Scientific Research Publishing

Journal of Data Analysis and Information Processing, 2021, 9, 249-270
https://www.scirp.org/journal/jdaip
ISSN Online: 2327-7203
ISSN Print: 2327-7211

Case Study on Data Analytics and Machine Learning Accuracy

Abdullah Z. Alruhaymi, Charles J. Kim

Department of Electrical Engineering and Computer Science, Howard University, Washington D.C, USA
Email: azmotairi@hotmail.com, ckim@howard.edu

How to cite this paper: Alruhaymi, A.Z. and Kim, C.J. (2021) Case Study on Data Analytics and Machine Learning Accuracy. *Journal of Data Analysis and Information Processing*, **9**, 249-270.
https://doi.org/10.4236/jdaip.2021.94015

Received: August 30, 2021
Accepted: September 21, 2021
Published: September 24, 2021

Abstract

The information gained after the data analysis is vital to implement its outcomes to optimize processes and systems for more straightforward problem-solving. Therefore, the first step of data analytics deals with identifying data requirements, mainly how the data should be grouped or labeled. For example, for data about Cybersecurity in organizations, grouping can be done into categories such as DOS denial of services, unauthorized access from local or remote, and surveillance and another probing. Next, after identifying the groups, a researcher or whoever carrying out the data analytics goes out into the field and primarily collects the data. The data collected is then organized in an orderly fashion to enable easy analysis; we aim to study different articles and compare performances for each algorithm to choose the best suitable classifies.

Keywords

Data Analytics, Machine Learning, Accuracy, Cybersecurity, Performance

1. Introduction

Data Analytics is a branch of data science that involves the extraction of insights from data to gain a better understanding. It entails all the techniques, data tools, and processes involved in identifying trends and measurements that would otherwise be lost in the enormous amount of information available and always getting generated in the world today. Grouping the dataset into categories is an essential step of the analysis. Then, we go ahead and clean up the data by removing any instances of duplication and errors done during its collection.

In this step, there is also the identification of complete or incomplete data and the implementation of the best technique to handle incomplete data.

The impact of missing values leads to an incomplete dataset in machine learning

(ML) algorithms' performance causes inaccuracy and misinterpretations.

Machine learning has emerged as a problem-solver for many existing situation problems. Advancement in this field helps us with Artificial intelligence (AI) in many applications we use daily in real life. However, statistical models and other technologies failed to remedy our modern luxury and were unsuccessful in holding categorical data, dealing with missing values and significant data points [1]. All these reasons arise the importance of Machine Learning Technology. Moreover, ML plays a vital role in many applications, e.g., cyber detection, data mining, natural language processing, and even disease diagnostics and medicine. In all these domains, we look for a clue by which ML offers possible solutions.

Since ML algorithms do training with part of a dataset and tests with the rest of the other dataset, unless missingness is entirely random and this is rarely happening, missing elements in especially training dataset can alter to insufficient capture of the entire population of the complete dataset. Therefore, in turn, it would lead to lower performance with the test dataset. However, if reasonably close values somehow replace the missing elements, the performance for the imputed dataset would be restored correctly to the level of the same as that of the intact, complete dataset. Therefore, this research intends to investigate the performance variation under different numbers of missing elements and under two other missingness mechanisms, missing completely at random (MCAR) and missing at random (MAR).

Therefore, the objectives of this research dissertation are:

1) Investigation of the data analytic algorithms' performance under full dataset and incomplete dataset.

2) Imputation of a missing element in the incomplete dataset by multiple imputation approaches to make imputed datasets.

3) Evaluation of the algorithms' performance with imputed datasets.

The general distinction between most ML applications is deep learning, and the data itself is the fuel of the process. Data analytics has evolved dramatically over the last two decades. Hence more research is in this area is inevitable. Since its importance, the background of the analytics field and ML is promising and shiny. Much research was conducted on ML accuracy measurements and data analysis, but few were done on incomplete data and imputed data and the comparison outcome. We aim to highlight the results drawn from different dataset versions and the missingness observed in the dataset.

To create an incomplete dataset, we shall impose two types of missingness, namely, MCAR and MAR into the complete dataset. MCAR missingness will be created by inputting N/A into some variable entries to create an impression of data missing by design. MAR missingness will also be generated by inputting N/A into some cells of variables in the dataset. This will create an impression that these incomplete variables are related to some other variables within the dataset that are complete and hence this will bring about the MAR missingness. These incomplete datasets will then be imputed using a multiple imputation by chained equations (MICE) to try and make it complete by removing the two types of missingness. The imputed dataset will then be used to train and test machine learning algorithms. And lastly, the performance of the algorithms with the imputed datasets will be duly compared to performance of the same algorithms with the initially complete dataset.

2. Research Methodology

This article is one chapter of the whole dissertation work, and to achieve the objectives of the dissertation, the following tasks are performed:

1) cyber-threat dataset selection

2) ML algorithms selection

3) investigation and literature review on the missingness mechanisms of MCAR and MAR

4) investigation of imputation approaches

5) application of multiple imputation approaches

6) evaluation of the algorithms under different missing levels (10%, 20%, and 30%) and two missingness mechanisms.

The first two items are discussed below, and the other items are detailed in the succeeding chapters. The proposed workflow of the research is as shown in the following Figure 1.

The methodology for this paper is through analyzing many online internets article and compare for accuracy to use later with the selected dataset performance, and from many algorithms used we select the best that we think are suitable for cybersecurity dataset analysis. Four major machine learning algorithms will be

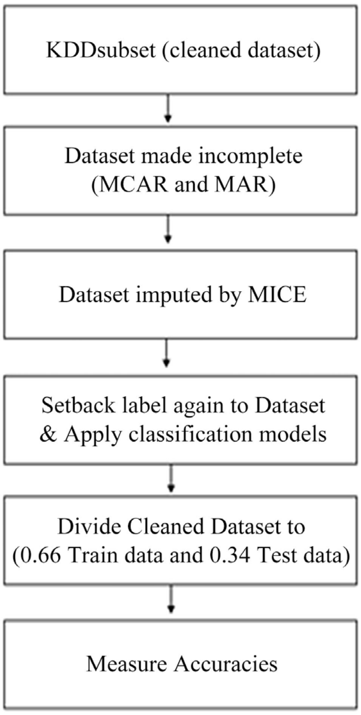

KDDsubset (cleaned dataset)
Dataset made incomplete (MCAR and MAR)
Dataset imputed by MICE
Setback label again to Dataset & Apply classification models
Divide Cleaned Dataset to (0.66 Train data and 0.34 Test data)
Measure Accuracies

Figure 1. Dissertation workflow.

utilized to train and test using portions of the imputed dataset to measure its performance in comparison with the complete dataset. These will include *decision tree*, *random forest*, *support vector machines and naïve bayes*. These will be discussed in depth later in the dissertation. To create the imputed dataset the multiple imputation by chained equations (MICE) method will be used for we consider it a robust method in handling missingness and therefore appropriate for this research. This method ideally fills in the missing data values by using the iteration of prediction models methodology where in each iteration a missing value is imputed by using the already complete variables to predict it.

3. Cyber-Threat Dataset Selection

The dataset selected for this research is KDDsubset dataset from Cyber security Domain, a sample from KDDCUP'99 which consisted of 494,021 records (or instances). As shown in **Figure 2**, Attack types represent more than 80% of the cyber dataset. Denial of Service (Dos) is the most dangerous kind.

It is columned in 42 features and the last feature in the data (42nd) is labeled as either *normal* or *attack* for the different sub-types of attacks (count 22) as shown in **Figure 3** below.

