# Optimization of IDS using Filter-Based Feature Selection and Machine Learning Algorithms

**Neha Sharma, Harsh Vardhan Bhandari, Narendra Singh Yadav, Harsh Vardhan Jonathan Shroff**

***Abstract*: Nowadays it is imperative to maintain a high level of security to ensure secure communication of information between various institutions and organizations. With the growing use of internet over the years, the number of attacks over the internet have escalated. A powerful Intrusion Detection System (IDS) is required to ensure the security of a network. The aim of an IDS is to monitor the active processes in a network and to detect any deviation from the normal behavior of the system. When it comes to machine learning, optimization is the process of obtaining the maximum accuracy from a model. Optimization is vital for IDSs in order to predict a wide variety of attacks with utmost accuracy. The effectiveness of an IDS is dependent on its ability to correctly predict and classify any anomaly faced by a computer system. During the last two decades, KDD_CUP_99 has been the most widely used data set to evaluate the performance of such systems. In this study, we will apply different Machine Learning techniques on this data set and see which technique yields the best results.***

***Keywords: Intrusion detection systems, KDDCUP99, Machine Learning, Classification.***

## I. INTRODUCTION

Currently, with a large amount of data and information present on the internet and with various organizations, the premier challenge is to make these systems stable and secure. This is where cyber security comes into picture. Cyber security is a branch of computer science which deals with safeguarding electronic data from criminals and unauthorized users. Network security is a branch of cyber security. Network security deals with prevention of unauthorized access, misuse, malfunction or destruction of network infrastructure by use of software or other physical means thereby creating a secure environment for computer systems. An Intrusion detection system (IDS) is a device or software that monitors a computer network and detects any anomalies. These Anomalies could range from malicious activities to violation of policies. An intrusion detection system can only detect such anomalies but not prevent them from happening. For prevention of anomalies, Intrusion prevention systems (IPS) were made. However, IPS is outside the scope of this project.

There is a huge difference between an IDS and an Anti-Virus. An IDS uses technical detective control which means it can only detect threats and intrusions after a system has been affected by them and is not meant to prevent such incidents. On the other hand, an Anti-Virus uses technical preventive control which means it can stop threats and risks to a system before a chance of infection. There are two ways in which IDSs work, Signature based and Anomaly based. Signature Based IDSs need to maintain a database which contains patterns or signatures of different attacks. If it matches with any of these patterns over a network, the IDS generates an alert. In Anomaly based detection, the systems normal behavior is monitored for anomalies. If there is any deviation from the normal behavior, an alert in generated. Anomaly based IDSs work on a set of rules. KDD_CUP_99 has been used very frequently for evaluating anomaly detection techniques since 1999. It was created by Stolfo et al and comes from the DARPA 98s evaluation of an IDS. DARPA 98 is compressed tcpdump data in binary form. The data was collected over a network for seven weeks and its size is 71.4 MB. This dataset contains exactly 4,94,021 rows and 41 attributes labelled as either normal or an attack.

## II. LITERATURE REVIEW

The main purpose of the Intrusion Detection system is to analyze and identify the attacks made by intruders. A lot of research work has already been done when it comes to dealing with the intrusion detection system. Classification of attacks can be done with the help of different classification models. This segment describes several methodologies adapted by various researchers and authors. All the approaches make the process of classification more approachable as well as more efficient. We will be describing three major researchers whose research gave us direction towards our result. The description is given below:

1). Sumaiya Thaseen, Ch. Aswani Kumar et al. adapted different tree-based classification algorithms which classifies network activities using NSL-KDD 99 dataset. The dataset is heavy and comprises various features. To work with multiple features, it is difficult to attain high efficiency during classification. To avoid the low efficiency, the author has utilized approaches to reduce the dimensionality of attributes of the dataset. The conclusion observed by the work of this specific author implies that Random Tree model provides the maximum amount of accuracy and minimum amount of false alarm rate. The author applied other widely used models to draw the conclusion of better predictive accuracy for decision tree by comparing the accuracy of different intrusion detection models.

