption by adopting variety algorithm of Machine-learning.

4.1 Description of Dataset:

Data-files:

Dataset File	File_description
[(" KddCup.Names")]	Features list
[("KddCup.data.gz")]	Complete dataset
[("KddCup.data_10_percent.gz")]	Subset of original 10% file
[("KddCup.newTestdata_10_percent_unlabeled.gz")]	10% unlabelled test-data file
[("KddCup.testdata.unlabeled.gz")]	Unlabelled test-data complete
[("KddCup.Testdata.unlabeled_10_percent.gz")]	Unlabelled test-data 10%
[("Corrected.gz")]	Test-data with corrected labeled data
[("Training_Attack_Types")]	list of Intrusion types
[("Typo-Correction.txt")]	Note on a typo in the corrected dataset

4.1.1 Features:

Dataset contains different features which are shown below:

Table 7. Individual TCP connections basic features.[28]

Features Names	Description of Features
Duration_	The Connection's Length
Protocol type	Protocol type
Services	destination of network service
Src_byte	Bytes from source to destination
Dst_byte	Bytes from destination to source
Flags_	connection status i.e. “normal” or” error”
Land	"1" for the same host or port, or "0"
wrong_fragment_	Identifies the number of the "wrong" fragments
Urgent_	Compile the urgent packages.

Table 8. Domain knowledge suggested within connection content features[28]

Features_Names	Description of Features
Hot_	Indicates “hot” number
num_failed_login	Number of unsuccessful login attempts
Logged-in	is "1" for successful login, or else "0,"
num-compromised	number of conditions that are "compromise"
root-shell	If the root shell is obtained, "1" and "0" respectively indicate so.
su_attempted	The attempted "su root" command is either "1" or "0."
num_root	Count the access of “root”
num_file_creation	number of operations used to create files
num_shell	hints of the shell's number
num-access_file	number of access operations to the control file
num_outbound_cmd	number of ftp outbound commands per session
is_hot-login	either "1" or "0" if the login is on the "hot" list.
is_guest-login	If you log in as a "guest," the value is 1, or nothing.

Table 9. Two seconds time window features calculated using traffic. [28]

Features	Description of each features
count	Count the past connections in two seconds to the same server
The below given features are referring the same host connection	
serror_rates	Errors “SYN” percentage connection
rerror_rate_	Errors “REJ” percentage of connection
same_srv-rate	same service connection percentages
diff_srv-rate	variety of services connection percentages
srv-count	Check the number of connections to the same service in the last two seconds.
Same service connections referring below are given	
srv_serror-rate	percent of connections that have an "SYN" error
srv_Rerror-rate	Number of connections that have "REJ" errors
srv_Diff_host-rate	proportion of variety hosts with connections

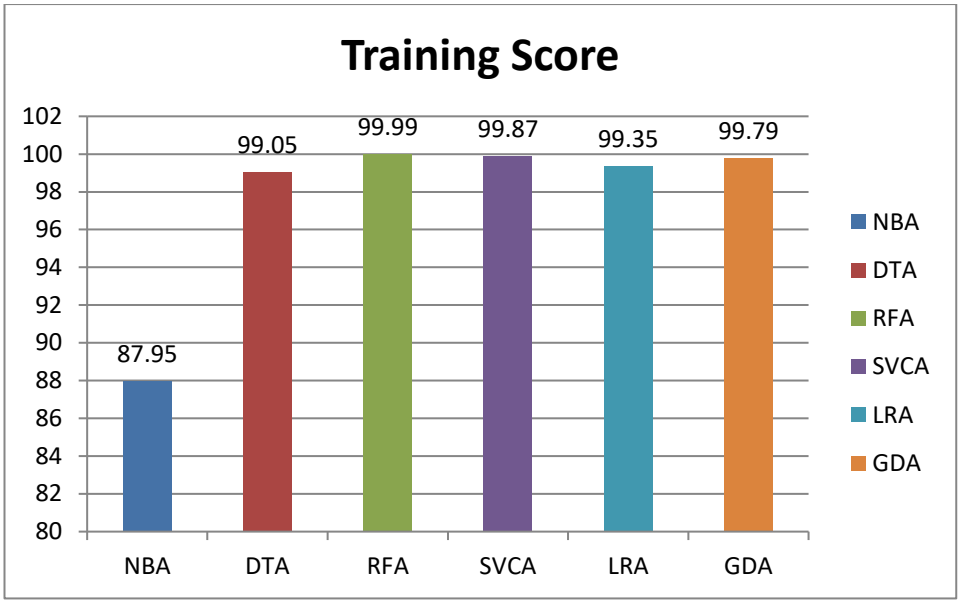

Figure 10. Training accuracy analysis.