Smurf has the highest count compared to other attacks labelled with only one specific attack in each instance, we can visualize the classes from the figure above that have a different number of attacks and observe that smurf is the most frequent attack. The simulated attacks fall in one of the following four categories: *Denial of Service* (*DoS*), *Probe*, *U2R*, or *R2L*. The column is a connection type and is either an *attack* or *normal*. It is publicly available and widely used in academic research, researchers often use the KDDsubset as a sample of the whole KDDCUP99 dataset which consists of nearly five million records. It covers all attack types and its much easier to make experimental analysis with it. The 41 features are divided into four categories: *basic*, *host*, *traffic*, *and content*. Feature number 2 for example, named protocol consists of only 3 kinds and the most used protocol type is ICMP. Most of its records are of the attack type. As shown below in **Figure 4**.

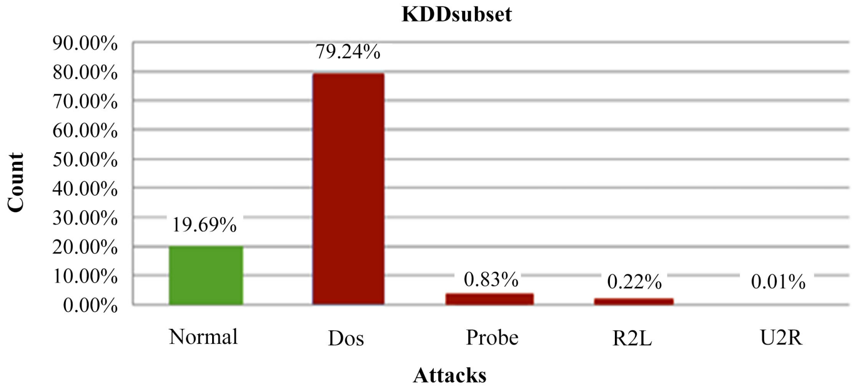


Figure 2. KDDsubset count of attack and normal.

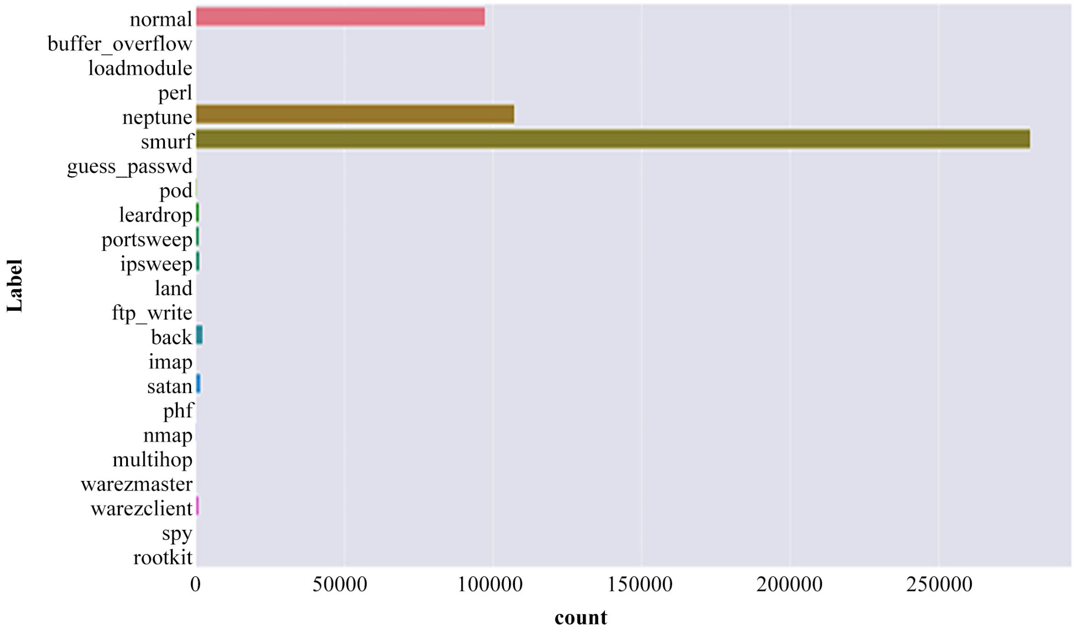


Figure 3. Attack sub-types of count.

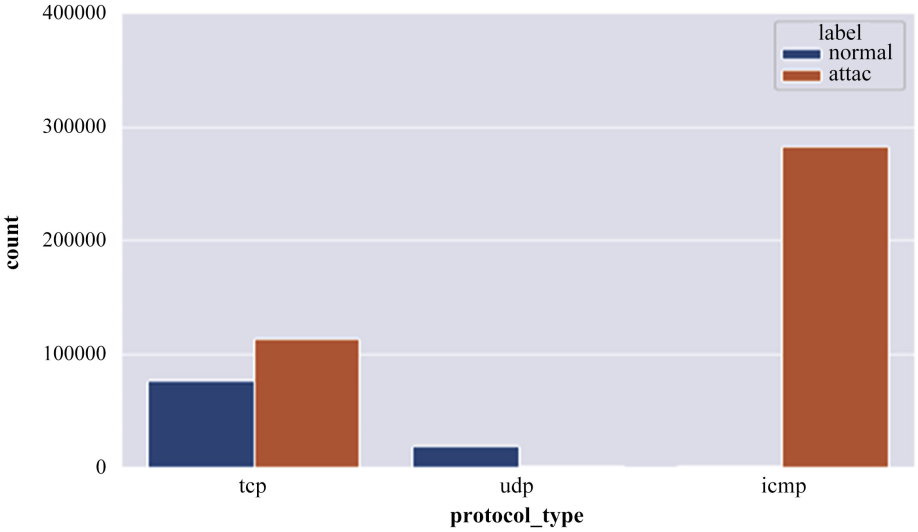


Figure 4. Protocol type has a lot of attacks in the ICMP.

The main problem of the KDDsubset is that it might contain of redundant records which is not ideal when we try to build a data analysis algorithm as it will make the model biased. Cyber security experts have developed advanced techniques to explore technologies that detect cyber-attacks using all of DARPA 1998 dataset used for intrusion detection. Improved versions of this are the 10% KDDCUP'99, NSL-KDD Cup, and *Gure KDDCUP* databases. The KDDCUP'99 dataset was used in the Third International Knowledge Discovery and Data Mining Tools Competition, which was held in conjunction with KDDCUP'99 and the Fifth International Conference on Knowledge Discovery and Data Mining. The competition task was to build a network intrusion detector, a predictive model capable of distinguishing between “bad” connections, called intrusions or attacks, and “good” normal connections. This database contains a standard set of data to be audited, which includes a wide variety of intrusions simulated in a military network environment [2]. The KDDCUP’99 dataset contains around 4,900,000 single connection vectors, every one of which includes 41 attributes and includes categories such as *attack or normal* [3], with precisely one specified attack-type of four main type of attacks:

1) Denial of service (DoS): The use of excess resources denies legit requests from legal users on the system.

2) Remote to local (R2L): Attacker having no account gains a legal user account on the victim’s machine by sending packets over the networks.

3) User to root (U2R): Attacker tries to access restricted privileges of the machine.

4) Probe: Attacks that can automatically scan a network of computers to gather information or find any unknown vulnerabilities.

All the 41 features are also labeled into four listed types:

1) Basic features: These characteristics tend to be derived from packet headers which are no longer analyzing the payload.

2) Content features: Aanalyzing the actual TCP packet payload, and here domain knowledge is used, and this encompasses features that include the large variety of unsuccessful login attempts.

3) Time-based traffic features: These features are created to acquire properties accruing over a 2-second temporal window.

4) Host-based traffic features: Make use of an old window calculated over the numerous connections. Thus, host-based attributes are created to analyze attacks, with a timeframe longer than 2 seconds [4].

Most of the features are of a continuous variable type, where MICE use multiple regression to account for uncertainty of the missing data, a standard error is added to linear regression and in calculating stochastic regression, a similar method of MICE called predictive mean matching was used. However, some variables (is_guest_login, flag, land, etc.) are of binary type or unordered categorical variables (discrete), MICE use a logistic regression algorithm to squash the predicted values between (0 and 1) by using the sigmoid function:

$$f(x) = 1\Big/\left(1+\mathrm{e}^{-x}\right). \tag{1}$$

Below is an illustrative table data of the KDDCUP’99 with 41 features from source: (https://kdd.ics.uci.edu/databases/kddcup99/task.html). As shown in **Table 1**, we remove two features number 20 and 21 because their values in the data are zeros.