**Revised Manuscript Received on November 30, 2020.**

***** Correspondence Author

**Neha Sharma*** Assistant Professor, Manipal University Jaipur, Rajasthan, India. Email: nehav.sharma@jaipur.manipal.edu

**Harsh Vardhan Bhandari** B.Tech, Information Technology, Manipal University Jaipur, Rajasthan, India. Email: harshvardhanbhandari98@gmail.com

**Narendra Singh Yadav**, Associate Professor, Manipal University, Jaipur, Rajasthan, India. Email: narendrasingh.yadav@jaipur.manipal.edu

**Harsh Vardhan Jonathan Shroff**, B.Tech, Information Technology, Manipal University, Jaipur, Rajasthan, India. Email: shroffharsh2@gmail.com

2)Fengli Zhang, Dan Wang et al. adapted an effective methodology of feature selection, nature of approach is based upon Bayesian Network model. The author has used the (NSL-KDD) dataset, the efficiency of the adapted model is calculated, and comparison is done with other generally used feature selection techniques. The comparison of different algorithms is done by a very convenient empirical results which implies that the features taken in consideration by this technique resulted in reduction of time taken to identify the attacks and amplify the classification precision. It has also increased the true positive rates significantly.

## III. RESEARCH METHODOLOGY

### A. Data Pre-Processing:

Unprocessed data may contain incomplete records, missing values, overlapping values, inconsistent data, null values.

1) Missing Values: Missing data can be found due to human errors, machine errors, or due to lack of updating a data set with time. Missing values turn out to be one of the major problems under data preprocessing. It is important to deal with the data which contains the missing values for the data completeness, there are a lot of conventional methods adapted by different researchers. The most popular method is mean, mode imputation. In this methodology the missing values of each attribute is replaced by the mean or mode of that attribute.

2) Categorical Encoding: There are various Machine Learning algorithms which do not function well with categorical variables. The categorical variable needs to be converted into numeric data. This is a very significant step for effective performance of different algorithms implemented. Various algorithm's performance differs based on how categorical variables are put into code. Categorical variables can be bifurcated into two sets: • Nominal • Ordinal

The Nominal data is a type of data which is used to label variables without providing any quantitative value. Ordinal data is a type of data where the variables have usual well-ordered categories. The basic approach is to use integer or label encoding but when categorical variables are nominal, using simple label encoding can be challenging. We can use one hot encoding for this situation.

3) Label Encoder Vs OneHot Encoder: The features in a dataset can contain one or more labels in numeric or in word format. It is easier for humans to make sense of the data in this manner; however, it will not be understandable for a computer. Therefore, for a computer to be able to understand these labels, we use encoding. There are two popular encoders that have been used in this project namely Label Encoder and OneHot Encoder. Label Encoder simply assigns a numeric value to each distinct label and replaces this value in the dataset. It can be aptly used when the labels have different priorities. Let's take an example to understand this better. Assume, the label encoder has assigned different sizes values as follows: Small =1, Medium= 2, Large =3 It is correct to say that small (1) < medium (2) < large (3). Now Assume the label encoder assigns Name of Countries some values as follows: France =1, Spain =2, Germany =3 In this case, it is incorrectly denoted that France < Spain < Germany. Thus, label encoding can be safely applied when different labels have different levels of importance. The solution to this issue is provided by OneHot Encoders. This method creates a new column for every distinct category of an attribute. It splits the column containing categorical data and depending on the value, assigns the value '1' to the column associated with that value. All the other columns take the value '0'. After applying OneHot Encoder we receive a dataset with 137 attributes.