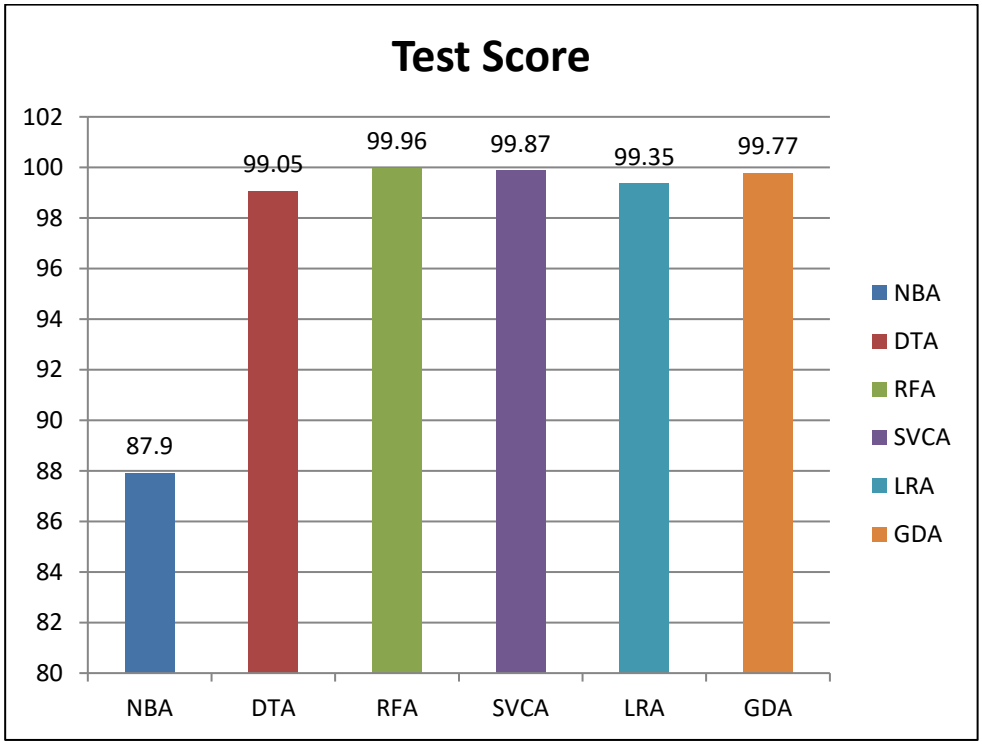

Figure 11. Testing accuracy Analysis.

5. Result Analysis

i) The train and test accuracy of each model analysis using Table:6 is given in Figure 10:

ii) Training and testing time analysis :

Different algorithms should be implemented on train and test dataset and the time required to train and test data results are given in below table-10:

Table 10. Time for Training and Testing the data using different algorithms

Algorithim	Time for Training	Time for Testing
NBA	2.3918	3.5568
DTA	6.0314	0.50002
RFA	52.5836	4.6414