The dataset used for testing the proposed regression model is the KDDsubset network intrusion cyber database, and since this dataset is quite a bit large and causes a time delay and slow execution of the R code due to limited hardware equipment, the data is therefore cleaned.

Table 1. Attributes of the cyber dataset total 41 features.

Nr	Name	Description
1	duration	Duration of connnection
2	Protocol_type	Connection protocol (tcp, udp, icmp)
3	service	Dst port mapped to service
4	flag	Normal or error status flag of connection
5	Src_bytes	Number of data bytes from src to dst
6	dst_bytes	Bytes from dst to src
7	land	1 if connection is from/to the same host/port; else 0
8	wrong_fragment	Number of "wrong" fragments (values 0, 1, 3)
9	urgent	Number of urgent packets
10	hot	Number of "hot" indicators
11	number_failed_logins	Number of failed login attempts
12	logged_in	1 if successfully logged in: else 0
13	num_compromised	number of "compromised" conditions
14	root_shell	1 if root shell is obtained; else 0
15	su_attempted	1 if "su root" command attempted; else 0
16	num_root	Number of "root" accesses
17	num_file__creations	Number of file creation operations
18	num_shells	Number of shell prompts
19	num_access_files	Number of operations on access control files
20	num_outbound_cmds	Number of outbound commands in and ftp session
21	Is_hot_login	1 if login belongs to "hot" list; else 0
22	Is_guest_login	1 if login is "guest" login else 0
23	count	number of connections to same host as current connection in the past two seconds
24	srv_count	Number of connections to same service as current connection in the past two seconds
25	serror_rate	% of connections that have "SYN" errors
26	srv_serror_rate	% of connections that have "SYN" errors
27	rerror_rate	% of connections that have "REJ" errors
28	srv_rerror_rate	% of connections that have "REJ" errors
29	same_srv_rate	% of connections to the same service
30	diff_srv_rate	% of connections to different services
31	Srv_diff_host_rate	% of connections to different hosts
32	dst_host_count	Count of connections having same dst host
33	dst_host__srv_count	Count of connections having same des host and using same service
34	des host same srv rate	% of connections having same dst host and using the same servce
35	dst_host_diff_srv_rate	% of different services on current host
36	dst_host_samesrc_port_rate	% of connections to current host having same src port
37	dst_host_srv_diff_host_rate	% of connections to same service coming from diff hosts
38	dst_host_serror rate	% of connections to current host that have an SO error
39	dst_host_srv_serror_rate	% of connections to current host and specified service that have an SO error
40	dst_host_rerror_rate	% of connections to current host that have an RST error
41	dst_host_srv_rerror_rate	% of connections to current host and specified service that have an RST error
42	connection_type	N or A

N = normal, A = attack, c = continuous, d = discrete. Features numbered 2, 3, 4, 7, 12, 14 and 15 are discrete types, and the others are of continuous type.

The used, cleaned dataset has 145,585 connections and 40 numerical features where three features are categorical, which are converted to numeric. For the other 39 features describing the various connections, the dataset is scaled and normalized by letting the mean be zero and the standard deviation be equal to one. But the dataset is skewed to the right because of the categorical variables as shown in Figure 5 below.

To make the data short, we exclude the label variables and add them later for testing by the machine learning algorithms. The dataset is utilized to test and evaluate intrusion detection with both normal and malicious connections labeled as either attack or normal. After much literature review on the same dataset and how the training and testing data are divided, we found that the best method could be letting the training data be 66% and testing data be 34%. The cleaned dataset contains 94,631 connections as training data and 50,954 connections as testing data. The training data employed for missingness mechanisms is the MCAR and MAR. The assumptions are done cell wise, so the total number of cells is 3,690,609 and sampling from it random. The labels were excluded from the training data to perform the missingness identification for two kinds of missingness. Then when the data is fed to the classifier models, we return the labels. The test data is taken from the original clean data and the label left unchanged. So, the test data is tested for all experiments of Machine Learning algorithms. Measurement for the accuracies of the cleaned data is posted before doing any treatment to it for the purpose of comparing missing data and imputed data with MICE with the baseline accuracy looked at for any useful data extracted from the main data analyzed. Now we feed the missing data and the imputed data to the classifiers. We analyze the performance of the four best-chosen classifiers on the dataset. The classifiers selected from literature were considered the best in the evaluation of performance.

4. ML Algorithms Selection

A huge substantial amount of data is available to organizations from a diverse log, application logs, intrusion detection systems, and others. Year over year, the data volume increases significantly and is produced by every person every second, with the number of devices connected to the internet being three times more than the world population.

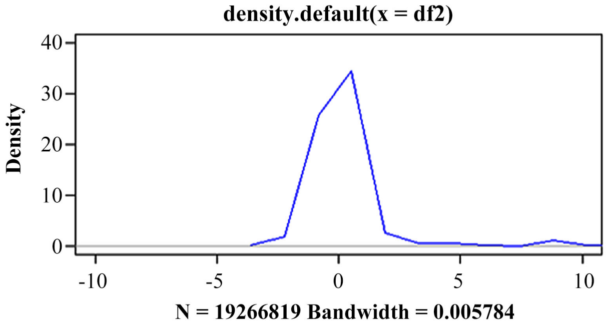


Figure 5. KDDsubset normal distribution.

A large part of which works within the framework of the internet of things (IoT) and these results to more than 600 ZB of data each year.

These facts show the significant development witnessed by the data in terms of type, size, and speed. This vast data is attributed to several reasons. The most important is digitization processes produced by companies and institutions in the past years and the widespread social media and applications of conversations and the internet of things. This growth in various technology fields has made the internet a tempting target for misuse and anomaly intrusion. Many researchers are thus engaged in analyzing the KDDCUP'99 for detecting intrusions. Analysis of the KDDsubset to test for the accuracy and misclassification of the data we find out of the 24 articles reviewed, 13 were straightforward with this dataset. In contrast, the others dealt with a modified versions of NSL-KDD and GureKDD. Some themes that are related directly to the KDDsubset were summarized in an Excel sheet and observed below.

Not all algorithms fail when there is missing data. Some algorithms use the missing value as a unique and different value when building the predictive model, such as classification and regression trees.

Summary review of 10 Articles. To find the most popular Algorithms to apply for the analysis.

Article 1. Summarized as shown in Table 2 below.

No accuracy was posted to this article that uses fuzzy inferences. It uses an effective set of fuzzy rules for inference approach which is identified automatically by making use of the fuzzy rule learning strategy and seems to be more effective for detecting intrusions in a computer system. Then rules were given to sugeno fuzzy system which classify the test data.

Article 2. Summarized as shown in Table 3 below.

Accuracy for this study was low with classifier called rule based enhanced genetic (RBEGEN). This is an enhancement of genetic algorithm.

Article 3. Summarized as shown in Table 4 below.

In this study, 11 algorithms were used, and we can note that we got high accuracies in decision tree and random forest.

Article 4. Summarized as shown in Table 5 below.

Table 2. Fuzzy Inference system.

Article	Classification Technique	Results											
		Accuracy	TT (sec)	prediction	TP	TN	FP	FN	Error rate	Precision	Recall	DR	FAR
1: Intrusion Detection system using fuzzy inference system	Sugenofuzzy inference system for generation of fuzzy rules and best first methods under select	.	.	.	.	.	.	.	.	.	.	.	,

Table 3. Enhance algorithm.

Article	Classification Technique	Results											
		Accuracy	TT (sec)	prediction	TP	TN	FP	FN	Error rate	Precision	Recall	DR	FAR
2: Intrusion Detection over networking KDD dataset using Enhance mining algorithm	Rule based enhance genetic (RBEGEN)	86.30%	.	.	.	.	.	.	.	83.21%	####	.	.

