JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Anomaly Based Intrusion Detection System Using Machine Learning Technique

[1]G.Mahalakshmi, [2]Dr. E.Uma

[1] Teaching Fellow and [2]Assistant Professor (Sl.Gr)
Department of Information Science and Technology, Anna University Chennai

Abstract: The network is a key in the modern world thereby controlling all human activities, where all the communications are happening via the internet. Since people rely on digital communication of pieces of information, creates the necessity for security. The Information passing over the air fabricates the information threat or attack on the network. From this, it is understandable the vital of rendering security to prevent attacks or threats. There are various security mechanisms are available like firewalls, anti-virus, IDSs, and many others. The most commonly used mechanism is Intrusion Detection System. There are different flavors of IDSs available. In this paper, a Network-based Anomaly Detection System is proposed. An intrusion detection system utilizing machine learning-based classification makes for an effective decision-making process for identifying the attacker nodes. The proposed IDS uses the novel machine learning algorithm for classifying the normal data packets and attacker data packets. The proposed system aims in reducing the False Positive rate and Increasing the packet delivery ratio.

Index Terms: **IDS, Chi^{2}FSACA, False Positive, Packet Delivery Ratio**

I. INTRODUCTION

The necessity for securing data is getting increased in recent days due to the availability and usage of networks (Wired, Wireless, or Adhoc). CIA (Confidentiality, Integrity, and Availability) plays an essential role in providing secure communication of data. Networks are a widely used technology all over the globe compared to other technologies. The network enables communication from person to person or person to the device. The information is getting shared with the help of network technology. Security is a buzz term in all domains (information security, cyber security). Usually, a widely used default technique for providing security in networks and computer systems are Anti-virus software and Firewalls. Apart from these another greatly using technique is IDSs. This Intrusion Detection System monitors and detects attacks and threats on the communication network thereby providing security[18].

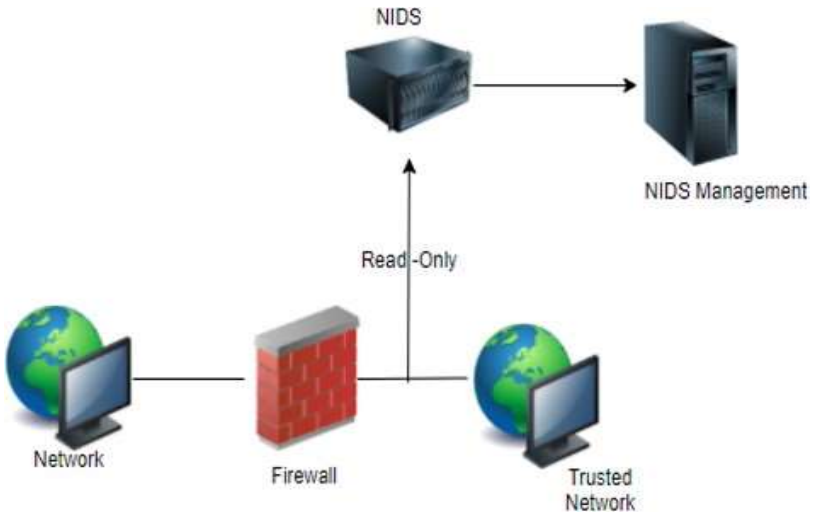


Figure 1.1: General Intrusion Detection System

The IDSs become a very important segment in the network by capturing the various types of attacks in the prime stages. IDS applies various classification methods for its decision-making process on concluding a normal packet or an attacker packet (i.e. Denial Of Service, U2R, R2L, TCP SYN Flood Attack, Password Attack).

1.1 Miscellaneous Classes of Attacks

Denial of Service (DOS)	This type of attack does prevention on authorized users from availing of resources or services.
User to Root (U2R)	This type of attack gains access of machines of local user and attempts to gain administration rights.
Remote to Local (R2L)	This type of attack performs a blind search or operation on legtimate user systems without having authentication login.
TCP SYN Flood Attack	An attacker exploits the use of the buffer space during a Transmission Control Protocol (TCP) session initialization handshake.
Password Attack	An attacker utilize the social media and open network to guess the user password or obtain the password from the database.

The rest of the thesis is marshaled as follows section 2 discusses the literature survey. Section 3 describes the architectural design of the proposed system. Section 4 describes the proposed algorithms and

experimental setup. Section 5 describes the performance analysis of the proposed algorithms. Section 6 discusses the conclusion and future enhancement.

2. LITERATURE SURVEY

This chapter gives an overview of literature surveys. This chapter represents some of the relevant work done by the researchers. Many existing techniques have been studied by the researchers on intrusion detection, a few of them are discussed below.

2.1. FEATURE SELECTION

Coelho, F. et al, [6]. The author of the principle of homogeneity between labels and data clusters is exploited in order to develop a semi-supervised Feature Selection method. This principle permits the use of cluster information to improve the estimation of facet relevance in order to increase selection performance. Mutual Information is used in a Forward-Backward search undertaking in order to evaluate the relevance of each facet to the data distribution and the existent labels, in a context of few labeled and many unlabeled instances.

