Journal of Information Technology Management

Print ISSN: 2008-5893

Online ISSN: 2423-5059

Machine Learning Algorithms Performance Evaluation for Intrusion Detection

Shyla*

*Corresponding Author, Ph.D., Department of Computer Science and Engineering, NSUT East Campus, Ambedkar Institute of Advanced Communication Technologies and Research, GGSIPU, India. E-mail: shylasinghit@gmail.com

Kapil Kumar

M.Tech., Department of Computer Science and Engineering, NSUT East Campus, Ambedkar Institute of Advanced Communication Technologies and Research, GGSIPU, India. E-mail: kapil.sharma0942211@gmail.com

Vishal Bhatnagar

Professor, Department of Computer Science and Engineering, NSUT East Campus, Ambedkar Institute of Advanced Communication Technologies and Research, India. E-mail: vishalbhatnagar@yahoo.com

Abstract

The steadily growing dependency over network environment introduces risk over information flow. The continuous use of various applications makes it necessary to sustain a level of security to establish safe and secure communication amongst the organizations and other networks that is under the threat of intrusions. The detection of Intrusion is the major research problem faced in the area of information security, the objective is to scrutinize threats or intrusions to secure information in the network Intrusion detection system (IDS) is one of the key to conquer against unfamiliar intrusions where intruders continuously modify their pattern and methodologies. In this paper authors introduces Intrusion detection system (IDS) framework that is deployed over KDD Cup99 dataset by using machine learning algorithms as Support Vector Machine (SVM), Naïve Bayes and Random Forest for the purpose of improving the precision, accuracy and recall value to compute the best suited algorithm.

Keywords: Intrusion Detection System, Naïve Bayes, Random Forest, Support Vector Machine.

DOI: 10.22059/jitm.2021.80024

Introduction

The intrusion detection systems (IDS) played a vital role in assuring security of the network from varied intrusions. An IDS can detect unauthorized access of systems from network attacks and take effective preventive measures on the basis of required security. The security providing centres manage IDS and computer hosts for attack counter measures. (Vinayakumar et al., 2019) found that there are many commercial IDS available and most of these commercial implementation are ineffective and insufficient, which introduces the need of more research on dynamic IDS. There are several Network Intrusion Detection Systems (NIDS) for precisely observing data flow to recognize intrusion in any network. The two main approaches of IDS are Misuse based IDS and Anomaly based IDS.

(Singh, Kalra & Solanki., 2019) found that in Misuse based IDS, the existing attacks that is already known holds a particular signature. The signature is based on different network packets patterns and data flow characteristics. The signatures that is used for known attacks is compared to the data flow patterns for intrusion detection. The misuse based IDS have tremendous precision, less false positives and accuracy. The misuse based IDS is unable to observe undetermined and new intrusions. The another form is Anomaly based IDS. (Xu et al., 2018) found that in this detection approach if a model detect any auspicious behaviour that is different from existing activities then any deviation from normal behaviour is considered as attack. Anomaly-based detection is preferred over signature based IDS because anomaly based have an ability to detect unknown attacks. The anomaly based IDS widely used neural networks, clustering, deep neural networks and k-means for supervised machine learning. The unsupervised machine learning techniques made their learned and tested models depending on factors as reliability, consistency and confidentiality among several activities.

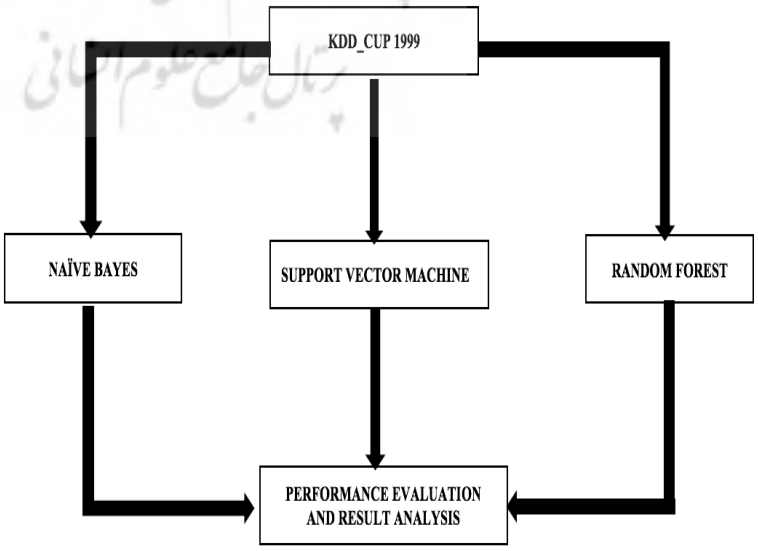

Figure 1. IDS Framework (Xu et al.,2018)

Figure 1 shows the IDS framework to evaluate the optimal performance by using various classifier for improving the precision and accuracy of IDS. The classification machine learning algorithms are used to classify the data into three phases where the first is pre-processing phase in which data is sensed, as every dataset is not readable. (Koli and Chavan, 2017) found that the first and most crucial step is to sense the information correctly and if the information is text categorical then convert this categorical data into binary categorical data in case of classification. The model does fast execution with binary data and accept the data to train precisely.

(Al-Jallad, Aljnidi & Desouki., 2019) found that next phase is to find the different categories for classification, if categories are limited according to the classification mode then proceed but if category is more standard then performance would get decrease and result would become inaccurate.

To overcome the issue, the merging of sample classes into limited category on the basis of their feature is the best methodology. It helps reducing the dimensions and increasing the performance. It is necessary to sense data about balanced or unbalanced classes and to perform the basic operation of pre-processing.

In pre-processing phase, authors are using data set of KDD Cup99 for all model and then pre-process the dataset by splitting the dataset into training and testing dataset and then transform training data into standard form and after that the training phase is used to apply training data to model and analyse the outcomes.