SVCA	485.2031	262.5731
LRA	386.3503	0.5148
GDA	1945.9869	17.5354

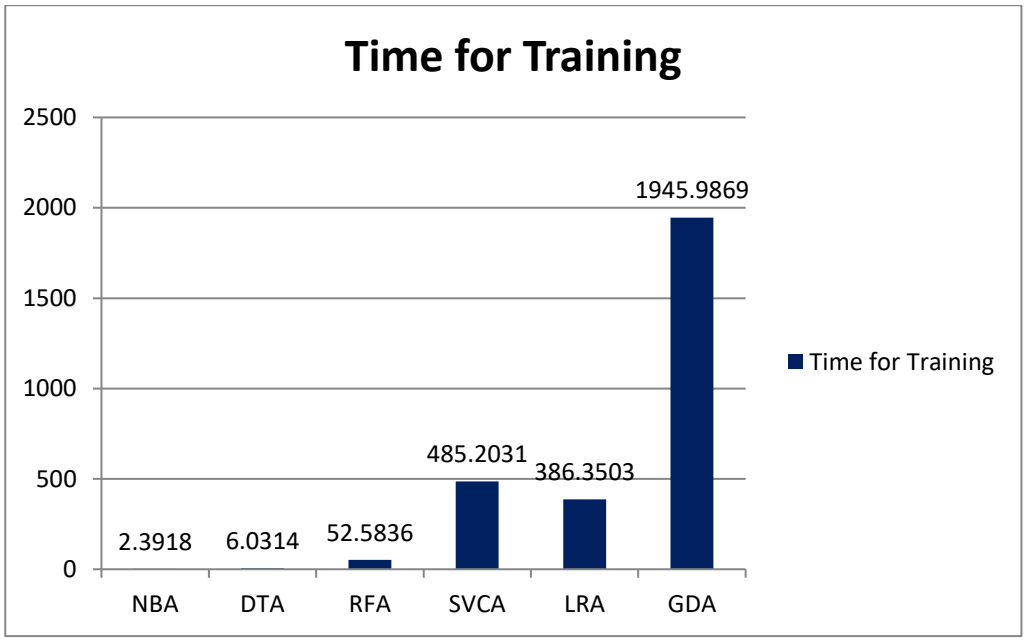

Figure 12. Analysis of the training time

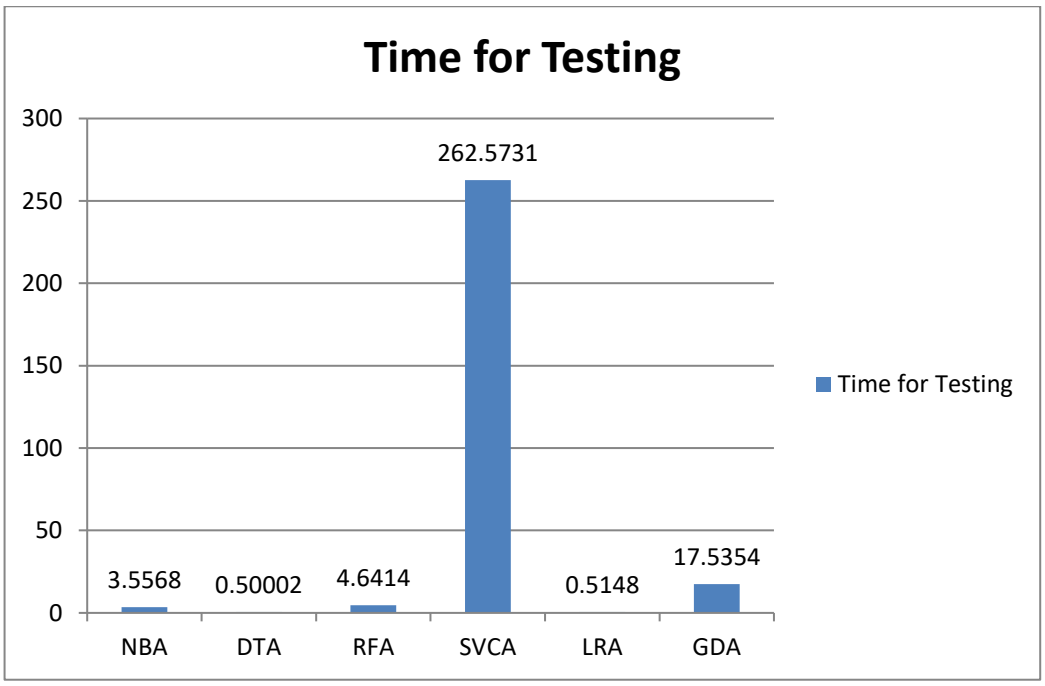

Figure 13. Analysis of the testing time

6. Classification Report using different Machine Learning Techniques:

Table 11. Classification report using Gaussian Naive Bayes

	precision	recall	f1-score	support
DoS	1.00	0.94	0.97	389717
R2L	0.03	0.42	0.05	125
U2R	0.01	0.83	0.03	6

Probe	0.02	0.99	0.04	456
Accuracy			0.94	390304
Macro-Avg	0.21	0.64	0.22	390304
Weighted-Avg	1.00	0.94	0.97	390304

Table 12. Classification report using Decision Tree

	precision	**recall**	**f1-score**	**support**
DoS	1.00	0.94	0.97	389717
R2L	0.64	0.84	0.72	125
U2R	1.00	0.50	0.67	6
Probe	0.02	1.00	0.04	456
Accuracy			0.95	390304
Macro-Avg	0.53	0.66	0.48	390304
Weighted-Avg	1.00	0.95	0.97	390304

Table 13. Classification report using Random Forest

	precision	**recall**	**f1-score**	**support**
DoS	1.00	1.00	1.00	389717
R2L	0.92	0.99	0.95	125
U2R	0.50	0.83	0.62	6
Probe	0.69	0.99	0.81	456
Accuracy			1.00	390304
Macro-Avg	0.62	0.76	0.68	390304
Weighted-Avg	1.00	1.00	1.00	390304