H. Gharaee and H, [9]. Hossein and Intrusion detection systems (IDS) are the main components of network security. IDSs monitor events of a system in a network, and probe the behavior in order to detect intrusions. One of the IDS models is peculiarity-based IDS which trains to distinguish between normal and abnormal traffic. One of the peculiarities based IDSs are based on the Genetic algorithm is an evolutionary optimization algorithm. This paper has proposed a peculiarity-based IDS using a Genetic algorithm and Support Vector Machine (SVM) with a new facet selection method. The new model has used a facet selection method based on genetics with an innovation in fitness function to reduce the dimension of the data, increase true positive detection and simultaneously decrease false positive detection. In addition, the computation time for training will also have a remarkable reduction. Results show that the proposed method can reach high accuracy and low false-positive rate (FPR) simultaneously, though it had earlier been achieved in earlier studies separately. This study proposes a method that can achieve more stable facets in comparison with other techniques. The proposed model experiment and test on KDD CUP 99 and UNSW-NB15 datasets. Numeric Results and comparison to other models have been presented.

Sumaiya, et al, [12]. The author's idea behind this model is to design a multi-class SVM that has not been adopted for IDS to decrease the training and testing time and increase the individual classification accuracy of the network attacks. The investigational results on the NSL-KDD dataset which is an enhanced version of the KDDCup 1999 dataset show that their proposed approach results in a better detection rate and reduced false alarm rate.

2.2. ANOMALY DETECTION SYSTEMS

Mohammadreza Ektela and et.al, [3]. used Support Vector Machine and classification tree Data mining technique for intrusion detection in networks. They compared C4.5 and Support Vector Machine

by the experimental result and found that the C4.5 algorithm has better performance in terms of detection rate and false alarm rate than SVM, but for U2R attack SVM performs better.

M.Govindarajan and et.al, [7]. proposed a new K-nearest neighbor classifier applied on the Intrusion detection system and evaluated performance in terms of Run time and Error rate on the normal and malicious dataset. This new classifier is more accurate than the existing K-nearest neighbour classifier.

R. Primartha and B. A. Tama, [11]. Intruders have become more and more sophisticated thus a deterrence mechanism such as an intrusion detection system (IDS) is pivotal in information security management. An IDS aims at capturing and repealing any malignant activities in the network before they can cause harmful destruction. An IDS relies on a well-trained classification model so the model is able to identify the presence of attacks effectively. This paper compares the performance of IDS by exerting a random forest classifier with respect to two performance measures, i.e. accuracy and false alarm rate. Three public intrusion data sets, i.e NSL-KDD, UNSW-NB15, and GPRS are employed in the experiment. Furthermore, different tree-size ensembles are considered whilst other best learning parameters are attained using a grid search. Our experimental results prove the superiority of the random forest model for IDS as it significantly outperforms the similar ensemble, i.e. ensemble of a random tree + naive Bayes tree and another single classifier, i.e. naive Bayes and neural network in terms of K-cross-validation method.

Roshan Chitrakar, et.al.[15], proposed a hybrid approach to intrusion detection by using k-Medoids clustering with Naïve Bayes classification and observed that it gives better performance than K-Means clustering technique followed by Naïve Bayes classification but also time complexity increases when increasing the number of data points.

2.3. CLASSIFIER APPROACH

L. Dhanabal, and S.p.Shantharajah, [17]. The NSL-KDD data set is analyzed and used to study the effectiveness of the various classification algorithms in detecting anomalies in the network traffic patterns. We have also analyzed the relationship of the protocols available in the commonly used network protocol stack with the attacks used by intruders to generate anomalous network traffic. The analysis is done using classification algorithms available in the data mining tool WEKA. The study has exposed many facts about the bonding between the protocols and network attacks.

Ondrej Linda, and et al, [13]. The author developed an IDS-NNM – Intrusion Detection System using Neural Network based Modeling to analyze the real network data to develop a specific window-based feature extraction technique and construct a combination of two neural network learning algorithms – the Error-Back Propagation and Levenberg Marquardt, for normal behavior modeling and their work shows the IDS-NNM algorithm is capable of capturing all intrusion attempts presented in the network communication while not generating any false alerts.

P. Amudha, and et.al, [14]. observed that Random forest gives better detection rate, accuracy, and false alarm rate for Probe and DOS attack & Naive Bayes Tree gives better performance in case of U2R and R2L attack. Also, the execution time of Naive Bayes Tree is more as compared to other classifiers.

R. China Appala Naidu et.al.(2012) [2], used three Data mining techniques SVM, Ripper rule, and C5.0 tree for Intrusion detection, and also compared the efficiency. By experimental result, the C5.0 decision tree is more efficient than others. All three Data mining techniques give a higher than 96% detection rate.

Deepthy k Denatious and et.al, [1]. describe different data mining techniques applied for detecting intrusions. Also, describe the classification of the Intrusion detection system and its working. For large amounts of network traffic, clustering is more suitable than classification in the domain of intrusion detection because of the enormous amount of data needed to collect to use classification.

2.4. INTRUSION DETECTION USING BIG DATA

P. Dahiya and D. Srivastava, [16]. They proposed a framework in which a facet reduction algorithm is used for reducing the less important facets and then applied the supervised data mining techniques on the UNSW-NB15 network dataset for fast, productive, and accurate detection of intrusion in the Netflow records using Spark. In this paper, we have used two facet reduction algorithms, namely, Canonical Correlation Analysis (CCA) and Linear Discriminant Analysis (LDA), and seven well-known classification algorithms. In order to compare the performance of the proposed framework, five performance metrics such accuracy, Specificity, Kappa, and Mean Abs. Error, FPR, Precision, Recall, ROC Area, and Training Time are used.