### B. Filter Based Feature Selection:

Feature selection is the process of selecting appropriate set of attributes from the available dataset. The appropriate set of attributes keeps only the significant and main attributes. This process is implemented for better visualization and implementation of different machine learning algorithms. It provides an efficient and more accurate methodology for learning of models. There are numerous feature selection techniques adapted in the field of data science. Broadly feature selection includes three methodologies: Filter, Wrapper and embedded. We have used Filter based feature selection methodology in our project. Filter methodology calculates each feature according to deviation or correlation and sets threshold to select feature, which is inappropriate for the classification performance of classifier. We are using correlation-based methodology in this project. This technique involves a table like structure termed as correlation matrix. The correlation matrix depicts the correlation coefficient between two attributes present in the dataset. The correlation coefficient can take 3 values in the range -1 to 1. Individual cell in the matrix represents the correlation among two attributes. The value of the correlation coefficient decides whether to include the attribute for further usage or not. It depicts how one attribute is affecting the other. If the correlation coefficient of one attribute verses another attribute is positive, then they are positively correlated with each other. If the correlation coefficient turns to be negative, then they are negatively correlated. These dependencies help us in minimizing the dimensionality of the dataset. We have to consider the effect of both the positive and negative coefficient for better selection of features. We apply a limit which is termed as filter to the correlation matrix, in this project it has been set to 0.2. We can decide the value of the filter by observing how the positive and negative coefficient are affecting the entire dataset.

### C. Classification Algorithms Used:

After data pre-processing, we obtained 2 datasets as a result of applying different encoding methods. We applied both Label Encoding and OneHot Encoding. Although it is incorrect to apply Label Encoding on categorical features, the results obtained in doing so were unexpected. Post feature selection the dataset which used Label encoding had 21 attributes and the dataset which used OneHot encoding had 69 attributes. Different algorithms were used for classification on both the datasets. 1) K-Nearest neighbor (KNN): The KNN algorithm uses the Euclidean distance to measure the similarity between all the points of the training data with the points of the test data. The KNN algorithm used here is that attributes (properties) are not weighted but are all same. The k model receives training data points close to the test data point. Most test data points are assigned to the class belonging to the k-neighbor training dataset. The number of neighbors is an important parameter that represents the value of k for the result.

2) Logistic Regression: Logistic regression is a classification algorithm. The reason for the analysis using logistic regression is to develop a model that provides a reasonable correlation between the dependent variable and one or more independent variables. We have independent variables and dependent variables, and we predict the behavior of dependent variables based on independent variables. This classification algorithm uses Sigmoid function and probability assumption to rationalize predictions from 0 to 1. We get curves that vary from 0 to 1. This line is the best adjustment line that corresponds to these records. The main advantage of this approach is that a general probability classification formula can be developed.

3) Support Vector Machine (SVM): SVM was initially developed in the 1960s and then improved until the 1990s. Today it is one of the most powerful weapons in various classification algorithms. The support vector machine creates a classification algorithm and a hyperplane in the N-dimensional space, where N is the total number of entities. Hyper levels are decision limits that classify the data point into different levels. The data points are two different sides of the hyperplane and form two different classes. It is one of the best classification methods with a variety of applications, including the classification of IDS attacks. In our dataset, creating a hyperplane divides into different attacks. We applied SVM on our dataset with parameters Kernel= poly, Degree = 2.

4) Decision Tree: We consider decision tree as a sequence of yes or no series of questions, questioning about the dataset ultimately leading to a specific class. This model is an interpretable model as it classifies the dataset based on the basic principle of decision making, which we use in the ideal world. Decision tree concludes to the final answer after answering multiple layers of yes or no questions with reference to the dataset.

5) Naïve Bayes: Naive Bayes is defined as a classification algorithm suitable for binary (two-class) and multi-class classification problems. The method is convenient to understand conceptually when described using binary or categorical input values. It is named as naive Bayes due to the calculation of the probabilities for individual hypothesis, they are streamlined to create their calculation controllable. Instead of attempting to compute the values of each feature value P(d1, d2, d3|h), they are anticipated to be provisionally independent provided the objective value and calculated as P(d1|h) * P(d2|H) and so on. It is a very imperative assumption which is not likely in real figures, i.e. that the feature does not interact. However, the methodology implements unexpectedly very well on data wherever this assumption does not hold.

Naïve Bayes has extensive usage in real valued attributes, by supposing a Gaussian distribution. This extended feature of Naïve Byes is termed as Gaussian Naïve Bayes Algorithm. Other functional methodology can be utilized to estimate the distribution of data, but the Gaussian Naïve Bayes, which is also denoted as Normal Distribution, is the simplest and less cumbersome to work with. The only thing to be applied on the dataset is the calculation of mean and standard deviation from the training data.