Table 14. using Support Vector Classifier classification report

	precision	**recall**	**f1-score**	**support**
DoS	1.00	0.99	1.00	389717
R2L	0.76	0.93	0.84	125
U2R	1.00	0.50	0.67	6
Probe	0.51	0.98	0.67	456
Accuracy			0.99	390304
Macro-Avg	0.66	0.68	0.63	390304
Weighted-Avg	1.00	0.99	1.00	390304

Table 15. Using Logistic Regression classification report

	precision	**recall**	**f1-score**	**support**
DoS	1.00	0.99	1.00	389717
R2L	0.74	0.90	0.81	125
U2R	1.00	0.50	0.67	6
Probe	0.53	0.96	0.68	456
Accuracy			0.99	390304
Macro-Avg	0.65	0.67	0.63	390304
Weighted-Avg	1.00	0.99	1.00	390304

Table 16. Using Gradient-descent classification report

	precision	**recall**	**f1-score**	**support**
DoS	1.00	1.00	1.00	389717
R2L	0.97	0.99	0.98	125
U2R	0.50	0.67	0.57	6
Probe	0.74	0.99	0.84	456

Accuracy			1.00	390304
Macro-Avg	0.64	0.73	0.68	390304
Weighted-Avg	1.00	1.00	1.00	390304

7. Conclusion

In this analysis of intrusion detection technique, the best possible machine learning algorithm to fit into a high performance effective IDT, six different machine learning algorithms have been modeled to classify normal and bad attack types. All the classifiers have been trained and tested using KddCup data set. The classifiers performance have been analyzed in the form of varied performance measures, such as evaluate their train and test result score, train and test timings using variety techniques and also produced the classification report of each technique. Depending upon the train and test, time and score of the analyzed report, it is found, the decision-tree-model (DTM) is one of the better classification technique of data implemented in this work and is evaluating the accurateness and time complication as per the classification result, which is better in comparison to other authors done in their research.

Conflict of Interest

The authors confirm that there is no conflict of interest to declare for this publication.

Acknowledgments

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors. The authors would like to thank the editor in chief, guest editors and anonymous reviewers for their comments that help improve the quality of this work.

References

[1] S. M. Othman, F. M. Ba-Alwi, N. T. Alsohybe, and A. Y. Al-Hashida, "Intrusion detection model using machine learning algorithm on Big Data environment," *J. Big Data*, vol. 5, no. 1, 2018, doi: 10.1186/s40537-018-0145-4.

[2] C. Aishwarya*, N. Venkateswaran, T. Supriya, and V. Sreeja, "Intrusion Detection System using KDD Cup 99 Dataset," *Int. J. Innov. Technol. Explor. Eng.*, vol. 4, no. 9, pp. 3169–3171, 2020, doi: 10.35940/ijitee.d2017.029420.

[3] "IDS Security 2022," 2022. https://www.comodo.com/ids-in-security.php

[4] V. Degeler, R. French, and K. Jones, "Self-Healing Intrusion Detection System Concept," in *Proceedings - 2nd IEEE International Conference on Big Data Security on Cloud, IEEE BigDataSecurity 2016, 2nd IEEE International Conference on High Performance and Smart Computing, IEEE HPSC 2016 and IEEE International Conference on Intelligent Data and S*, 2016, no. February 2021, pp. 351–356. doi: 10.1109/BigDataSecurity-HPSC-IDS.2016.27.

[5] T. Wisanwanichthan and M. Thammawichai, "A Double-Layered Hybrid Approach for Network Intrusion Detection System Using Combined Naive Bayes and SVM," *IEEE Access*, vol. 9, pp. 138432–138450, 2021, doi: 10.1109/ACCESS.2021.3118573.

[6] N. Mishra and S. Mishra, "A Review on Big Data Classification : using Machine Learning Technique to Classify Intrusion," *Int. J. Res. Anal. Rev.*, vol. 7, no. 1, pp. 162–165, 2020, [Online]. Available: http://ijrar.org/viewfull.php?&p_id=IJRAR1BAP032

[7] A. Mohammad, A. Maen, K. Szilveszter, and A. Mouhammd, "Evaluation of Machine Learning Algorithms for Intrusion Detection System in WSN," in *International Symposium on Intelligent Systems and Informatics*, 2017, vol. 15, pp. 277–282. doi: 10.1109/SISY.2017.8080566.

[8] H. Shapoorifard and P. Shamsinejad, "Intrusion Detection using a Novel Hybrid Method Incorporating an Improved KNN," *Int. J. Comput. Appl.*, vol. 173, no. 1, pp. 5–9, 2017, doi: 10.5120/ijca2017914340.

[9] W. Wang, M. Zhu, X. Zeng, X. Ye, and Y. Sheng, "Malware traffic classification using convolutional neural network for representation learning," in *International Conference on Information Networking*, 2017, no. January, pp. 712–717. doi: 10.1109/ICOIN.2017.7899588.