Osama Faker, [10]. Big Data and Deep Learning Techniques are integrated to improve the performance of intrusion detection systems. Three classifiers are used to classify network traffic datasets, and these are Deep Feed-Forward Neural Network (DNN) and two ensemble techniques, Random Forest and Gradient Boosting Tree (GBT). To select the most relevant attributes from the datasets, we use a homogeneity metric to evaluate facets. Two recently published datasets UNSW NB15 and CICIDS2017 are used to evaluate the proposed method. 5-fold cross-validation is used in this work to evaluate the machine learning models. We implemented the method using the distributed computing environment Apache Spark, integrated with Keras Deep Learning Library to implement the deep learning technique while the ensemble techniques are implemented using Apache Spark Machine Learning Library. The results show a high accuracy with DNN for binary and multiclass classification on the UNSW NB15 dataset with accuracies of 99.16% for binary classification and 97.01% for multiclass classification. While the GBT classifier achieved the best accuracy for binary classification with the CICIDS2017 dataset at 99.99%, for multiclass classification DNN has the highest accuracy with 99.56%.

3. SYSTEM DESIGN

3.1. PROPOSED ALGORITHM

Several ML methods have been propounded to monitor and probe network traffic for different anomalies. Most of these methods (classifiers) identify the peculiarity by looking for variations from a basic normal traffic model. Usually, these models are trained with a set of attack-free traffic data that is collected over a long period. Any ML peculiarity detection method is one of three broad categories that are Supervised, Unsupervised, or Semi-supervised learning methods. In this paper, we will focus on Supervised learning classifiers.

Supervised learning uses training data which is a set of examples with paired input records and their fitting outputs. In this learning, the correct answer is known in advance, and the learner algorithm iteratively makes predictions on the training data and stops only when an acceptable level of performance is achieved. Thus, this method is appropriate when there is a specific target value. Supervised learning difficulties can be defined as either classification difficulties or regression difficulties. The most famous supervised learning algorithms are Decision Trees (DT), Support Vector Machines (SVM), Neural networks (NN), Random Forests (RF), etc.

3.2 SYSTEM ARCHITECTURE

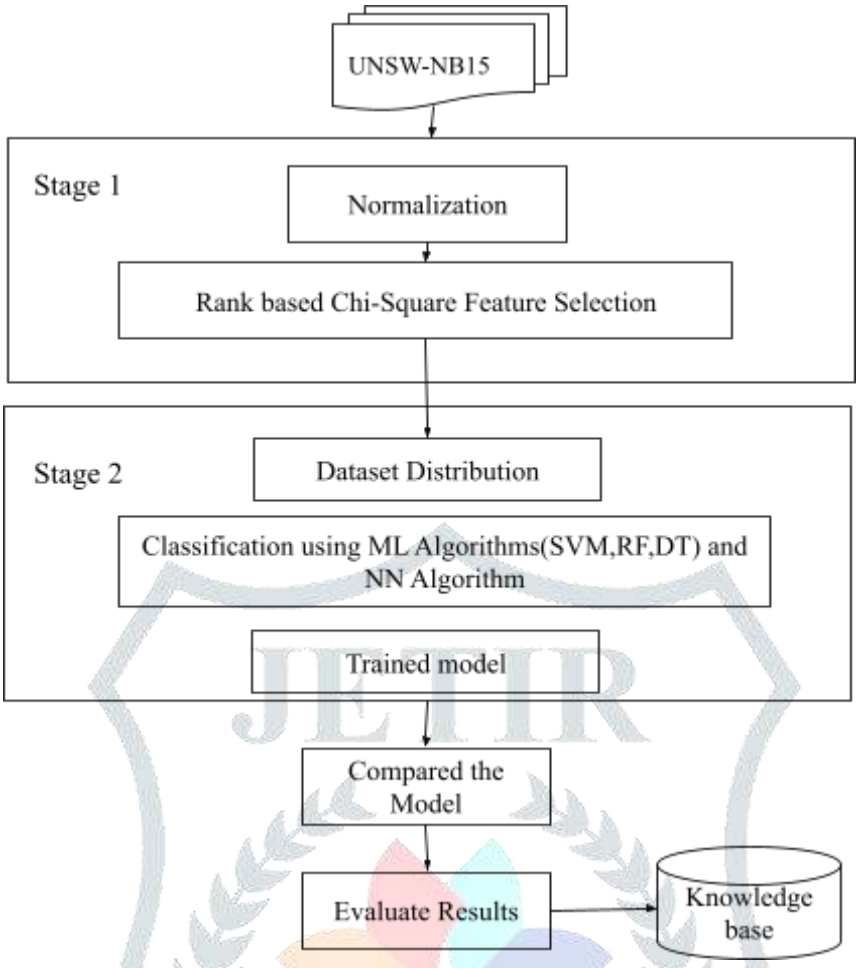


Figure 3.1: Architecture Diagram

Figure 3.1 represents the system architecture of the proposed system, where applying a Machine Learning algorithm to identity intrusion. Here UNSW-NB15 dataset is given as input and given input raw data converted into a form that fits machine learning algorithms then the feature selection process will be done for preprocessed data using chi-square and the selected features will be sent to the classification module.