The Machine learning algorithms is defined as supervised learning and unsupervised learning. (Xu et al., 2018) found that the IDS models is built by using supervised Learning algorithms based on signature-based approach, where the attacks are already known and have perquisite signatures for training datasets using support vector machines (SVM), linear regression, Naive Bayes, logistic regression, random forest, linear discriminant analysis, decision trees, and neural networks. These algorithms are majorly used as supervised learning algorithms to model signature-based IDS. The training datasets and parameters defines accuracy of signature-based machine learning IDS.

In last two decades, information technology is growing rapidly and computer security becomes an essence for industry, business and various fields.

Problem Statement

The available IDS is based on misuse IDS, which means the detection of available intrusions is unable to detect unknown intrusions. For solving the issues related to misuse based IDS the concept of anomaly based IDS is introduced for detection of newly emerging intrusions. The problem associated with existing security system is the behaviour of intrusions that changes

periodically and require re-training of the systems. The system can predict the abnormal behaviour as normal if training set holds intrusion. To overcome the existing issues the following objectives has been made .

Objectives

- The objectives of the study is to develop an improved IDS for detecting attacks. The objectives include:
- The Machine Learning algorithms SVM, Naïve Bayes and Random Forest are used to detect intrusions.
- The accuracy and performance of algorithms are compared.
- Design a machine learning system for detecting and tracking of attacks.
- Validation and verification of IDS is made by deploying machine learning algorithms using python programming language and KDD Cup99 dataset.

The remainder phases are arranged as Section 1 that includes the introduction of IDS. Section 2 defines literature review. The Section 3 shows process framework. The Section 4 shows framework deployment. The section 5 shows learning algorithm computations .The Section 6 includes performance analysis. The Section 7 shows research limitations and Section 8 includes conclusion and future scope.

Literature Review

The information security in each and every sector makes it necessary to build a system that provides a high security, safe and reliable communication among various organization, assets and communication over the internet and other network under the threat of intrusion and misuses. (Xu et al., 2018) uses two different datasets one is Modbus-based gas pipeline control traffic and another is OPCUA-based batch processing traffic to detect attacks by using SVM and Random Forest machine learning algorithms. Authors found that the accuracy of SVM is 92.53 percent with execution time 11712 seconds and Random Forest is 99.84 percent with execution time of 281 second for dataset one and for data set two the Accuracy of SVM is 90.81 percent with execution time of 0.019 seconds and for Random forest accuracy is 99.98 percent with execution time of 52.31 seconds.

(Almseidin et al., 2017) authors conducted various experiments to test and evaluate the performance and efficiency of the machine learning algorithms as Naive Bayes, Random Forest, J48, Random Tree and Decision Tree. These classifiers depends on KDD dataset and found that Random forest have the maximum accuracy rate of 93.77 percent. (Singh, Kalra &

Solanki., 2019) authors reviewed IDS based hybrid approach by using machine learning algorithms as SVM and KNN for future extraction and classification of data.

(Mohammadi & Namadchian, 2017) used several methods to perform experiment to determine performance of machine learning algorithms by using KDD dataset. Results showed that the minimum value of false negative is obtained by decision table classifier and the highest accuracy and precision value is obtained by random forest classifier with an accuracy of 93.77 percent and minimum false positive rate.

(Abubakar & Pranggono, 2017) authors presents machine learning IDS for SDN. Authors used signature and flow-based IDS to detect intrusions in the defined software. Pattern recognition accuracy is compared with other neural network models. The drawbacks of signature based IDS is overcome by using the flow-based anomaly technique with machine learning. The results show 97 percent accuracy of the trained model. (Yin et al., 2017) authors used KDD Cup99 dataset will be used to determine best algorithm. The implemented experiments show that SVM can be a useful tool for IDS and found that SVM achieved 94.43 percent average accuracy rate.

(Vinayakumar et al., 2019) authors showed graph based semi supervised clustering technique and a precise outlier detection approach that is used for intrusion problem in hybrid framework. Authors proposed multistage system by using machine learning algorithms to construct IDS that is foremost important implication in the information security. The experiments performed on fetched datasets shows the improvement of introduced methods. (Aljawarneh et al., 2018) authors proposed a hybrid model to determine the intrusion threshold value by using optimal features and network transaction for training. Authors found accuracy of 99.81percent for binary classification and 98.56 percent for multi class NSL-KDD datasets.

Existing Issues

- First problem is related to false classification of attacks that shows low accuracy and precision, (Yin et al., 2017) found that this is because of unbalanced classes and a large number of category of attacks classes that is based on same features.

- Second, problem is related to placement of IDS, the system obtained the information from the payload of packets on a network. (Mohammadi & Namadchian, 2017) found that the information can be modified by attacker, as the information takes time to reach the IDS, if collected data are modified then result would be different.

- Third, problem is related to low detection rate as it takes analysis time to detect whether it is intrusion or normal attack, (Yin et al., 2017) found that this is because number of attack category is large while training the model.

Process Framework

The process framework is proposed by authors to define the procedure followed by using machine learning algorithms for IDS. Authors used KDD Cup99 dataset for analysis. The dataset is then preprocessed by using python programming language where data ambiguity is removed. The dataset is divided into independent and dependent variable and if duplicate value exist then drop them.

In pre-processing phase, the authors used dataset in csv file format and import it for data transformation by transferring the data into data frame to make it easy to read. Then the redundancy in dataset is removed and after removing the redundancy authors used binary encoder to convert all the text categorical data into binary data, as the model of machine only accept the data in numeric and binary form and binary form makes the execution fast and another benefit is to resolve the problem of columns that contains the large number of data categories. The dataset is then splits into independent and dependent data. The total columns were fifty-five, independent data contains fifty-four columns and target variable or dependent data was the last column after extracting the data the next phase was to find the missing value if exist, then replace them by mean. In next phase of pre-processing the data splits into training and test set in the ratio of (75:25) for cross validation. In next phase, the training data set is converted into a standard form, but dimensions were large with forty one columns.