Table 4. Features extraction.

Article	Classification Technique	Results											
		Accuracy	TT (sec)	prediction	TP	TN	FP	FN	Error rate	Precision	Recall	DR	FAR
3: Detecting anomaly based network intrusion using feature extraction and classification techniques	Decision Tree	95.09%	1.032	0.003	4649	2702	279	100	4.9	94.34%	97.34%	.	,
	MLP	92.46%	20.59	0.004	4729		2419	562	29	7.54	89.38	99.56%	
	KNN	92.78%	82.956	13.24	4726	2446	535	23	7.22	89.83%	99.52%		
	Linear SVM	92.59%	78.343	2.11	4723	2434	547	26	7.41	89.62%	99.45%		
	Passive aggressive	90.34	0.275	0.001	4701	2282	699	48	9.66	89.62%	99.45%		
	RBF SVM	91.67%	99.47	2.547	4726	2960	621	23	8.33	89.39%	99.52%		
	Random Forest	93.62%	1.189	0.027	4677	2560	621	23	6.38	91.74%	98.48%		
	AdaBoost	93.52%	29.556	0.225	4676	2553	428	73	6.48	91.61%	98.46%		
	Gausian NB	94.35%	244	0.006	4642	2651	330	107	5.65	93.36%	97.75%	-	-
	MultionmINB	91.71%	0.429	0.001	4732	2357	624	17	8.29	88.35%	99.64%	-	-
	Adratic Discriminat Ana	93.23%	1.305	0.0019	4677	2530	451	72	6.77	91.20%	84.87%	-	-

Table 5. Outlier detection.

Article	Classification Technique	Results											
		Accuracy	TT (sec)	prediction	TP	TN	FP	FN	Error rate	Precision	Recall	DR	FAR
4: Frature Classification and outlier detection to increased accuracy in intrusion detection system.	C45	99.94%	199.33 before 23.14 after	.	.	.	.	.	.	.	.	.	.
	KNN	99.90%	0.37 before 0.23 After	.	.	.	.	.	.	.	.	.	.
	Naïve bayes	96.16%	5.63 before 1.36 after	.	.	.	.	.	.	.	.	.	.
	Random forest	99.94%	554.63 before 205.97 after	.	.	.	.	.	.	.	.	.	.
	SVM	99.94%	699.07 before 186.53 after	.	.	.	.	.	.	.	.	.	.

Three datasets were used in this study and one of them is the KDDCUP'99 to compare accuracy and execution time before and after dimensionality reduction.

Article 5. Summarized as shown in Table 6 below.

Five algorithms were used.

Article 6. Summarized as shown in Table 7 below.

Singulars Values Decomposition (SVD) is eigenvalues method used to reduce a high-dimensional dataset into fewer dimensions while retaining important information and uses improved version of the algorithm (ISVD).

Article 7. Summarized as shown in Table 8 below.

In article number 7 above two classifiers were used and for J48 we have high accuracy results.

Table 6. Three-based data mining.

Article	Classification Technique	Results											
		Accuracy	TT (sec)	prediction	TP	TN	FP	FN	Error rate	Precision	Recall	DR	FAR
5: Intrusion Detection with Three based Data Mining classification techniques by using KDDdataset	Hoeffding Tree	97.05%	.	.	.	.	.	.	2.9499	.	.	.	.
	J48	98.04%	.	.	.	.	.	.	1.9584	.	.	.	.
	Random Forest	98.08%	.	.	.	.	.	.	1.1918	.	.	.	,
	Random Tree	98.03%	.	.	.	.	.	.	1.9629	.	.	.	.
	Req Tree	98.02%	.	.	.	.	.	.	1.9738	.	.	.	.

Table 7. Data reduction.

Article	Classification Technique	Results											
		Accuracy	TT (sec)	prediction	TP	TN	FP	FN	Error rate	Precision	Recall	DR	FAR
6: Using an imputed Data detection method in intrusion detection system	SVD	43.73%	45.154	10.289	43.67%	56.20%	53.33%	43.8	0.5	.	.	.	.
	ISVD	94.34%	189.232	66.72	92.86	.	.	95.82	0.55				

Table 8. Comparative analysis.

Article	Classification Technique	Results											
		Accuracy	TT (sec)	prediction	TP	TN	FP	FN	Error rate	Precision	Recall	DR	FAR
7: Comparative analysis of classification algorithms on KDD'99 Dataset	J48	99.80%	1.8	.	.	.	.	.	.	99.80%	99.80%	.	.
	Naïve bayes	84.10%	47	.	.	.	.	.	.	97.20%	77.20%	.	.

Article 8 and 9. Summarized as shown in Table 9 below.

We have 10 classifuers used to take measurement metrics for the dataset preprocessed and non-preprocessed and results are better with the processed dataset.

Accuracies Percentage for the above Articles:

As shown in Figure 6 below, the percent of accuracy for each classifier is highlighted.

These tables represent a paper that group major attack types and separates the 10% KDDCUP'99 into five files according to the attack types (DoS, Probe, R2L, U2R, and normal). Summarized as shown in Table 10 below.

Based on the attack type, DoS and Probe attacks involve several records, while R2L and U2R are embedded in the data portions of packets and usually involve only a single instance.

The above articles were summarized in measuring metrics, and had nearly 61 classifiers results with the best applicable algorithms.

Table 9. Problems in dataset.

Article	Classification Technique	Results											
		Accuracy	TT (sec)	prediction	TP	TN	FP	FN	Error rate	Precision	Recall	DR	FAR
8: Problems of KDD Cup 99 Dataset existed and data preprocessing	Naïve Bayes	96.31%	6.16	381.45	.	.	.	.	.	.	.	.	.
	Bayes Net	99.65%	43.08	57.38	.	.	.	.	.	.	.	.	.
	Liblinear	99.00%	3557.76	6.1	.	.	.	.	.	.	.	.	.
	MLP	92.92%	22,010.2	25.4	.	.	.	.	.	.	.	.	.
	IBK	100.00%	0.39	79,304.62	.	.	.	.	.	.	.	.	.
	Vote	56.84%	0.17	3	.	.	.	.	.	.	.	.	.
	OneR	98.14%	5.99	6.38	.	.	.	.	.	.	.	.	.
	J48	99.98%	99.43	5.8	.	.	.	.	.	.	.	.	.
	Random forest	100.00%	122.29	4.23	.	.	.	.	.	.	.	.	.
	Random Tree	100.00%	14.79	19.09	.	.	.	.	.	.	.	.	.
9: Problems of KDD Cup 99 Dataset existed and data preprocessing	Naïves Bayes	90.45%	3.21	116.06	.	.	.	.	.	.	.	.	.
	Bayes Net	99.13%	26.65	35.04	.	.	.	.	.	.	.	.	.
	Liblinear	98.95%	708.36	4.27	.	.	.	.	.	.	.	.	.
	MLP	99.66%	11,245.8	53.31	.	.	.	.	.	.	.	.	.
	IBK	100.00%	0.19	48,255.78	.	.	.	.	.	.	.	.	.
	Vote	56.84	0.13	3.45	.	.	.	.	.	.	.	.	.
	OneR	98.98%	3.76	5.35	.	.	.	.	.	.	.	.	.
	J48	99.97%	99.56	5.26	.	.	.	.	.	.	.	.	.
	Random Forest	99.99%	10.89	5.63	.	.	.	.	.	.	.	.	.
	Random Tree	100.00%	10.45	4.26	.	.	.	.	.	.	.	.	.

Table 10. Dataset grouping.