3.3 USE CASE DIAGRAM

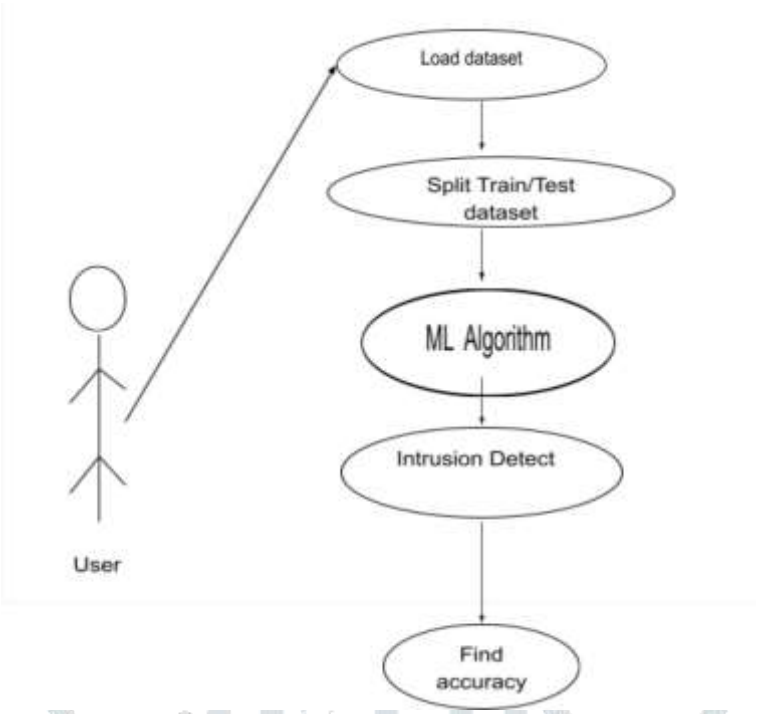


Figure 3.2: Use Case Diagram

Figure 3.2 represents a use case diagram, in which the user upload dataset is pre-processed and applied to the algorithm. They are analyzed for the intrusion of four different types.

4. IMPLEMENTATION

The proposed work is implemented in Python 3.7 with libraries Tensorflow, Keras, pandas, matplotlib, and other mandatory libraries. We downloaded the dataset from kdd.ics.uci.edu. The data downloaded contains a train set and test set separately with two different classes of intrusions. The training dataset is considered as the train set and the test dataset is considered as the test set.

Have collected the dataset for the intrusion detection system with the following details from the UNSW-NB15 dataset and we applied a Machine learning algorithm.

4.1 DATASET DETAILS

It is important to note that the test data is not from the same probability distribution as the training data, and it includes specific attack types not in the training data. This makes the task more realistic. Some intrusion experts believe that most novel attacks are variants of known attacks and the "signature" of known attacks can be sufficient to catch novel variants.

4.2 FUNCTIONAL REQUIREMENTS

Data Collection

The data collection undertaking involves the selection of quality data for analysis. The UNSW-NB15 intrusion dataset was taken from uci.edu for machine learning implementation. The job of a data analyst is to find ways and sources of collecting relevant and comprehensive data, interpreting it, and analyzing results with the help of statistical techniques [19].

4.2.1 Data Preprocessing

The purpose of preprocessing is to convert raw data into a form that fits machine learning. Structured and clean data allows a data scientist to get more precise results from an applied machine learning model. The technique includes data formatting, cleaning, and sampling. Here, the technique used is

1. Data normalization
    a. Min-Max Normalization: For each facet, the minimum value of that facet gets transmogrified into a 0, the maximum value gets transmogrified into a 1, and each other value gets transmogrified into a decimal between 0 and 1.
2. Data encoding
    b. Label encoding: It is the process of changing the non-numeric labels into numeric labels for the machine-understandable form purpose.

Feature selection

Feature selection is the undertaking of attaining the score for each potential feature and then attaining the excellent ‘k’ features. Scoring is done by counting the frequency of a feature in training positive and negative class samples separately and then attaining a function of both. There are many features that have to be monitored for intrusion detection out of which certain features will be useful and others may be useless. The removal of useless features enhances the accuracy and decreases the computation time thereby achieving higher performance. The chi-square feature selection metric is used in our model.

Dataset Splitting

The process of chunking the dataset has to be done for training as well as for testing the model. Basically, the dataset is sectioned into three subgroups called:

Training, Testing, and validation sets.

Training set. A data scientist uses a training set to train a model and define the optimal parameters it has to learn from data.

Test set. A test set is needed for an evaluation of the trained model and its capability for generalization. It’s crucial to use different subsets for training and testing to avoid model overfitting, which is the incapacity for generalization we mentioned above.

Model Training

After a data scientist has preprocessed the collected data and split it into train and test can proceed with a model training. This process entails "feeding" the algorithm with training data. An algorithm will process data and output a model that is able to find a target value (attribute) in new data and an answer you want to get with predictive analysis. The purpose of model training is to develop a model.

Model Evaluation and Testing

The performance of the proposed model is evaluated using different metrics like TPR, FPR, FNR, Accuracy, Precision, Recall, and F1 Score.