In next phase the principle component analysis algorithm is used to get principle components by reducing the dimensions. In training phase, authors used the result of principle component analysis to train the model. In prediction phase, result of training model, would be used with test data to found the prediction or classification of attacks after that confusion matrix is required to compute the outcomes of the proposed methodology.

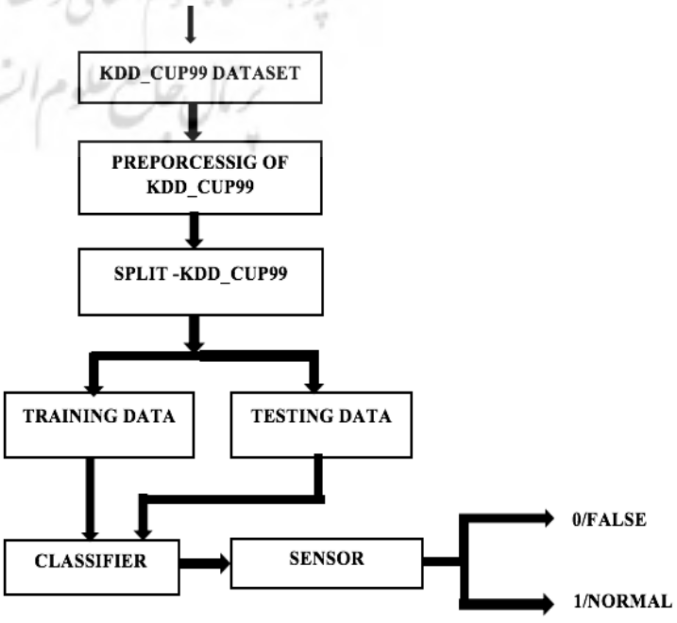

Figure 2. Proposed IDS Framework
(Yin et al., 2017)

Figure 2 shows proposed IDS framework for the operation of a machine learning algorithms with intrusion detection where KDD Cup 99 dataset is observed to generate classifiers from dataset training and testing. These classifiers generate sensor result as false or normal values.

Framework Deployment

The proposed framework is implemented by using proposed algorithm and to evaluate the performance of system authors used the standard dataset as KDD Cup99 for intrusion detection. The general steps of algorithm of proposed system can be divided into three phases as pre-processing phase, training phase and prediction phase.

- ***Pre-processing phase***

Importing the dataset.

Import<KDD Cup99>

If dataset is not in suitable form then transforming the dataset into a suitable data frame.

dataset->data frame.

Splitting of the dataset into independent and dependent variables.

X<-dataset and Y<-dataset.

If duplicate element exists then drop.

If text categories exist then transform categorical data from text data to binary.

binary encoder->X[columns].

If missing value exist then replace missing values with mean.

missing value<-simpleImputer(missingvalue<-np.nan , strategy->mean).

missingvalue->missingvalue.fit(X).

The dataset is divided into training set and testing set having test size of =0.25.

X_train,Y_train,X_test,Y_test<-split(X,Y,0.25).

If dataset is large then do dimensional reduction.

pca->pca(no of component) and pca.fit(X_train).

- ***Training Phase***

Use preprocessed data.

Fitting the independent and dependent training set into model.

Classifier(X_train, Y_train).

Execute.

Result.

- ***Prediction Phase***

Use the preprocesses test data.

Predict the Outcomes based on passing the independent testing set to classifier.

Classifier->Predict(X_test).

Data Analytics Tools

Authors used Python with Pandas including pandas data frames and libraries to develop the IDS by using machine learning aspects. Python and Pandas is widely used in various areas including data analytics, economics and statistics. Pandas is the licensed open-source Python library which shows the better performance and user friendly information analytical tool for the Python programming language. Pandas library uses NumPy functionalities. The most important libraries in Python is NumPy and Pandas.

- **Pandas:** Pandas is an open source system built over NumPy for providing high accuracy, performance, data analysis and data structure tools for Python programming language. This provides fast data analysis, transformation and visualisation. Pandas is best for different types of data as tabular data, matrix data, time series data, with multiple row and columns, and for other form of statistical data.

- **NumPy:** NumPy act as a library for scientific computing in Python. It is required to obtain tools for working with multidimensional array objects with high performance. (Xu et al., 2018) defines that NumPy arrays can be of two types vectors and matrices where Python vectors are one dimensional array and Python Matrices are two dimensional arrays with only one row and column.

- **Pandas Data frames:** Pandas data frames is used to create a data structure with axes labelled as rows and columns for tabular data. Default format of a data frame can be pd.Dataframe(data, index, column). Authors need to declare the data, columns and index value to create a data frame. (Al-Jallad, Aljnidi & Desouki., 2019) found that in Pandas data frames it is preferred to have at least two-dimensional data where index defines the row name and columns values.

Data Description

Authors used KDD Cup99 dataset that is widely used for the evaluating anomaly based IDS. This 1999 data set includes huge variety of intrusions captured in network by considering data from DARPA'98 IDS program. The KDD Cup99 training dataset introduces 4,900,000 single connection vectors which holds labelled 41 features as attack or secured with one specific intrusion type. The intrusions will fall in one of the following categories:

- **Denial of Service Attack (DoS):** (Aljawarneh et al., 2018) found that DoS is an attack where intruders made memory and resources occupied to execute existing requests and reject legitimate requests to the system.

- **User to Root Attack (U2R):** The attack where intruders obtains unfortified access of user the system and is vulnerable to deuterate and exploit user authenticity and integrity to obtain system access by passwords sniffing, phishing and malware.

- **Remote to Local Attack (R2L):** It is an attack where attacker gains the ability to transmit information to a user system over a TCP/IP network and deuterate system susceptibility to achieve local authenticity depicting as user of the system without having authentic access account on that machine.