Article	Classification Technique	Class name	Results											
			AA	TT (sec)	Test time (sec)	TP	TN	FP	FN	Error rate	Precision	Recall	DR	FAR
10: Application of Data mining to Network Intrusion detection: Classifier selection model	Bayes Net	Dos	90.62%	628	.	94.60%	.	0.20%	.	.	.	.	.	.
		Probe				88.80%	.	0.12%	.	.	.	.	.	.
		U2R				30.30%	.	0.30%	.	.	.	.	.	.
		R2L				5.20%	.	0.60%	.	.	.	.	.	.
	Naïves Bayes	Dos	78.32%	557	.	79.20%	.	1.70%	.	.	.	.	.	.
		Probe				94.80%	.	13.30%	.	.	.	.	.	.
		U2R				12.20%	.	0.90%	.	.	.	.	.	.
		R2L				0.10%	.	0.30%	.	.	.	.	.	.
	J48	Dos	92.06%	1585	.	96.80%	.	1.00%	.	.	.	.	.	.
		Probe				75.20%	.	0.20%	.	.	.	.	.	.
		U2R				12.20%	.	0.10%	.	.	.	.	.	.
		R2L				0.10%	.	0.50%	.	.	.	.	.	.
	NB Tree	Dos	92.28%	295.88	.	97.40%	.	1.20%	.	.	.	.	.	.
		Probe				73.30%	.	1.10%	.	.	.	.	.	.
		U2R				1.20%	.	0.10%	.	.	.	.	.	.
		R2L				0.10%	.	0.50%	.	.	.	.	.	.
	Decision Table	Dos	91.66%	6624	.	97%	.	10.70%	.	.	.	.	.	.
		Probe				57.60%	.	40%	.	.	.	.	.	.
		U2R				32.80%	.	0.30%	.	.	.	.	.	.
		R2L				0.30%	.	0.10%	.	.	.	.	.	.
	Jrip	Dos	0.923	207.47	.	97.40%	.	0.30%	.	.	.	.	.	.
		Probe				83.80%	.	0.10%	.	.	.	.	.	.
		U2R				12.80%	.	0.10%	.	.	.	.	.	.
		R2L				0.10%	.	0.40%	.	.	.	.	.	.
	One R	Dos	0.8931	375	.	94.20%	.	6.80%	.	.	.	.	.	.
		Probe				12.90%	.	0.10%	.	.	.	.	.	.
		U2R				10.70%	.	2.00%	.	.	.	.	.	.
		R2L				10.70%	.	0.10%	.	.	.	.	.	.
	MLP	Dos	0.9203	350.15	.	96.90%	.	1.47%	.	.	.	.	.	.
		Probe				74.30%	.	0.10%	.	.	.	.	.	.
		U2R				20.10%	.	0.10%	.	.	.	.	.	.
		R2L				0.30%	.	0.50%	.	.	.	.	.	.

As a result of the literature review, and observations, a conclusion was made to select the following four most influential and widely used algorithms which are; decision tree, random forest, support vector machine and naïve bayes. To apply for the dissertation study.

1) Decision tree: [5] This is a flowchart like tree system that is the working area of this classifier, and it is divided into subparts via identifying lines. It uses entropy and information gain to build the decision tree by selecting features that increase the IG and reduce the entropy. Classification and Regression Trees, abbreviated CART is a useful tool that works for classification or regression of the predictive modeling problems.

To build a decision tree model, we follow these steps:

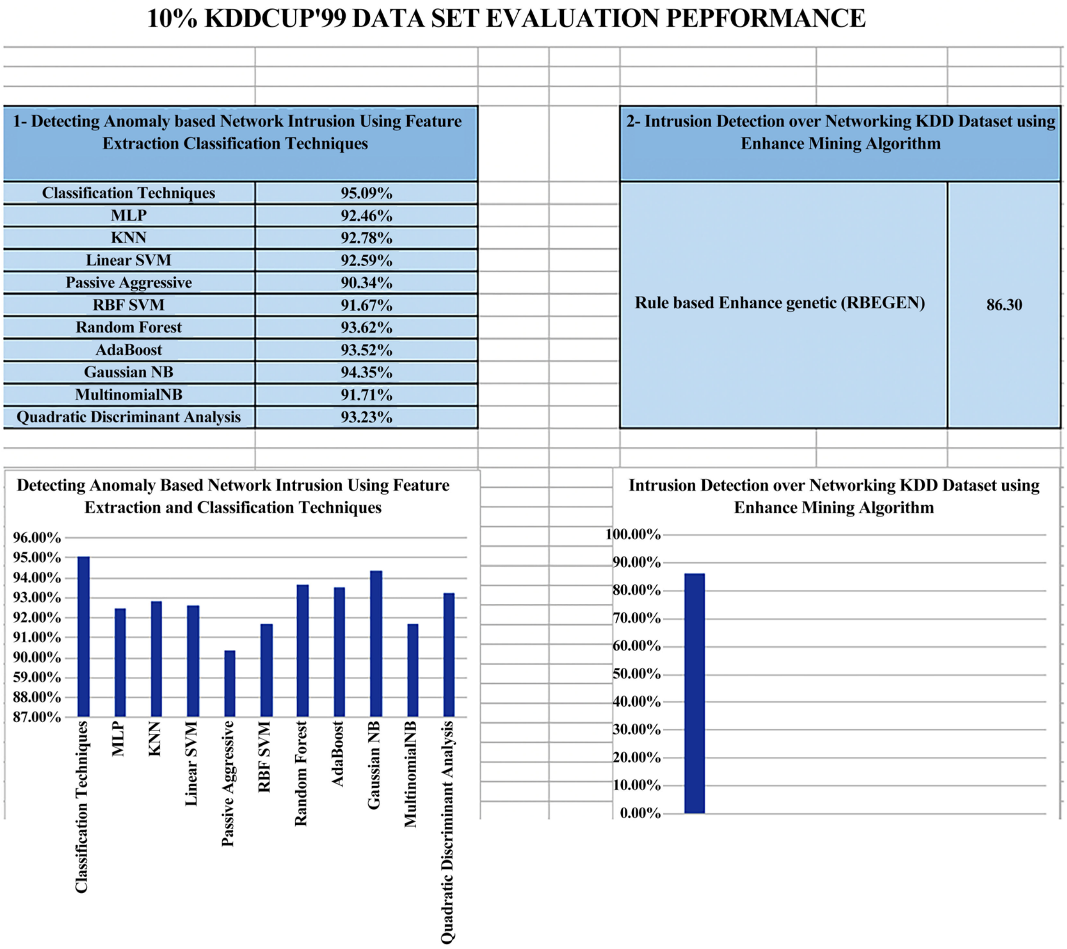
a) The Entropy should be calculated before the splitting for the Target Columns.

b) Select a feature with the Target column and calculate the IG (Information Gain) and the Entropy.

c) Then, the largest information gain should be selected.

d) The selected features are set to be the root of the tree that then splits the rest of the features.

e) The algorithm repeats from 2 to 4 until the leaf has a decision target.