[10] G. Zhao, C. Zhang, and L. Zheng, "Intrusion detection using deep belief network and probabilistic neural network," in *Proceedings - 2017 IEEE International Conference on Computational Science and Engineering and IEEE/IFIP International Conference on Embedded and Ubiquitous Computing, CSE and EUC 2017*, 2017, vol. 1, no. May, pp. 639–642. doi: 10.1109/CSE-EUC.2017.119.

[11] B. A. Tama, M. Comuzzi, and K. H. Rhee, "TSE-IDS: A Two-Stage Classifier Ensemble for

Intelligent Anomaly-Based Intrusion Detection System," *IEEE Access*, vol. 7, pp. 94497–94507, 2019, doi: 10.1109/ACCESS.2019.2928048.

[12] R. Jiadong, L. Xinqian, W. Qian, H. Haitao, and Z. Xiaolin, "An Multi-Level Intrusion Detection Method Based on KNN Outlier Detection and Random Forests," *J. Comput. Res. Dev.*, vol. 56, no. 3, pp. 566–575, 2019, doi: 10.7544/issn1000-1239.2019.20180063.

[13] P. Tao, Z. Sun, and Z. Sun, "An Improved Intrusion Detection Algorithm Based on GA and SVM," *IEEE Access*, vol. 6, pp. 13624–13631, 2018, doi: 10.1109/ACCESS.2018.2810198.

[14] Y. Ding and Y. Zhai, "Intrusion detection system for NSL-KDD dataset using convolutional neural networks," *ACM Int. Conf. Proceeding Ser.*, pp. 81–85, 2018, doi: 10.1145/3297156.3297230.

[15] T. Su, H. Sun, J. Zhu, S. Wang, and Y. Li, "BAT: Deep Learning Methods on Network Intrusion Detection Using NSL-KDD Dataset," *IEEE Access*, vol. 8, pp. 29575–29585, 2020, doi: 10.1109/ACCESS.2020.2972627.

[16] Y. Fu, Y. Du, Z. Cao, Q. Li, and W. Xiang, "A Deep Learning Model for Network Intrusion Detection with Imbalanced Data," *Electron.*, vol. 11, no. 898, pp. 1–13, 2022, doi: 10.3390/electronics11060898.

[17] K. Vengatesan, A. Kumar, R. Naik, and D. K. Verma, "Anomaly based novel intrusion detection system for network traffic reduction," in *Proceedings of the International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud), I-SMAC 2018*, 2018, no. August, pp. 688–690. doi: 10.1109/I-SMAC.2018.8653735.

[18] Z. Ahmad, A. Shahid Khan, C. Wai Shiang, J. Abdullah, and F. Ahmad, "Network intrusion detection system: A systematic study of machine learning and deep learning approaches," *Trans. Emerg. Telecommun. Technol.*, vol. 32, no. 1, pp. 1–29, 2021, doi: 10.1002/ett.4150.

[19] S. Sah, "Machine Learning: A Review of Learning Types," *ResearchGate*, no. July, 2020, doi: 10.20944/preprints202007.0230.v1.

[20] E. Burns, "machine learning," *Machine Learning*, 2021.

[21] "machine-learning-algorithms." https://www.javatpoint.com/machine-learning-algorithms

[22] A. Abedinia, "Survey of the Decision Trees Algorithms (CART, C4.5, ID3) | by Aydin Abedinia | Medium," 2019.

[23] https://medium.com/@abedinia.aydin/survey-of-the-decision-trees-algorithms-cart-c4-5-id3-97df842831cd

[24] "machine-learning-decision-tree-classification-algorithm."https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm

[25] "Random Forest Algorithm," 2018. https://www.javatpoint.com/machine-learning-random-forest-algorithm

[26] "logistic-regression-in-machine-learning." https://www.javatpoint.com/logistic-regression-in-machine-learning

[27] "Gradient Descent in Machine Learning." https://www.javatpoint.com/gradient-descent-in-machine-learning

[28] N. Mishra, S. Mishra, and B. Patnaik, "A Novel Intrusion Detection System Based on Random Oversampling and Deep Neural Network," *Indian J. Comput. Sci. Eng.*, vol. 13, no. 6, pp. 1924–1936, 2022, doi: 10.21817/indjcse/2022/v13i6/221306136.

[29] S. J. Stolfo, W. Fan, W. Lee, A. Prodromidis, and P. K. Chan, "KDD-CUP-99 Task Description," 1999.

[30] http://kdd.ics.uci.edu/databases/kddcup99/task.html