- **Probing Attack:** (Aburomman & Reaz., 2017) found that in this attack intruders made attempt to acquire information from user system over a network for the motives of exploiting information security.

The most of the attacks that is known to the system is assigned to a unique signatures that makes it feasible to detect attacks. The KDD Cup99 datasets is containing additional 14 attacks types and 24 training attack types in test data.

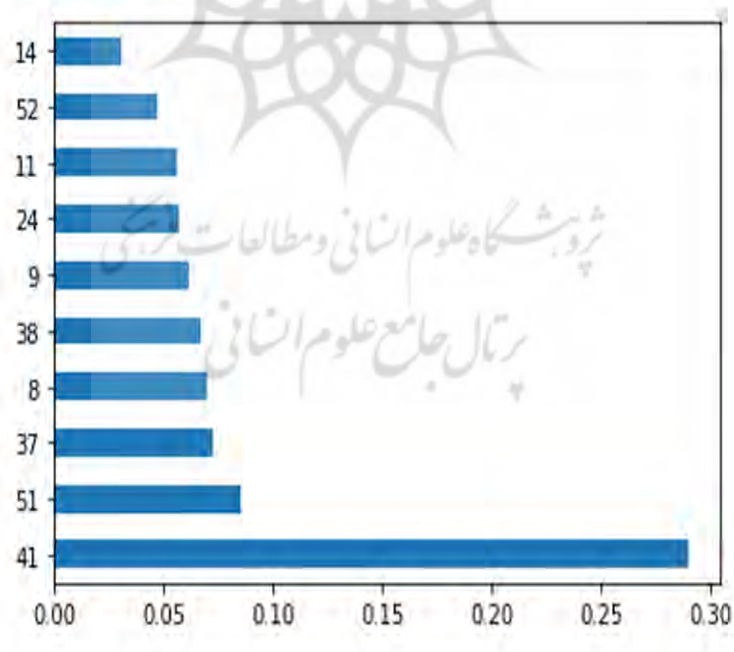

Figure 3. Main Features In Dataset KDD Cup99

Figure 3 shows the main features of KDD Cup99 dataset. There are ten main features on which the outcome of classifier depends and by using these features the detection rate,

training time and accuracy can be improved. These features are based on column indexing of the dataset.

The graphs represent the relationship between independent and dependent data and relationship between categorical to understand the best features for selection to train the model or a classifier.

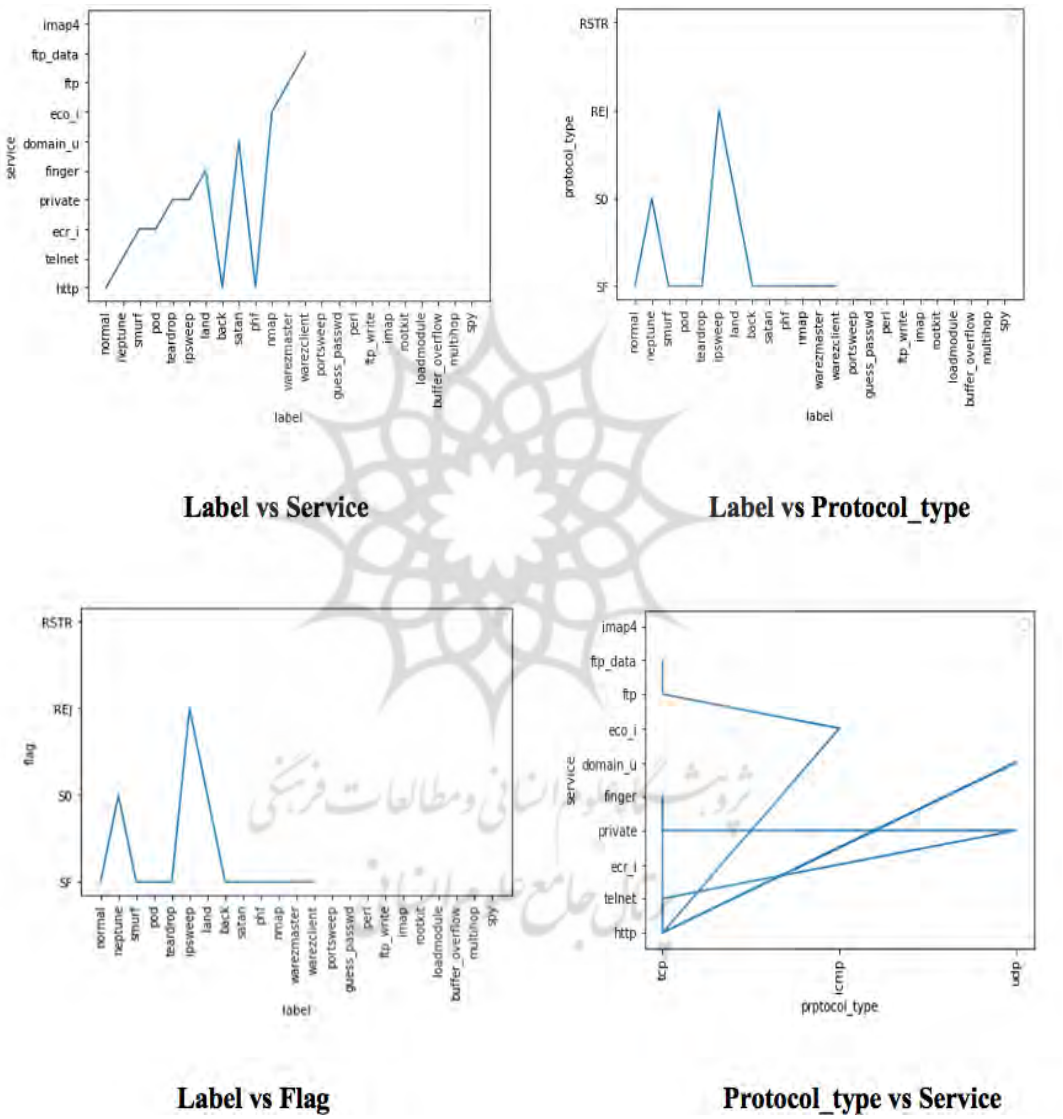

Figure 4. Categorical Data Relationship

Figures 4 shows the categorical data relationship among features by describing the relationship between the features that is related with each other. The graphs shows the relation among Label and Service, Label and Protocol_type, Label an Flag and Protocol_type an Service.