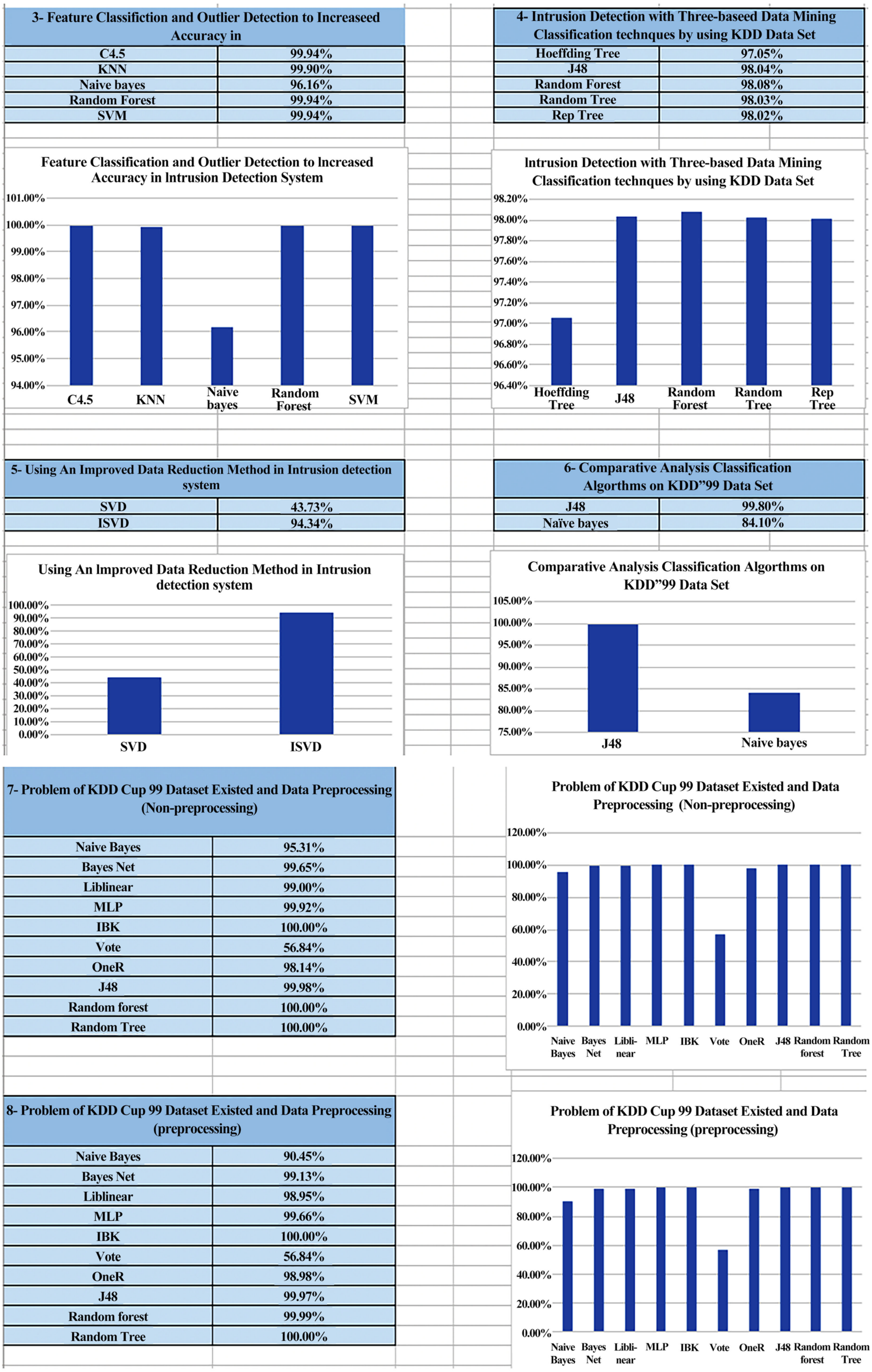


10% KDDCUP'99 DATA SET EVALUATION PEPFORMANCE

1- Detecting Anomaly based Network Intrusion Using Feature Extraction Classification Techniques	
Classification Techniques	95.09%
MLP	92.46%
KNN	92.78%
Linear SVM	92.59%
Passive Aggressive	90.34%
RBF SVM	91.67%
Random Forest	93.62%
AdaBoost	93.52%
Gaussian NB	94.35%
MultinomialNB	91.71%
Quadratic Discriminant Analysis	93.23%

2- Intrusion Detection over Networking KDD Dataset using Enhance Mining Algorithm	
Rule based Enhance genetic (RBEGEN)	86.30

Detecting Anomaly Based Network Intrusion Using Feature Extraction and Classification Techniques
96.00%
95.00%
94.00%
93.00%
92.00%
91.00%
90.00%
59.00%
88.00%
87.00%
Classification Techniques
MLP
KNN
Linear SVM
Passive Aggressive
RBF SVM
Random Forest
AdaBoost
Gaussian NB
MultinomialNB
Quadratic Discriminant Analysis
Intrusion Detection over Networking KDD Dataset using Enhance Mining Algorithm
100.00%
90.00%
80.00%
70.00%
60.00%
50.00%
40.00%
30.00%
20.00%
10.00%
0.00%



3- Feature Classification and Outlier Detection to Increaseed Accuracy in

C4.5	99.94%
KNN	99.90%
Naive bayes	96.16%
Random Forest	99.94%
SVM	99.94%

4- Intrusion Detection with Three-baseed Data Mining Classification technques by using KDD Data Set

Hoeffding Tree	97.05%
J48	98.04%
Random Forest	98.08%
Random Tree	98.03%
Rep Tree	98.02%

Feature Classification and Outlier Detection to Increased Accuracy in Intrusion Detection System

Intrusion Detection with Three-based Data Mining Classification technques by using KDD Data Set

5- Using An Improved Data Reduction Method in Intrusion detection system

SVD	43.73%
ISVD	94.34%

6- Comparative Analysis Classification Algorithms on KDD"99 Data Set

J48	99.80%
Naïve bayes	84.10%

Using An Improved Data Reduction Method in Intrusion detection system

Comparative Analysis Classification Algorthms on KDD"99 Data Set

7- Problem of KDD Cup 99 Dataset Existed and Data Preprocessing (Non-preprocessing)

Naive Bayes	95.31%
Bayes Net	99.65%
Liblinear	99.00%
MLP	99.92%
IBK	100.00%
Vote	56.84%
OneR	98.14%
J48	99.98%
Random forest	100.00%
Random Tree	100.00%

Problem of KDD Cup 99 Dataset Existed and Data Preprocessing (Non-preprocessing)

8- Problem of KDD Cup 99 Dataset Existed and Data Preprocessing (preprocessing)

Naive Bayes	90.45%
Bayes Net	99.13%
Liblinear	98.95%
MLP	99.66%
IBK	100.00%
Vote	56.84%
OneR	98.98%
J48	99.97%
Random forest	99.99%
Random Tree	100.00%

Problem of KDD Cup 99 Dataset Existed and Data Preprocessing (preprocessing)

Figure 6. Accuracies for different algorithms depicted.

In short, entropy measures the homogeneity of a sample of data. The value is between zero and one, zero only when the data is completely homogenous, and one only when data is non-homogenous.

$$\text{Entropy} = \sum_{i=1}^{c} -p_i * \log_2\left(p_i\right) \tag{2}$$

Information gain measures the reduction in entropy or surprise by splitting a dataset according to a given value of a random variable. A larger information gain suggests a lower entropy group or groups of samples, and hence less surprise.

$$IG\left(D_P, f\right) = I\left(D_P\right) - \frac{N_{\text{left}}}{N} I\left(D_{\text{left}}\right) - \frac{N_{\text{right}}}{N} I\left(D_{\text{right}}\right) \tag{3}$$

where:

f: feature split on

D_p: dataset of the parent mode

D_{left}: dataset of the left child node

D_{right}: dataset of the right child node

I: impurity criterion (Gini index or Entropy)

N: total number of samples

N_{left}: number of samples at left child node

N_{right}: number of samples at right child node [6].

Figure 7 below indicates how the Decision Tree works.

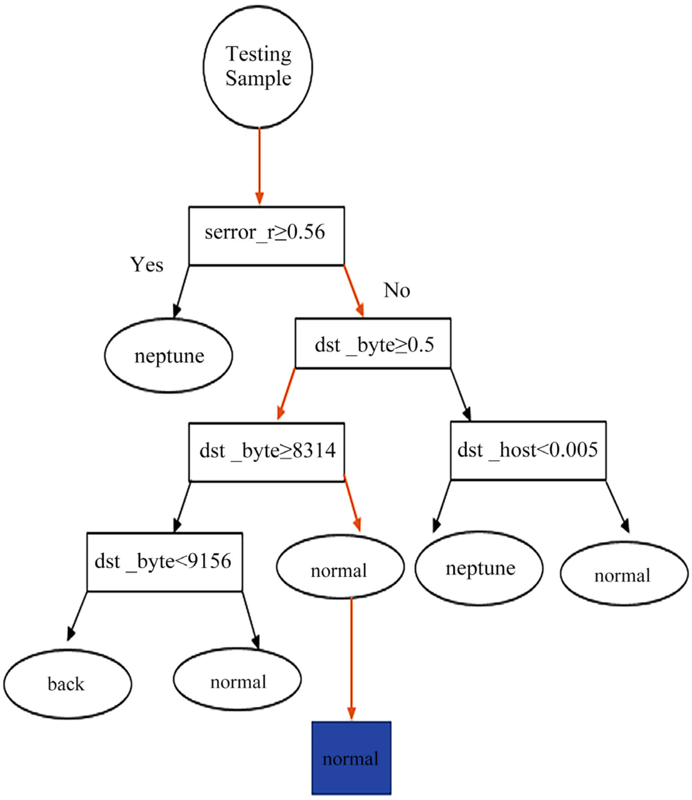


Figure 7. Decision Tree algorithm.