Apart from the typical classification models we used a few ensemble techniques mentioned below to get the best possible result out of the models.

6) Random Forest Classifier: We utilize the basic knowledge of decision making to build the basic block of random forest model. The Random forest is a data model which is prepared with the help of many decision trees. Instead of simply rounding-off the prediction of individual trees, this model utilizes significant concepts. Random forest model is a type of an ensemble algorithm, which associates more than one, same or different kind of algorithms. It develops a group of decision tree from arbitrarily selected subclass of training set. Initially there is an original dataset, we derive Boot-strap dataset. A bootstrap dataset is the result of sampling, that is randomly selecting samples from the original dataset and adding it to the bootstrap dataset. Duplication of the samples are allowed while creating the bootstrap dataset. It is not necessary that all the records from the original dataset will be present in the bootstrap dataset. We plot different decision tress with the help of bootstrap dataset in a randomized fashion. While creating the decision tree root node is decided by subset of variables at each step. Subset of variables implies; we take randomly subset of features derived from the original dataset for selecting the root node for the formation of decision tree. The feature which have more sample splitting ability is considered as the root node. Different decision tree has different nature. Finally, in a test row we classify the target value by comparing the whole tuple by analyzing the different decision tree. Whatever target value is generated according to individual decision tree, is considered and voting count is done. The tuple is classified into the category which receives the maximum number of votes.

7) Bagging (Bootstrap Aggregation): Bagging is a popular ensemble learning technique. Other ensemble models such as the voting classifier work by using different classification algorithms and taking the most frequent output as its result, bagging on the other hand takes a few records from a dataset to create smaller dataset samples called bootstrapped samples. Bagging allows replacements in its bootstrapped samples in the sense that one record can occur in multiple dataset samples. Let us say that the number of dataset samples generated are ‘N’, then in bagging each of these samples will be trained on a different classifier i.e. for ‘N’ samples it must train ‘N’ classifiers. Thus, bagging creates a stronger model which is an aggregation of the classifiers it trains on the generated sample datasets. The resulting classifier has better accuracy and lower error rate than its constituent classifiers and is called a strong classifier.

- In this project, decision tree was selected as the weak classifier for bagging and 10 bootstrapped datasets were created.
- The strong classifier was aggregated by using the voting technique to get the best output from these 10 weaker classifiers.

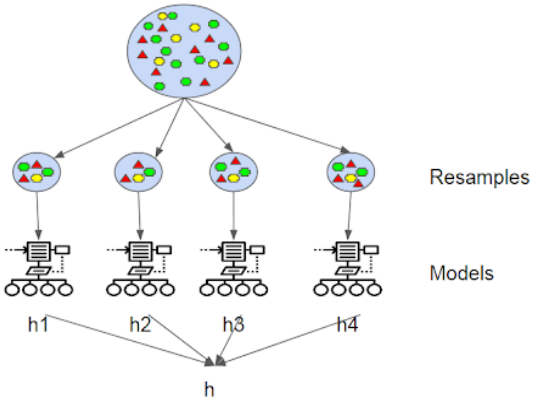


**Fig 3.1: Bagging**

8) Boosting: Like bagging, boosting also generates sample datasets. But there is a catch. Each sample in the record is allotted some weight. Initially all records have equal probability of being selected for a sample dataset. The records are selected at random to create the first dataset sample. Now, a classification model is trained on this sample dataset and the accuracy is calculated. The records that are incorrectly classified are noted and their weight is altered. These weights are updated in such a manner that the probability of these records being selected in the next sample dataset increases. In the next sample some of the incorrectly classified records might be correctly classified. This cycle is repeated till the specified number of models are generated. Each model has higher accuracy and lower error rate than the previous model. In the end all these models are aggregated to create a strong learning model. This strong model has higher accuracy and lower error rate than any of its constituent models.

In this project, decision tree has been selected as the weak classifier and 10 weak models were created to build the strong model. The learning rate has been set to 2 as it yields maximum accuracy. To understand the concept of learning rate, let us say the learning rate is 'N'. Learning rate decreases the importance of each classifier by a factor of 'N'.