Formulation of Prediction Metrics

Authors specify that prediction metrics is used to determine the results by using algorithmic computations.

$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP} \quad \text{(Linda et al., 2011)} \quad (1)$$

$$Recall = \frac{TP}{TP + FN} \quad \text{(Linda et al., 2011)} \quad (2)$$

$$Specificity = \frac{TN}{TN + FP} \quad \text{(Linda et al., 2011)} \quad (3)$$

$$Precision = \frac{TP}{TP + FP} \quad \text{(Linda et al., 2011)} \quad (4)$$

$$True - Positive\ Rate = \frac{TP}{TP + FN} \quad \text{(Linda et al., 2011)} \quad (5)$$

$$False - Positive\ Rate = \frac{FP}{FP + TN} \quad \text{(Linda et al., 2011)} \quad (6)$$

$$True - Negative\ Rate = \frac{TN}{TN + FP} \quad \text{(Linda et al., 2011)} \quad (7)$$

$$False - Negative\ Rate = \frac{FN}{FN + TP} \quad \text{(Linda et al., 2011)} \quad (8)$$

Here TP- True Positive, TN-True Negative, FN-False Negative and FP- False Positive.

Learning Algorithms Computations

The Machine learning algorithms is defined as supervised learning and unsupervised learning. (Xu et al., 2018) found that the IDS models is built by using supervised Learning algorithms based on signature-based approach, where the attacks are already known and have perquisite signatures for training datasets are support vector machines (SVM), Naive Bayes and random forest. The selection of classification algorithms is made by considering key factors as interpretability, accuracy, nature of data, model assumptions and rate of convergence.

- **Interpretability:** Interpretability is the degree to which machine can comprehend the predictions made. The model is highly interpretable if the predictions made can be easily comprehended. There are classification algorithms as SVM, Naïve Bayes and Random Forest generate high interpretability that allows to know how model is using

data to make better predictions. The decision tree algorithm is highly interpretable and comes with overfitting problem.

- **Accuracy:** The aim is to obtain the higher accuracy by using learning algorithms. The different learning algorithms posses different accuracy depending on the nature of data. If the dataset is having highly unbalanced classes then the accuracy of correctly classifying classes is low. In the case of unbalanced classes linear regression possess low accuracy and decision tree shows ability of class bias to incorporate with data.

- **Nature of Data:** The nature of data significantly plays a vital role in selection of learning algorithm. The highly unbalanced classified data affects the performance of algorithm. If the features in the data are categorical then algorithms as trees are considered due to their high interpretability. When the datasets have lot of features in comparison with data points then algorithm as SVM, Naïve Bayes and Random Forest algorithms are used for generalization of data.

- **Model Assumptions:** To state the ground statistical inference about the model parameters or to quantify variability model assumption is made. The different algorithms made different assumptions. The assumption of data being linearly separable is made by Logistic regression, Decision tree assumes that decision boundaries lie parallel to coordinate axis and random forest assumes the method of averaging performance of several random classifiers. Depending on the type of data and assumption algorithm selection is to be made.

- **Rate of Convergence:** The iterative algorithms converges, that means the output come closer at certain value. The algorithm such as SVM, Naïve Bayes and Random Forest have slower convergence rate as compared to Decision Tree and Logistic Regression. The dataset size also affects the convergence rate for algorithms.

Naive Bayes

The Naive Bayes classifiers are a combination of classification algorithms based on Bayes' theorem. (Aburomman and Reaz, 2017) found that Naïve bayes is a combination of algorithms which depends on a primary principal that states, each and every batch of features is being classified and is independent of each other. The Naive bayes algorithm is completely defined as conditional probability and the maximum likelihood occurrence. The Naïve bayes classifiers is a family of the probabilistic classifier which is based on bayes theorem with naïve independent assumptions among the features.

$$Posterior = \frac{Prior \times Likelihood}{Evidence}$$ (El Kourdi et al., 2004) (9)

Table 1. Performance of Naïve Bayes classifier

	Precision	Recall	F1-Score	Support
0	0.99	0.77	0.87	10894
1	0.07	1.00	0.12	405
2	0.24	0.50	0.32	197
3	0.34	0.88	0.01	8
4	0.99	0.69	0.81	17613
Accuracy			0.73	29117
Macro Avg	0.46	0.77	0.43	29117
Weighted Avg	0.97	0.73	0.82	29117

Table 1 determines the classification report of the model that determines the five attribute to analysis the result by computing the accuracy, precision, recall and f1- score for Naïve Bayes classification algorithms.