True Positive Rate (TPR): This metric gauges the relativeness between the various attacks. TPR will return the value of 1 if all the attacks are correctly detected which is always not possible. TPR is also referred to as Detection Rate (DR). The formula for DR is expressed as:

$$TPR = \frac{TP}{TP+FN} \quad \text{(Eq.1)}$$

• **False Positive Rate (FPR):** This metric gauges the relationship between the wrongly detected attacks and the overall number of normal cases. FPR is expressed as:

$$\text{FPR} = \frac{FP}{FP+TN} \quad \text{(Eq. 2)}$$

• **False Negative Rate (FNR):** FNR refers to failure in detecting the attacks and considered it as genuine nodes. The FNR is expressed as:

$$\text{FNR} = \frac{FN}{FN+TP} \quad \text{(Eq. 3)}$$

• **Accuracy:** The overall accuracy of the proposed system in classifying the normal behavior and attacks is expressed as:

$$\text{Accuracy} = \frac{\text{TP+TN}}{TP+TN+FP+FN} \quad \text{(Eq.4)}$$

Precision (P): P is the metric used to measure the quality of the result with the percentage value of the sum of true positives and false positives with the total number of true positives.

• **Recall (R):** Refers to the percentage of total relevant results correctly classified, true positives (TP), divided by the total true positives and false negatives (FN) instances:

• **F1 score:** The F1 score merges both the precision and the recall value as a single measure.

5. RESULTS AND DISCUSSION

This section details the implementation and experimental setup of the proposed model. Python programming version 3.7 is used with the Tensor Flow, Keras, and pandas libraries. The UNSW-NB 15 dataset is given as input for the model. The result shows that intrusion detection is productive using ML algorithms and which algorithm has a high accuracy rate. Figure 6.1 shows the results of preprocessed data using min-max normalization and non-numerical data are converted into numerical data using a Label encoder.

Algorithm: Chi^{2}FSACA

Step 1: Input

Data = Training Data and Testing Data of UNSW_NB15 dataset

Step 2: Preprocessing

(i) Data Encoding

Dff = data.get_dummies()

(ii) Normalization

Dff = (data(num) -np.min(num)/np.max(num)) **Step 3: Feature**

Selection by chi-square Initialize S = {F1...Fn}

For each feature {f} in the training set, Compute chi-square metric using If (x^2 < t hreshold)

S = S - {f}

Else Continue; End For

Step 4: Dataset Splitting

Take training data with the reduced feature set and randomly split it into a training set, validation set, and test set.

Step 5: Classification using four different Machine Language Algorithm

Train the four different models Random Forest, Decision Tree, SVM, and Neural Network with optimized model parameters.

End For

For every test data

Predict the label y for accuracy of each sample End

Display confusion matrix of test data

Step 6: Output

Display the accuracy of each model and performance metrics of test data for each model.