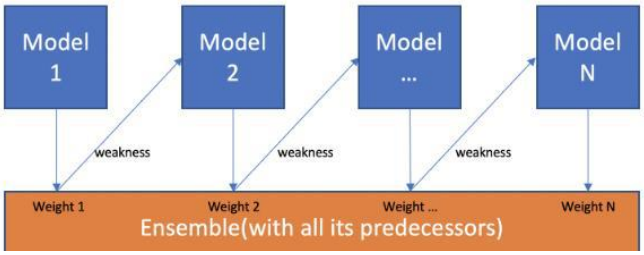


**Fig 3.2: Boosting**

9) Voting Classifier: Voting classifier is an ensemble learning technique. It takes two or more classifiers, trains them and finds considers the output in case of each algorithm. Each algorithm may give a different algorithm with different input. Then it counts which output is obtained the maximum number of times by these algorithms and selects that output to be the result. The voting can be set to 'Hard' or 'Soft'. The Voting Classifier method can be found in the sklearn. ensemble library of python. This method of ensemble learning can be modified easily by using different Machine Learning algorithms to obtain different accuracies. It gives better accuracy than individual classification algorithms and is a very handy ensemble technique. The target or dependent variable in KDDCUP99 is a categorical feature. On applying OneHot Encoding the 'Label' attribute splits into 23 columns, each is a different attack type. Due to this the model becomes a multi output model. The classification models are not meant to handle Multi output classification problems. To resolve this problem, we have used something called as Multi Output Classifier.

### D. Multi Output Classifier:

Most classification algorithms are aimed at predicting the value of a single target attribute. However, some problems require prediction of multiple features in the same model. This is where Multi-Output Classifier comes into picture. It overcomes the problem by fitting one classifier per target variable. It is used to extend the utility of classifiers that do not natively support multi output classification. Multi-Output Classifier () is available in the sklearn. Multi-output library of python. This strategy holds high importance in this project. The use of OneHotEncoder () creates as many new columns/features as there are categories for every attribute it is applied on. Hence, we obtained 22 additional target attributes in the modified dataset. The use of Multi Output Classifier enabled the use of algorithms like K Nearest Neighbor, Gaussian Naïve Bayes and Decision tree on the modified dataset.

## IV. RESULTS AND DISCUSSION

A bar graph was made for the all the algorithms, comparing their accuracies. It has been observed that the decision tree classifier when used individually, tends to over fit on the model. However, when used with Ensemble learning techniques it provides much better results.

### A. In the Label Encoded dataset, we applied the following algorithms and found their accuracies:

i. K- Nearest Neighbors (KNN)
ii. Decision Tree
iii. Logistic Regression
iv. Support Vector Machine (SVM)
v. Random Forest
vi. Bagging
vii. Boosting
viii. Voting Classifier

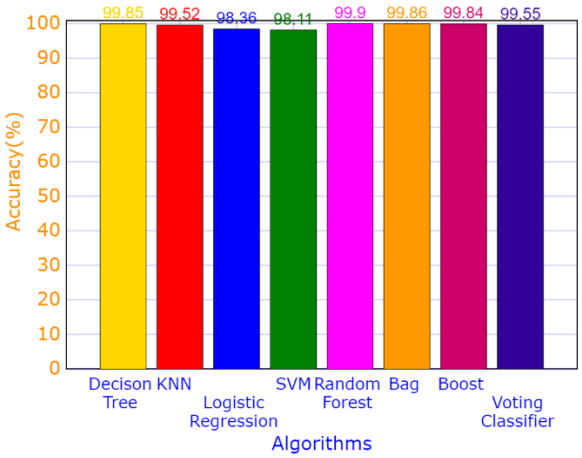


**Fig 4.1: Bar Graph of Accuracies of Algorithms used with Label Encoder**

The accuracies of the algorithms are:

**Table 1. Accuracies of various algorithms using Label Encoder**

| SNO. | ALGORITHM NAME | ACCURACY |
|---|---|---|
| 1 | K- Nearest Neighbors (KNN) | 99.52 |
| 2 | Decision Tree | 99.85 |
| 3 | Logistic Regression | 98.36 |
| 4 | Support Vector Machine (SVM) | 98.11 |
| 5 | Random Forest | 99.90 |
| 6 | Bagging | 99.86 |
| 7 | Boosting | 99.84 |
| 8 | Voting classifier | 99.55 |

**B. In the OneHot Encoded dataset we applied the following algorithms and found their accuracies:**

i. Decision Tree
ii. Gaussian Naïve Bayes
iii. K-Nearest Neighbors (KNN)
iv. Random Forest
v. Bagging
vi. Boosting
vii. Voting Classifier

In the OneHot Encoded dataset we had to use Multi Output Classifier which extends the scope of some classification algorithms to work on models which have multiple values to predict. This is a useful tool when the target variable contains categorical features.

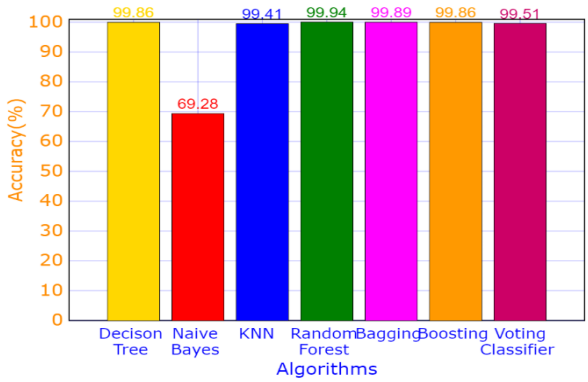


**Fig 4.2: Bar Graph of Accuracies of Algorithms used with OneHot Encoder**

The accuracies of the algorithms are:

**Table 2. Accuracies of various algorithms using OneHot Encoder**

| SNO. | ALGORITHM NAME | ACCURACY |
|---|---|---|
| 1 | Decision Tree | 99.86 |
| 2 | Gaussian Naïve Bayes | 69.28 |
| 3 | K-Nearest Neighbors (KNN) | 99.41 |
| 4 | Random Forest | 99.94 |
| 5 | Bagging | 99.89 |
| 6 | Boosting | 99.86 |
| 7 | Voting Classifier | 99.51 |

Keep in mind that the voting classifier in both the datasets uses a smaller testing and training dataset.

Using the regular testing and training set was very time consuming and therefore we decided to use 40,000 rows for training and 20,000 rows for testing.

## V. CONCLUSION

In this project we have used Filter based feature selection method to remove the insignificant attributes which would only have a negative impact on the accuracy of the machine learning models. The correlation matrix method of feature selection was used to find the correlation between the target attributes and the independent attributes. The feature selection techniques can only be applied once the dataset has been converted to a machine friendly dataset with only numbers that the machine can crunch. There must be no strings in the dataset and no missing values. To tackle the problem of missing values we have used Simple Imputer and we convert categorical string values to numeric data using 2 different encoding techniques 1. Label Encoding 2. OneHot Encoding. Although using Label Encoder should be the wrong method to use on categorical data, its use in this project yielded unexpected results. Moreover, it created a simple and comparatively smaller dataset which was easier to work on. The accuracy of all the machine learning models used generated high accuracies (in the range 98-100 %). This may be the result of the problems in the KDDCUP99 dataset as mentioned earlier. The OneHot Encoding technique can be applied only on features with multiple values but no priority. In this technique a new column is created for every value in the categorical attribute. This method works better than the Label encoding method and is the correct approach to follow. After applying this technique, the new dataset generated has 141 attributes out of which only 69 are selected after feature selection. The accuracy of all the machine learning models used generated high accuracies. The Decision Tree algorithm alone gives very high accuracy as it tends to overfit to the model. Using Decision Tree with ensemble techniques can solve this issue. In this project Decision Tree has been used in 4 different ensemble techniques like bagging, boosting, Random Forest and voting classifier. The target attribute in KDDCUP99 is a categorical feature, hence when OneHot encoding is applied to it our model becomes a Multi output model. Multi Output classification is out of the scope of most classification algorithms. The use of Multi Output Classifier can extend the scope of these classification algorithms, thus can be successfully applied on Multi Output Targets. Now, coming to the classification algorithms used. Apart from the typical machine learning algorithms like SVM, Logistic Regression, K Nearest Neighbors etc. we have also implemented Bagging, Boosting, Random Forest and Voting Classifier. Bagging and Boosting create smaller datasets balled bootstrap samples and train these datasets on individual models creating weak classifiers. In the end these weak classifiers are aggregated to make a strong classifier with greater accuracy and lower error rate. Random Forest creates different Decision trees and trains each tree individually and creates a forest of trees. It tests these trees and the predicted value which occurs the most in the forest is passed as the output of the Random Forest Classifier.