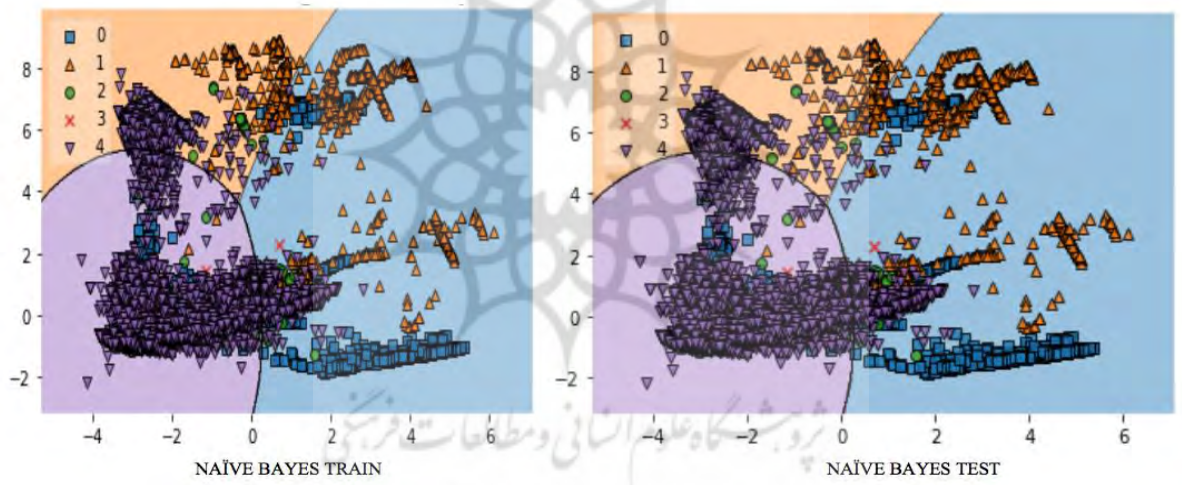

Figure 6. Naïve Bayes Train and Test Results

Figure 6 represents trained and test model comparison by using graphs that shows both the result are close to each other in the context of classification result.

Support Vector Machine

The SVM is specified as supervised machine learning algorithm which is amalgamation of learning algorithms for classification, regression and analysis of data. (Yin et al., 2017) found that SVM is defined by separating hyperplanes using discriminative classifiers. The optimal hyperplane is generated by using SVM algorithm which classifies given labelled training dataset. SVM are support vector network where model is a depiction of data points. The data in vector space is traced into different data categories that is divided by a clear gap. SVM can

effectively perform linear and non linear classification by tracing inputs into dimensional feature spaces.

SVM learning method is kind of supervise learning methods used for classification. It is called margin classifiers because SVM decreases the empirical classification error and maximize the geometric mean simultaneously. (Yin et al., 2017) found that in SVM two parallel hyperplanes are constructed to divide the data inputs for creation of maximum separated hyperplanes to transform vectors into high dimensional spaces. The generalization error would depend on the marginal distance between these hyperplane.

Let consider the points $<(r_1, p_1),(r_2,p_2)\ldots.(r_n, p_n)>$ and for two dimensional vector w.r+p = 0 where w = direction vector and b is scalar constant and once the hyper plane is created then the hyperplane is used to make prediction. The hypothesis function is defined as H(r).

$$H(r) = \begin{cases} +1, w.r + p \geq 0 \\ -1, w.r + p \leq 0 \end{cases} \quad \text{(Koli \& Chavan, 2017)} \quad (10)$$

This shows the points above the plane for class +1 and below the plane for class -1.The goal of hyperplane is to separate the data accurately.

This is for creating datasets where variable P denotes p_sample values and Q contains two classes

P, Q = m_b(p_samples = 500, cen = 2,

SVM_ran_s = 0, cl_s = 0.40)

import matplotlib.pyplot as plt

Table 2. Performance of SVM classifier

	Precision	Recall	F1-Score	Support
0	1.00	1.00	1.00	10894
1	0.99	0.97	0.98	405
2	0.92	0.93	0.93	197
3	0.80	0.50	0.50	8
4	1.00	1.00	1.00	17613
Accuracy			1.00	29117
Macro Avg	0.94	0.88	0.90	29117
Weighted Avg	1.00	1.00	1.00	29117

Table 2 shows the accuracy, precision, recall and f1- score of linear SVM classification algorithms.

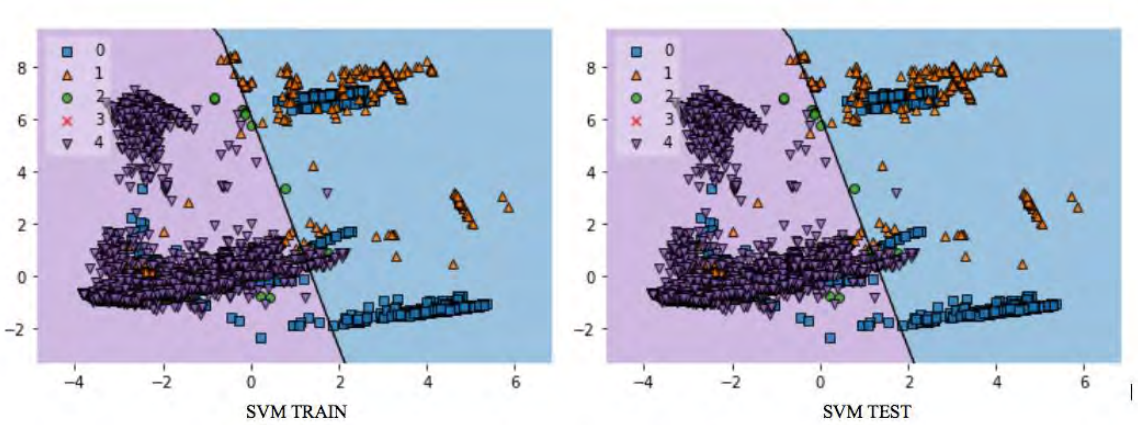

Figure 8. Naïve Bayes Train and Test Results

Figure 8 represents trained and test model comparison by using graphs that shows both the result are close to each other in the context of classification result.

Random Forest

The other form of classification and regression model is random forest which is the supervised learning algorithm. (Xu et al., 2018) found that the random forest algorithm generates decision trees for obtaining prediction from each and every specified data sample and opted the best solution. Random forest method is selected over decision tree algorithm as it averages the result by reducing over fitting. The random forest algorithm involves selection of samples randomly from dataset to construct a decision tree to achieve the precise prediction for every decision tree. The decision tree and prediction is made for every randomly selected sample. The most precise prediction is selected as a final result.