2) Random Forest: [7] The random forest algorithm is a supervised classification algorithm like a decision tree and instead of one tree, this classifier uses multiple trees and merges them to obtain better accuracy and prediction. In random forests each tree in the ensemble is built from a sample drawn with a replacement from a training set called bagging (Bootstrapping) and this improves stability of the model. Figure 8 below shows the mechanism of this algorithm.

3) Support vector machines (SVM): [8] SVM is memory efficient and uses a subset of training points in the decision. It is a set of supervised machine learning procedures used for classification, regression, and outlier detection. Different kernel functions can be specified for a decision function—example of supervised learning algorithms which belong to both the regression and classification categories of machine learning algorithms. Regarding limitations of data dimensionality and limited samples, this classifier does not suffer from these mentioned limitations.

It contains three functions linear, polynomial, and sigmoid and so the user can select any one of the functions to classify the data.

Kernel Functions

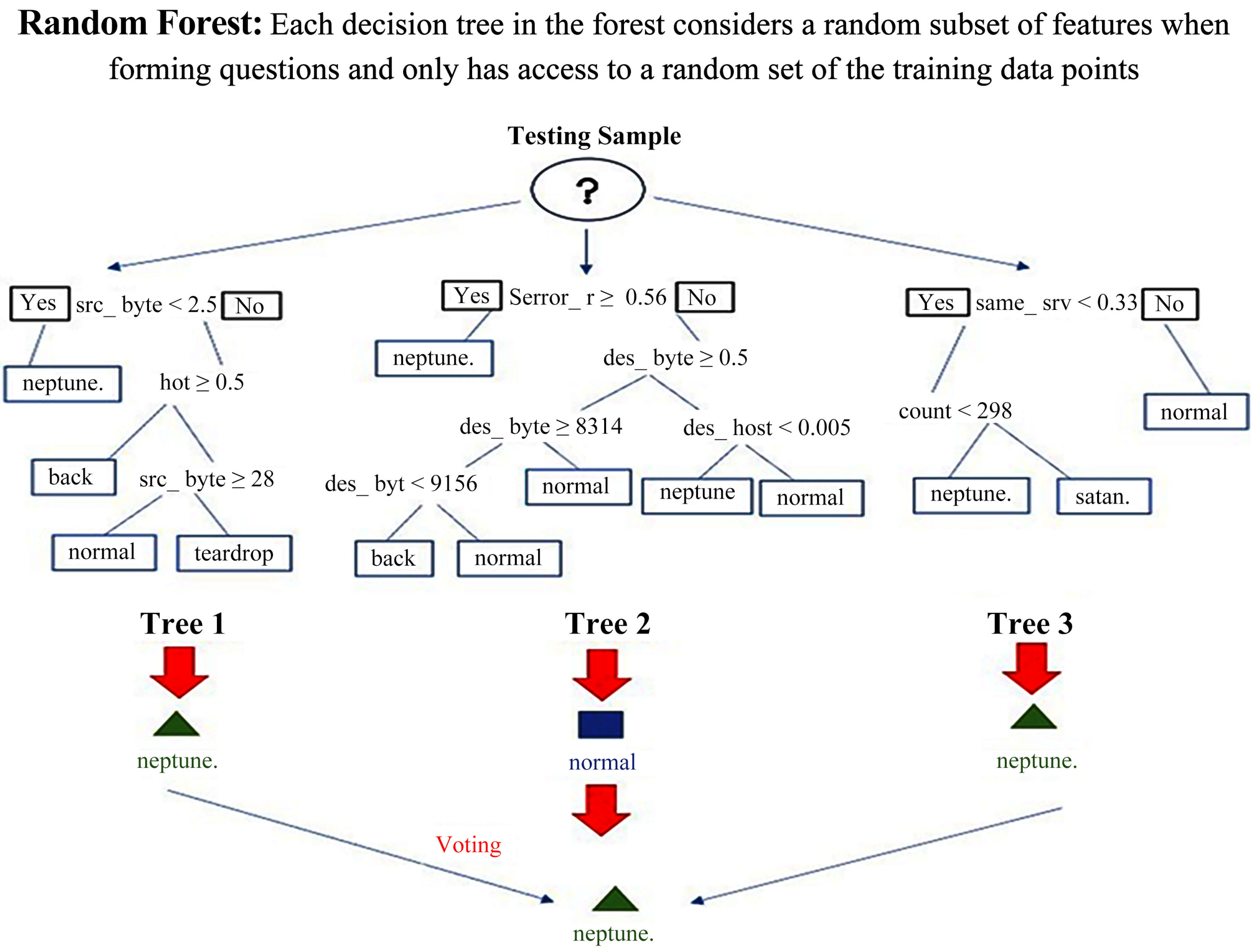


Figure 8. Three decision trees consist Random Forest model.

One of the most important features of SVM is that it utilizes the kernel trick to extend the decision functions to be used for linear and non-linear cases.

Linear Kernel. It can be used as a dot product between any two observations and it's the simplest kernel function.

$$k(x,y)=x^{\mathrm{T}}\cdot y \tag{4}$$

Polynomial kernel: It is a more complex function that can be used to distinguish between non-linear inputs. And it can be represented as:

$$k(x,y)=\left(x^{\mathrm{T}},y\right)^{p} \quad \text{or} \quad k(x,y)=\left(x^{\mathrm{T}},y+1\right)^{p} \tag{5}$$

where p is the polynomial degree. Radial basis function is a kernel function that helps in non-linear cases, as it computes the similarity that depends on the distance between the origin or from some points:

RBF (Gaussian):

$$\phi(x)=\exp\left(-x^{2}/\left(2\sigma^{2}\right)\right),\ \sigma>0 \tag{6}$$

Figure 9 below shows how SVM work.

4) Naïve Bayes: [10] Naïve Bayes classifiers are a family of simple "probabilistic classifiers" based on applying bayes' theorem with strong (naïve) independence assumptions between the features. This classifier provides a simple approach, with precise semantics, to represent and learn probabilistic knowledge.

Naïve Bayes works on the principle of conditional probability as given by the bayes theorem and formulates linear regression using probability distribution.

The posterior probability of the model parameters is conditional upon the training inputs and outputs: The aim is to determine the posterior distribution for the model parameters. The Bayes Rule is shown in Equation (7) below:

$$P(A\mid B)=\frac{P(B\mid A)P(A)}{P(B)} \tag{7}$$

$$y\sim N\left(B^{\mathrm{T}}X,\sigma^{2}I\right) \tag{8}$$

$$P(\beta\mid y,X)=\frac{P(y\mid\beta,X)*P(\beta\mid X)}{P(y\mid X)} \tag{9}$$

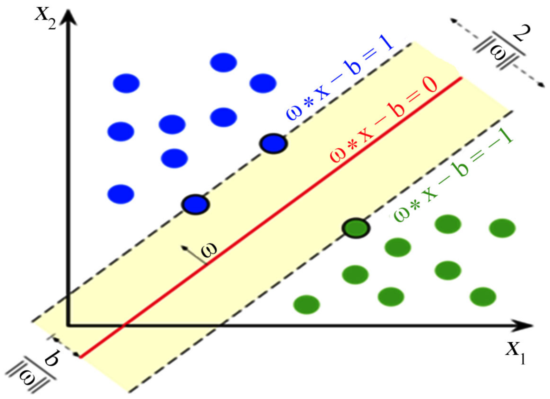


Figure 9. Support Vector Machine algorithm [9].