Voting Classifier is an unorthodox algorithm that has been used in this project. This technique takes 2 or more classification algorithms and trains them. It predicts the output from each of its sub classifiers, takes a vote and passes the value that occurred most frequently as the output. As it takes many classification algorithms it takes a lot of time to execute. Therefore, we used 40,000 tuples for training and 20,000 tuples for testing. The classifier still yields high accuracy. In conclusion, the voting classifier is the strongest algorithm we have used in this project and even though the number of tuples used to train and test the model are fairly low (about 10% of the whole dataset for training and about 5% for testing) it gives significantly high accuracy. If we use the whole dataset, the accuracy will supposedly increase.

The accuracies of these algorithms are unusually high mainly because of the problems within the KDD dataset. The solution to this problem is the use of a newer dataset called NSL-KDD which was made using KDDCUP99.

The advantages of NSL-KDD over KDDCUP99 are as follows:

• Removes all the redundant records in the dataset thus making the size of the dataset more reasonable.

• Due to low redundancy the classification algorithms won't be biased towards frequently occurring records.

• This biased nature of KDD prevents the Machine learning algorithms to learn the infrequent attacks that are more dangerous like U2R and R2L. NSL has smaller, less redundant and uniformly spread attack entries so the algorithms can learn each attack without being biased.

• The NSL dataset is divided into training and testing dataset called KDDTrain+ and KDDTest+. This makes the classification rates of various Machine learning algorithms differ along a wider range hence making it easier to evaluate these algorithms and select the ones which produce maximum accuracy.

In the above-mentioned ways, we can select the best algorithm for optimization of Intrusion detection systems among the various algorithms out there.