Random forests comes from a combination of tree classifier in which individual decision tree depends on the same distribution pattern for entire tree and independently sampled random vector values. The random forest algorithm combines outcomes of generated multiple decision trees as base classifiers. (Yin et al., 2017) found that for generalization of classifiers the key issues are correlation between base trees and adaptability of each decision tree.

$$normfi_i = \frac{fi_i}{\Sigma j \varepsilon all\ features fi_i} \qquad \text{(Ding \& Zhai, 2018)} \qquad (11)$$

normfi sub(i) = normalized feature i, fi sub(i) = feature i.

$$RFfi_i = \frac{\sum_j normfi_{ij}}{\Sigma j \mathcal{E} all\ features, k \mathcal{E} all\ trees\ normfi_{jk}}$$ (Ding & Zhai, 2018) (12)

RFfi sub(i) = feature i that is calculated from all trees, normfi (ij) = i normalized features for tree j.

Table 3. Performance of Random Forest classifier

	Precision	Recall	F1-Score	Support
0	1.00	1.00	1.00	10894
1	1.00	1.00	1.00	405
2	0.98	0.97	0.98	197
3	0.71	0.62	0.67	8
4	1.00	1.00	1.00	17613
Accuracy			1.00	29117
Macro Avg	0.94	0.92	0.93	29117
Weighted Avg	1.00	1.00	1.00	29117

Table 3 determines the accuracy, precision, recall and f1- score for Random Forest classification algorithms.

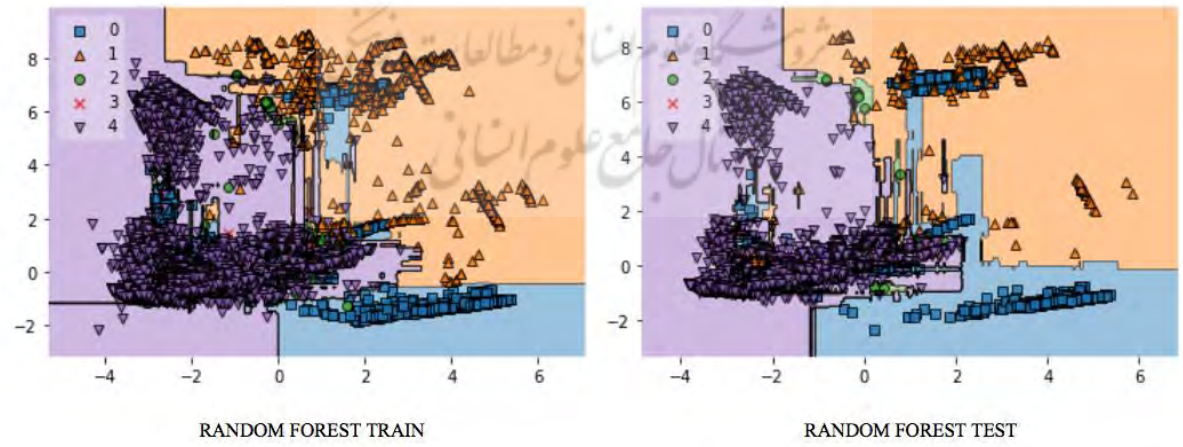

Figure 10. Random Forest Train and Test Results

Figure 10 represents trained and test model comparison by using graphs that shows both the result are close to each other in the context of classification result.

Performance Analysis

The performance analysis of SVM, Random Forest and Naïve Bayes machine learning algorithm is computed as classification report for all the models as it shows the optimal classifier for classification.

Table 4. Performance of Classifiers

Method	Accuracy	Precision	Recall	F1-Score
Naïve Bayes	0.73	0.5	0.77	0.73
Linear SVM	1.00	1.00	0.88	0.91
Random Forest	1.00	0.99	0.89	0.91

Table 4 determines the accuracy, precision, recall and f1- score for Naïve Bayes, SVM and Random Forest classification algorithms.

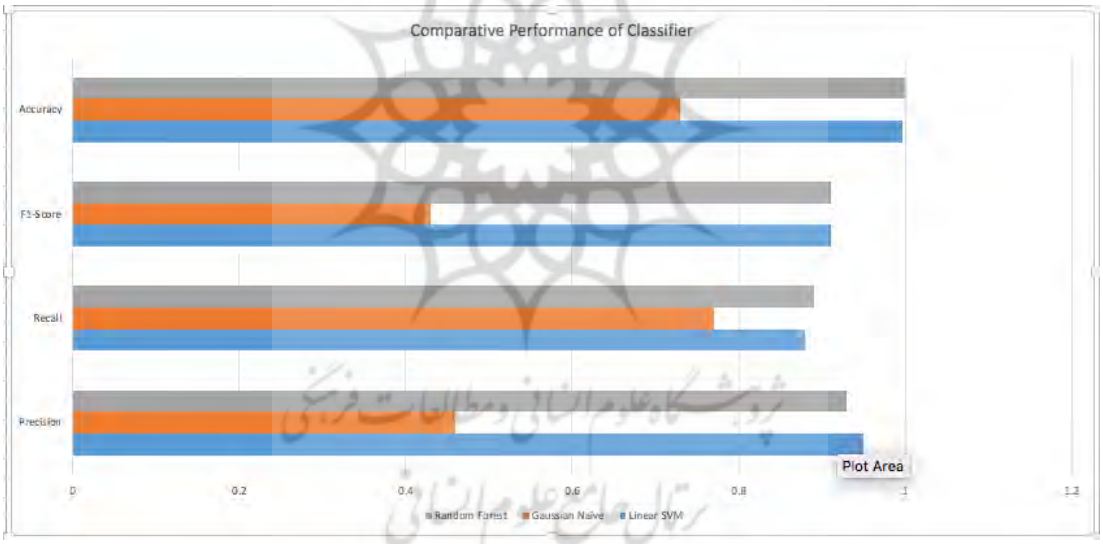

Figure 11. Classification Result Analysis

Figure 11 shows the classification result analysis using graph that shows the accuracy, precision and recall value for Naïve Bayes, SVM and Random Forest.