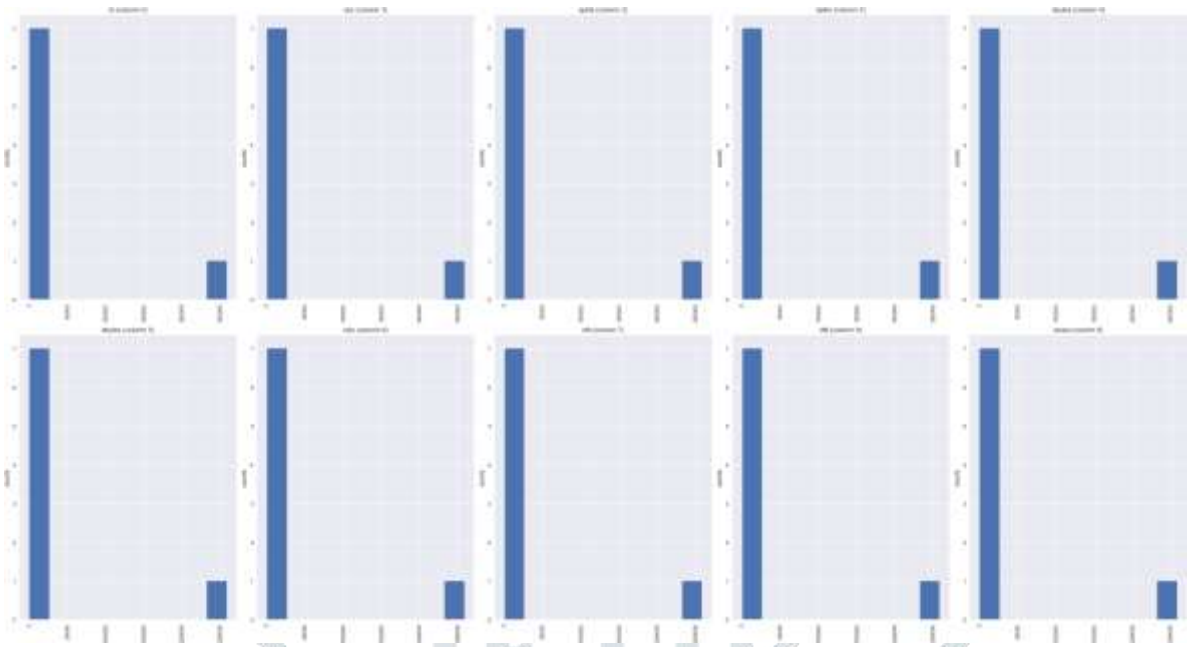


Figure 4.1: Preprocessed data

```
service_5 is IMPORTANT for Detection
service_6 is IMPORTANT for Detection
service_7 is IMPORTANT for Detection
service_8 is NOT an important Detection. (Discard service_8 from model)
service_9 is IMPORTANT for Detection
service_10 is IMPORTANT for Detection
service_11 is IMPORTANT for Detection
service_12 is IMPORTANT for Detection
state_0 is NOT an important Detection. (Discard state_0 from model)
state_1 is NOT an important Detection. (Discard state_1 from model)
state_2 is IMPORTANT for Detection
state_3 is IMPORTANT for Detection
state_4 is IMPORTANT for Detection
state_5 is IMPORTANT for Detection
state_6 is NOT an important Detection. (Discard state_6 from model)
state_7 is IMPORTANT for Detection
state_8 is IMPORTANT for Detection
state_9 is NOT an important Detection. (Discard state_9 from model)
state_10 is NOT an important Detection. (Discard state_10 from model)

In [44]: dff.drop(columns=['Id', 'proto_91', 'service_0', 'state_0', 'state_1',
                'state_6', 'state_9', 'state_10'],inplace = True)
```

Figure 4.2: Feature Selection

Figure 4.2 shows the results of Feature selection using the chi-square algorithm, where it eliminates 10 unimportant features, which are not necessary classification processes, and selected attributes are split and trained with models.

Figure 4.3 shows the results of the Accuracy and Confusion Matrix of the ML algorithm. Here we used Random Forest, SVM, Neural Network, and Decision Tree for the classification process. And Figure 4.4 shows the error values of the ML algorithms.