5. Accuracy of Machine Learning

The aim is not only to hypothesize and prove them. But also, to answer the following scientific questions; first, whether scientific inquiry is about missing data and how the classifiers perform. Which one works better? and how missingness impacts Machine Learning Performance? Does the accuracy increase or decrease if we impute data correctly with the correct imputation number? Can the restoration of the missing data accuracy be one metric for evaluating the performance? The number of instances a model can predict correctly.

Accuracy is an essential measurement tool that measures values from the experiment, how close to the actual value? How relative a measured value is to the "true" value, and precision of how close the data is to each other?

Although the accuracy in some models may be achieved in high results, it is still not significantly accurate enough, and we need to check for other factors.

Because the model might give good results and still be biased toward some frequent records, some classification algorithms will also consume extended hours or even more to get the required products. Certainly, classification accuracy alone can be misleading if we have an unequal number of observations in each class or more than two classes in the dataset. Suppose we import specific libraries to solve the mentioned problem above by balancing types called oversampling or down sampling. In that case, that means the library will create equal samples fed to the ML algorithm to classify. Also, the best evaluation metrics for such is the confusion Matrix; as shown in **Figure 10** below, is it appropriate enough for data to be analyzed? A confusion matrix is a technique for summarizing the performance of a classification algorithm.

Accuracy = (correctly predicted class/total testing class) × 100%, the accuracy can be defined as the percentage of correctly classified instances

Acc = (TP + TN)/(TP + TN + FP + FN).

Where TP, FN, FP, and TN represent the number of true positives, false negatives, false positives, and true negatives, respectively.

Also, you can use standard performance measures:

Sensitivity = TP/TP + FN; R = TP/(TP + FN)

Specificity = TN/TN + FP

Precision = TP/TP + FP; P = TP/(TP + FP)

True-Positive Rate = TP/TP + FN

False-Positive Rate = FP/FP + TN

True-Negative Rate = TN/TN + FP

False-Negative Rate = FN/FN + TP

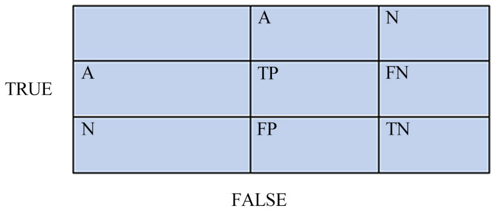
(T for true and F for false, N for negative and P for positive).

For good classifiers, TPR and TNR both should be nearer to 100%. Similar is the case with precision and accuracy parameters.

On the contrary to: FPR and FNR both should be as close to 0% as possible.

Detection Rate (DR) = number of instances notified/total number of instances estimated.

FPR = FP/ALL POSITIVE; TPR = TP/ALL NEGATIVE



TRUE

	A	N
A	TP	FN
N	FP	TN

FALSE

Figure 10. Confusion Matrix for attack (A) and normal (N).

5.1. On Missing Data

The more accurate the measurement, the better it will be. The missing values are imputed with best guesses and used to work if the missing values are small then drop the records with missing values if the data is large enough. However, this was the case before the multivariate approach, but now this is not the case anymore.

Accuracy with missing data and because all the rows and columns are of numerical values so, when we make the data incomplete with R code, the replacement for empty cells is done using NA.

And the classifiers may not work with the NA, instead, we can substitute with mean for each variable or median or mode or just with constant number -9999 and we run the code and this works with a constant number. We conclude that the outcome of accuracy may decrease, or maybe some algorithms will not respond.

5.2. On Imputed Data

We assume that with reasonable multivariate imputation procedures, the accuracy will be close enough to the baseline accuracy of the original dataset before we make it incomplete, then we impute for both mechanisms. The results will be shown in chapter 5.

6. Conclusion

This paper provides a survey of different machine learning techniques for measuring accuracy for different ML classifiers used to detect intrusions in the KDDsubset dataset. Many algorithms have shown promising results because they identify the attribute accurately. The best algorithms were chosen to test our dataset and results posted in a different chapter. The performance of four machine learning algorithms has been analyzed using complete and incomplete versions of the KDDCUP'99 dataset. From this investigation it can be concluded that the accuracy of these algorithms is greatly affected when the dataset containing missing data is used to train these algorithms and they cannot be relied upon in solving real world problems. However, after using the multiple imputation by chained equation (MICE) to remove the missingness in the dataset the accuracy of the four algorithms increases exponentially and is even almost equal to that of the original complete dataset. This is clearly indicated by the confusion matrix in section 5 where TNR and the TPR are both close to 100% while the FNR and FPR are both close to zero. This paper has clearly indicated that the performance of machine learning algorithms decreases greatly when a dataset contains missing data, but this performance can be increased by using MICE to get rid of the missingness. Some classifiers have better accuracy than others, so we should be careful to choose the suitable algorithms for each independent case. We conclude from the survey and observation that the chosen classifiers work best with cybersecurity systems, while others are not and may be helpful in different domains. A Survey of many articles provides a beneficial chance for analyzing detecting attacks and offers an opportunity for improved decision-making in which model is the best to use.

Acknowledgements

The authors would like to thank the anonymous reviewers for their valuable suggestions and notes; thanks, extended to Scientific Research/Journal of Data Analysis and Information Processing.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Fatima, M. and Pasha, M. (2017) Survey of Machine Learning Algorithms for Disease Diagnostic. *Journal of Intelligent Learning Systems and Applications*, **9**, 1-16. https://doi.org/10.4236/jilsa.2017.91001

[2] Kim, D.S. and Park, J.S. (2003) Network-Based Intrusion Detection with Support Vector Machines. In: *International Conference on Information Networking*. Springer, Berlin, Heidelberg, 747-756. https://doi.org/10.1007/978-3-540-45235-5_73

[3] Tavallaee, M., Bagheri, E., Lu, W. and Ghorbani, A.A. (2009) A Detailed Analysis of the KDD CUP 99 Data Set. 2009 *IEEE Symposium on Computational Intelligence for Security and Defense Applications*, Ottawa, 8-10 July 2009, 1-6. https://doi.org/10.1109/CISDA.2009.5356528

[4] Sainis, N., Srivastava, D. and Singh, R. (2018) Feature Classification and Outlier Detection to Increased Accuracy in Intrusion Detection System. *International Journal of Applied Engineering Research*, **13**, 7249-7255.

[5] Sharma, H. and Kumar, S. (2016) A Survey on Decision Tree Algorithms of Classification in Data Mining. *International Journal of Science and Research* (*IJSR*), **5**, 2094-2097. https://doi.org/10.21275/v5i4.NOV162954

[6] Singh, S. and Gupta, P. (2014) Comparative Study ID3, Cart and C4. 5 Decision Tree Algorithm: A Survey. *International Journal of Advanced Information Science and Technology* (*IJAIST*), **27**, 97-103.

[7] Chen, J., Li, K., Tang, Z., Bilal, K., Yu, S., Weng, C. and Li, K. (2016) A Parallel Random Forest Algorithm for Big Data in a Spark Cloud Computing Environment. *IEEE Transactions on Parallel and Distributed Systems*, **28**, 919-933. https://doi.org/10.1109/TPDS.2016.2603511

[8] Suthaharan, S. (2016) Support Vector Machine. In: *Machine Learning Models and Algorithms for Big Data Classification*. Springer, Boston, 207-235. https://doi.org/10.1007/978-1-4899-7641-3_9

[9] Larhman (2018) Linear Support Vector Machines. https://en.wikipedia.org/wiki/Support-vector_machine

[10] Chen, S., Webb, G.I., Liu, L. and Ma, X. (2020) A Novel Selective Naïve Bayes Algorithm. *Knowledge-Based Systems*, **192**, Article ID: 105361. https://doi.org/10.1016/j.knosys.2019.105361