Detection Rate

The detection rate is computed as the ratio between number of correctly detected intrusion and total number of intrusion.

$$DR = \frac{True\ Positive}{False\ Negative + True\ Positive} \quad \text{(Ding \& Zhai, 2018)} \quad (13)$$

The detection rate computes that Random Forest brings out to be the best suited classifier among Naïve Bayes and SVM and the data classification report shows the detection rate of Naïve Bayes, SVM and Random Forest classifiers. The Naïve Bayes shows the detection rate of 0.971, Linear SVM shows 0.994 and Random Forest shows 0.999.

Research Limitations

The research limitations presents the granularity of research faced by authors for deploying IDS model and shows the limitations of the research followed by authors

- The expanding and vast nature of research domain can make it possible for authors to exempt some of existing research work from literature review conducted by authors.
- The lack of journal accessibility make it possible for authors that some papers are excluded from the study as there exist some journals that cannot be openly accesible
- The proposed framework can be changed by considering overall study of the IDS by considering various journals.
- The authors included the relevant or necessary details regarding the different framework and phases in development of IDS.

Conclusion and Future Scope

The information security is the important aspect to build secure and reliable networks by providing rapid development to information technology. In this paper authors deployed IDS model by using machine learning algorithms as SVM, Naïve Bayes and Random forest to detect intrusions and compute the performance of various algorithms. Authors used KDD Cup99 dataset for comparing the accuracy, precision and detection rate of all the algorithms and for categorization authors used the binary encoder. The performance analysis computes that Random Forest is the best classifier among Naïve Bayes and SVM, and the classification report predicts that Naïve Bayes shows the detection rate of 0.971, SVM shows 0.994 and Random Forest shows 0.999. The paper provides reasonable insight to the research objectives. The future works in the area of IDS are:

- The future work includes multiclass classification algorithm to design machine learning system for detecting and tracking of attacks with IDS models to reduce false positive rates.
- Research on the privacy and security techniques will increase in the future based on the increasing necessity of information security in various fields as social networking, communication, data transfer, digitization and transactions.

References

Abubakar, A., & Pranggono, B., (2017). Machine learning based intrusion detection system for software defined networks. *International Conference on Emerging Security Technologies (EST)*, 138-143.

Aburomman, A. A., & Reaz, M. B. I., (2017). A survey of intrusion detection systems based on ensemble and hybrid classifiers. *Computers & Security*, *65*, 135-152.

Al-Jallad, K., Aljnidi, M., & Desouki, M. S., (2019). Big data analysis and distributed deep learning for next-generation intrusion detection system optimization. *Journal of Big Data*, *6*(1), 80-88.

Aljawarneh, S., Aldwairi, M., & Yassein, M. B., (2018). Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model. *Journal of Computational Science*, *25*, 152-160.

Almseidin, M., Alzubi, M., Kovacs, S., & Alkasassbeh., M, (2017). Evaluation of machine learning algorithms for intrusion detection system. *International Symposium on Intelligent Systems and Informatics (SISY)*, 277-282.

Bulrajoul , W., James, A., & Shaikh., S., (2019). A new architecture for network intrusion detection and prevention. *IEEE Access*, *7*, 558-573.

Chandre, P. R., Mahalle, P. N., & Shinde, G. R., (2018). Machine learning based novel approach for intrusion detection and prevention system: A tool based verification. *Global Conference on Wireless Computing and Networking (GCWCN)*, 135-140.

Ding, Y., & Zhai, Y., (2018). Intrusion detection system for NSL-KDD dataset using convolutional neural networks. *International Conference on Computer Science and Artificial Intelligence*, 81-85.

El Kourdi, M., Bensaid, A., & Rachidi, T. E., (2004). Automatic Arabic document categorization based on the Naïve Bayes algorithm. *Proceedings of the Workshop on Computational Approaches to Arabic Script-based Languages*, 51-58.

Koli, M. S., & Chavan, M. K. (2017). An advanced method for detection of botnet traffic using intrusion detection system. *International Conference on Inventive Communication and Computational Technologies (ICICCT)*, 481-485.

Linda, O., Manic, M., Vollmer, T., & Wright, J., (2011). Fuzzy logic based anomaly detection for embedded network security cyber sensor. *Symposium on Computational Intelligence in Cyber Security (CICS)*, 202-209.

Mohammadi, S., & Namadchian, A., (2017). A new deep learning approach for anomaly base IDS using memetic classifier. *International Journal of Computers Communications & Control*, *12*(5), 677-688.

Selvakumar, B., & Muneeswaran, K., (2019). Firefly algorithm based feature selection for network intrusion detection. *Computers & Security*, *81*, 148-155.

Singh, R., Kalra, M., & Solanki, S., (2019). A Hybrid Approach for Intrusion Detection Based on Machine Learning. *International Conference on Intelligent Sustainable Systems (ICISS)*, 187-192.

Vinayakumar, R., Alazab, M., Soman, K. P., Poornachandran, P., Al-Nemrat, A., & Venkatraman, S., (2019). Deep learning approach for intelligent intrusion detection system. *IEEE Access*, *7*, 525-550.

Xu, C., Shen, J., Du, X., & Zhang, F., (2018). An intrusion detection system using a deep neural network with gated recurrent units. *IEEE Access*, *6*, 697-707.

Yin, C., Zhu, Y., Fei, J., & He, X., (2017). A deep learning approach for intrusion detection using recurrent neural networks. *IEEE Access*, *5*, 954-961.

Bibliographic information of this paper for citing:

Shyla; Kumar, Kapil & Bhatnagar, Vishal (2021). Machine Learning Algorithms Performance Evaluation for Intrusion Detection. *Journal of Information Technology Management*, 13(1), 42-61.