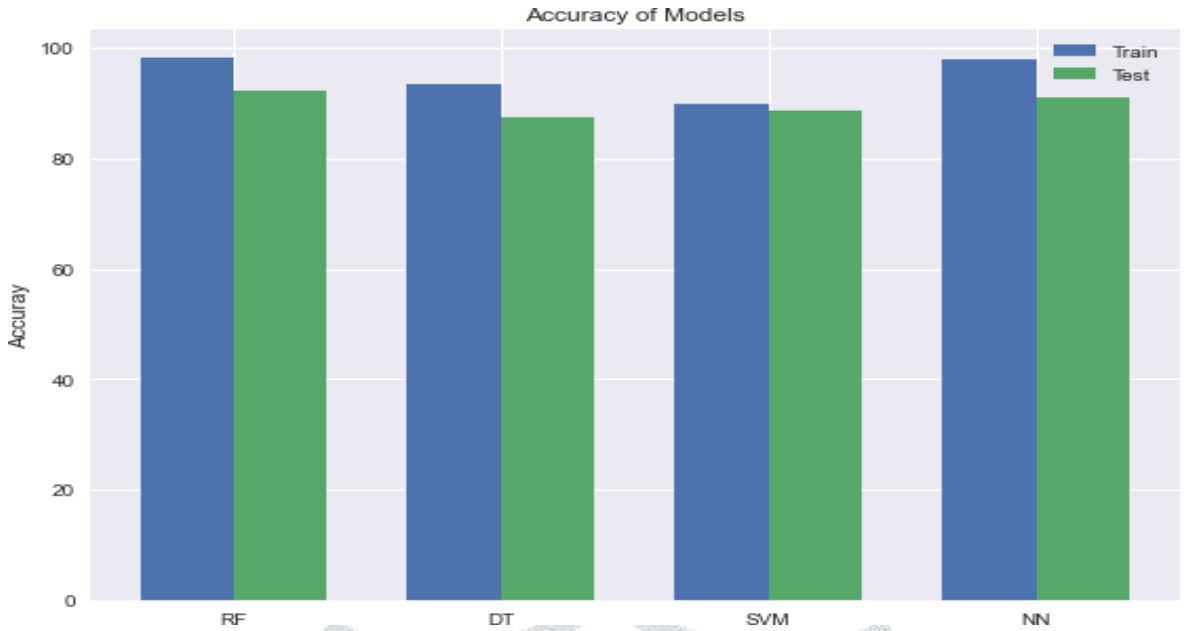


Figure 4.3: Training vs Testing Accuracy

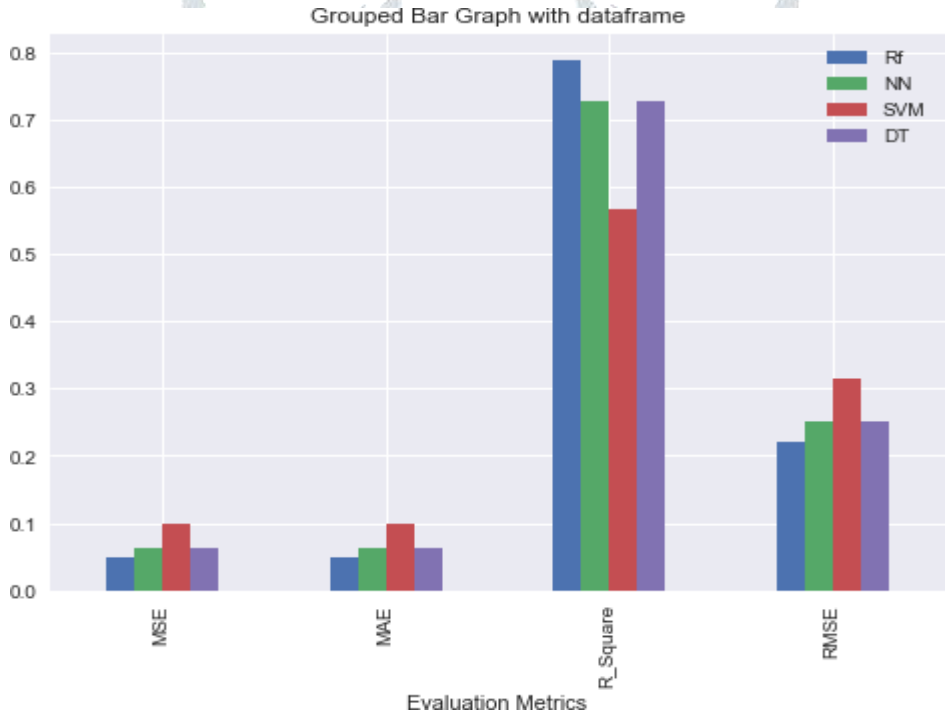


Figure 4.4: Evaluation Metrics of Model

6. CONCLUSION

The proposed system trains the intrusion detection model to detect the attacker/malicious user using the chi-square algorithm for effective choice of preferred or the necessary features to reduce the processing time of the entire features. The suitable feature election helps the system to identify the attackers efficiently which can lead to a reduction in the false-positive rate. The selected features are further used for the classification process. The proposed model uses machine learning classification algorithms, using this model, we can able to classify the normal user and malicious users and can able to prevent the entry of unauthorized or crook vehicles into the

communication network. The results of the experiments show the performance of the proposed model. For future work, we propose to focus on other types of attacks with the same and different machine learning techniques and deep learning techniques.

REFERENCES

[1] Deepthy K Denatious & Anita John, "Survey on Data Mining Techniques to Enhance Intrusion Detection", International Conference on Computer Communication and Informatics (ICCCI 2012).

[2] R.China Appala Naidu and P.S.Avadhani, "A Comparison of Data Mining Techniques for Intrusion Detection", International Conference on Advanced Communication Control and Computing Technologies (ICACCCT).

[3] Mohammadreza Ektefa, Sara Memar, Fatimah Sidi and Lilly Suriani Affendey, "Intrusion Detection Using Data Mining Techniques", pp 200-203, IEEE, 2010.

[4] M. Al-Zewairi, S. Almajali, and A. Awajan. 2017. Experimental Evaluation of a Multi-layer Feed-Forward Artificial Neural Network Classifier for Network Intrusion Detection System. 2017 International Conference on New Trends in Computing Sciences (ICTCS), Amman, Jordan, pp. 167-172, IEEE

[5] L. Breiman. 2001. Random Forests. Machine Learning, 45(1), pp. 5-32. V. Chandola, A. Banerjee, and V. Kumar. 2009. Anomaly Detection: A Survey. ACM Computing Surveys, 41(3), pp. 1-15.

[6] Coelho, F., A. Braga and M. Verleysen. "Cluster homogeneity as a semi-supervised principle for feature selection using mutual information." ESANN 2012.

[7] M.Govindarajan and Rlvl.Chandrasekaran, "Intrusion Detection Using k-Nearest Neighbor" pp 13-20, ICAC, IEEE, 2009

[8] R. Di Pietro and L. V. Mancini, eds.. "Intrusion Detection Systems. "Springer Science & Business, vol. 38. Media 2008.

[9] H. Gharaee and H. Hosseinvand, "A new feature selection IDS based on genetic algorithm and SVM, "8th International Symposium on Telecommunications (IST), Tehran, 2016, pp. 139-144, doi: 10.1109/ISTEL.2016.7881798, 2016.

[10] Osama Faker. "Intrusion Detection Using Big Data and Deep Learning Techniques". MS Thesis, Cankaya University, 2019.

[11] Primartha, Rifkie and Bayu Adhi Tama. "Anomaly detection using random forest: A performance revisited." 2017 International Conference on Data and Software Engineering (ICoDSE) (2017): 1-6, 2018.

[12] Sumaiya, Thaseen Ikram, Aswani Kumar, and Cherukuri. “Intrusion Detection Model using Fusion of Chi-Square Feature Selection and multi class SVM.” Journal of King Saud University - Computer and Information Sciences 2017.

[13] Ondrej Linda, Todd Vollmer, and Milos Manic. “Neural Network Based Intrusion Detection System for Critical Infrastructures.” International Joint Conference on Neural Networks, 2009.

[14] P Amudha and H Abdul Rauf, “Performance Analysis of Data Mining Approaches in Intrusion Detection”, IEEE, 2011.

[15] Roshan Chitrakar and Huang Chuanhe, “Anomaly based Intrusion Detection using Hybrid Learning Approach of combining kMedoids Clustering and Naïve Bayes Classification”, IEEE, 2012.

[16] Dahiya, Priyanka, and Devesh Kumar Srivastava. “Network Intrusion Detection in Big Dataset Using Spark.” Procedia Computer Science. Vol. 132. Elsevier B.V., 2018. 253–262. Procedia Computer Science. Web.