## REFERENCES


1. West, D., 2000, "Neural network credit scoring models." Computers & Operations Research 27(11-12), 1131-1152.
2. Malhotra, R., and Malhotra, D. K., 2003, "Evaluating consumer loans using neural networks." Omega, 31(2), 83-96.
3. Hsieh, N. C., 2004, "An integrated data mining and behavioral scoring model for analyzing bank customers." Expert Systems with Applications 27(4), 623-633.
4. Angelini, E., Tollo, G. d., and Roli, A., 2008, "A neural network approach for credit risk evaluation." The Quarterly Review of Economics and Finance 48(4), 733–755.
5. Paredes, R., and Vidal, E., 2000, "A class-dependent weighted dissimilarity measure for nearest neighbor classification problems." Pattern Recognition Letters 21(12), 1027-1036.
6. Hand, D. J., and Vinciotti, V., 2003, "Choosing k for two-class nearest neighbor classifiers with unbalanced classes." Pattern Recognition Letters 24(9-10), 1555-1562.
7. Islam, M. J., Wu, Q. M. J., Ahmadi, M., and Sid-Ahmed, M. A., 2007, "Investigating the Performance of Naive- Bayes Classifiers and K-Nearest Neighbor Classifiers"International Conference on Convergence Information Technology. IEEE Computer Society
8. Huang, Z., Chen, H., Hsu, C. J., Chen, W. H., and Wu, S., 2004, "Credit rating analysis with support vector machines and neural networks: a market comparative study." Decision Support Systems 37(4), 543– 558.
9. Gestel, T. V., Bart Baesens, Dijcke, P. V., Garcia, J., Suykens, J. A. K., and Vanthienen, J., 2006b, "A process model to develop an internal rating system: Sovereign credit ratings." Decision Support Systems 42(2), 1131-1151.
10. Chen, W. H., and Shih, J. Y., 2006, "A study of Taiwan's issuer credit rating systems using support vector machines." Expert Systems with Applications 30(3), 427-435.
11. Steenackers, A., and Goovaerts, M. J., 1989, "A credit scoring model for personal loans." Insurance: Mathematics and Economics, 8(1), 31-34.
12. Laitinen, E. K., 1999, "Predicting a corporate credit analyst's risk estimate by logistic and linear models." International Review of Financial Analysis 8(2), 97-121.
13. Alfo, M., Caiazza, S., and Trovato, G., 2005, "Extending a Logistic Approach to Risk Modeling through Semiparametric Mixing." Journal of Financial Services Research, 28(1), 163-176.
14. Proceedings of the 2011 International Conference on Industrial Engineering and Operations Management Kuala Lumpur, Malaysia, January 22 – 24, 2011
15. (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 8, No.7, 2017
16. Wang, Hong, Qingsong Xu, and Lifeng Zhou. 2015. Large Unbalanced Credit Scoring Using Lasso-Logistic Regression Ensemble. PLoS ONE 10: e0117844. [Google Scholar] [CrossRef] [PubMed]
17. Bellotti, Tony, and Jonathan Crook. 2009. Support Vector Machines for Credit Scoring and Discovery of Significant
18. 8 Features. Expert Systems with Applications. [Google Scholar] [CrossRef]
19. Lessmann, Stefan, Bart Baesens, Hsin Vonn Seow, and Lyn C. Thomas. 2015. Benchmarking State-of-the-Art Classification Algorithms for Credit Scoring: An Update of Research. European Journal of Operational Research 247: 124–36. [Google Scholar] [CrossRef]
20. Wójcicka, Aleksandra. 2017. Neural Networks vs. Discriminant Analysis in the Assessment of Default. Electronic Economy, 339–49. [Google Scholar] [CrossRef]
21. Bacham, Dinesh, and Janet Zhao. 2017. Machine Learning: Challenges and Opportunities in Credit Risk Modeling. Available online: https://www.moodysanalytics.com/risk-perspectives-magazine/managing-disruption/spotlight/machine-learning-challenges-lessons-and-opportunities-in-credit-risk-modeling (accessed on 2 April 2018).
22. Bastos, João A. 2014. Ensemble Predictions of Recovery Rates. Journal of Financial Services Research 46: 177–93. [Google Scholar] [CrossRef


## AUTHORS PROFILE

**Neha Sharma*** is an Assistant Professor in department of Information Technology, Manipal University Jaipur, India. She is currently pursuing her PhD. in Network Security from Manipal University Jaipur, India. She has an overall experience in industry and academics of more than 10 years. She has many National and International publications to her credit.

**Harsh Vardhan Bhandari** has recently graduated from Manipal University Jaipur with a degree in B. Tech, Information Technology. He recently developed a chatroom application using sockets in python. The application was hosted on a Digital Ocean server which enabled remote users to communicate with one another using the client program.

**Dr. Narendra Singh Yadav** received his M.Tech. degree in Computer Science from Birla Institute of Technology, Ranchi and Ph.D degree from Malaviya National Institute of Technology, Jaipur. He is currently an Associate Professor (Senior Scale) of Information Technology with Manipal University Jaipur, Jaipur. His research interests include Computer Networks, Network Security, Digital Forensics, Cyber Security, Malware and Reverse Engineering. He is an active member of various professional bodies and has published over 50 papers in International/national Journals/conferences.

He is a certified ethical hacker (CEH), certified computer hacking forensics investigator (CHFI), certified ec-council instructor (CEI), certified security analyst (ECSA), Microsoft technology Associate (MTA) and ISO 27001 LA.

**Harsh Vardhan Jonathan Shroff** has recently graduated from Manipal University Jaipur with a degree in B. Tech, Information Technology. He is currently working on a number of security projects.