[17] L. Dhanabal, and S.p.Shantharajah. “A Study on NSL KDD Dataset for Intrusion Detection System based on Classification Algorithms". International Journal of Advanced Research in Computer and Communication Engineering, 2015, 4(6), pp. 446-452.

[18] G.Mahalakshmi and E.Uma, “Machine Learning-Based Feature Selection For Intrusion Detection System in VANET”, International e-Conference on Artificial Intelligence, Network Security and Data Science, IeCAN-2020, December 2020.

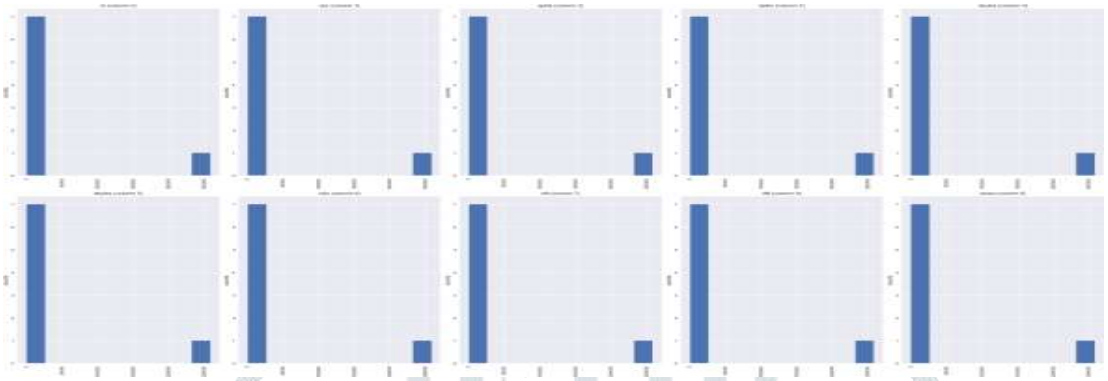
[19] G.Mahalakshmi, E.Uma, M.Aroosiya and M.Vinitha, “ Intrusion Detection System Using Convolutional Neural Network on UNSW-NB15 Dataset”, International Conference on Advances in Parallel Computing Technologies and Applications (ICAPTA 2021), IOS Press 2021.

ANNEXURE

SCREENSHOTS

Normalization :

The following screenshot shows the results of preprocessed data using min-max normalization and non-numerical data are converted into numerical datas using Label encoder.



Feature Selection

The following screenshot shows the results of Feature selection using chi-square. We eliminate 10 features which are not important for the classification process and selected features are split and trained with models.

```
service_5 is IMPORTANT for Detection
service_6 is IMPORTANT for Detection
service_7 is IMPORTANT for Detection
service_8 is NOT an important Detection. (Discard service_8 from model)
service_9 is IMPORTANT for Detection
service_10 is IMPORTANT for Detection
service_11 is IMPORTANT for Detection
service_12 is IMPORTANT for Detection
state_0 is NOT an important Detection. (Discard state_0 from model)
state_1 is NOT an important Detection. (Discard state_1 from model)
state_2 is IMPORTANT for Detection
state_3 is IMPORTANT for Detection
state_4 is IMPORTANT for Detection
state_5 is IMPORTANT for Detection
state_6 is NOT an important Detection. (Discard state_6 from model)
state_7 is IMPORTANT for Detection
state_8 is IMPORTANT for Detection
state_9 is NOT an important Detection. (Discard state_9 from model)
state_10 is NOT an important Detection. (Discard state_10 from model)
```

```
In [44]: dff.drop(columns=['10', 'proto_91', 'service_0', 'state_0', 'state_1',
                'state_6', 'state_9', 'state_10'],inplace = True)
```

Classification Results:

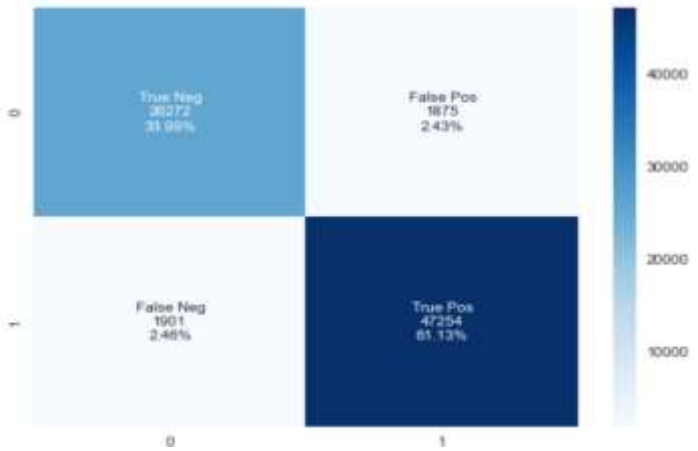
The following screenshot shows the results of the confusion matrix to summarize the performance: they give direct comparisons of values like True Positives, False Positives, True Negatives and False Negatives for four different machine learning algorithms.

1. **Random Forest: Confusion Matrix**

```
Accuracy: 98.28
Accuracy of CV: 92.16
Execution time: 236.59853649139404
Confusion matrix

 [[26272  1875]
 [ 1901 47254]]
Report
```

	precision	recall	f1-score	support
0	0.93	0.93	0.93	28147
1	0.96	0.96	0.96	49155
accuracy			0.95	77302
macro avg	0.95	0.95	0.95	77302
weighted avg	0.95	0.95	0.95	77302

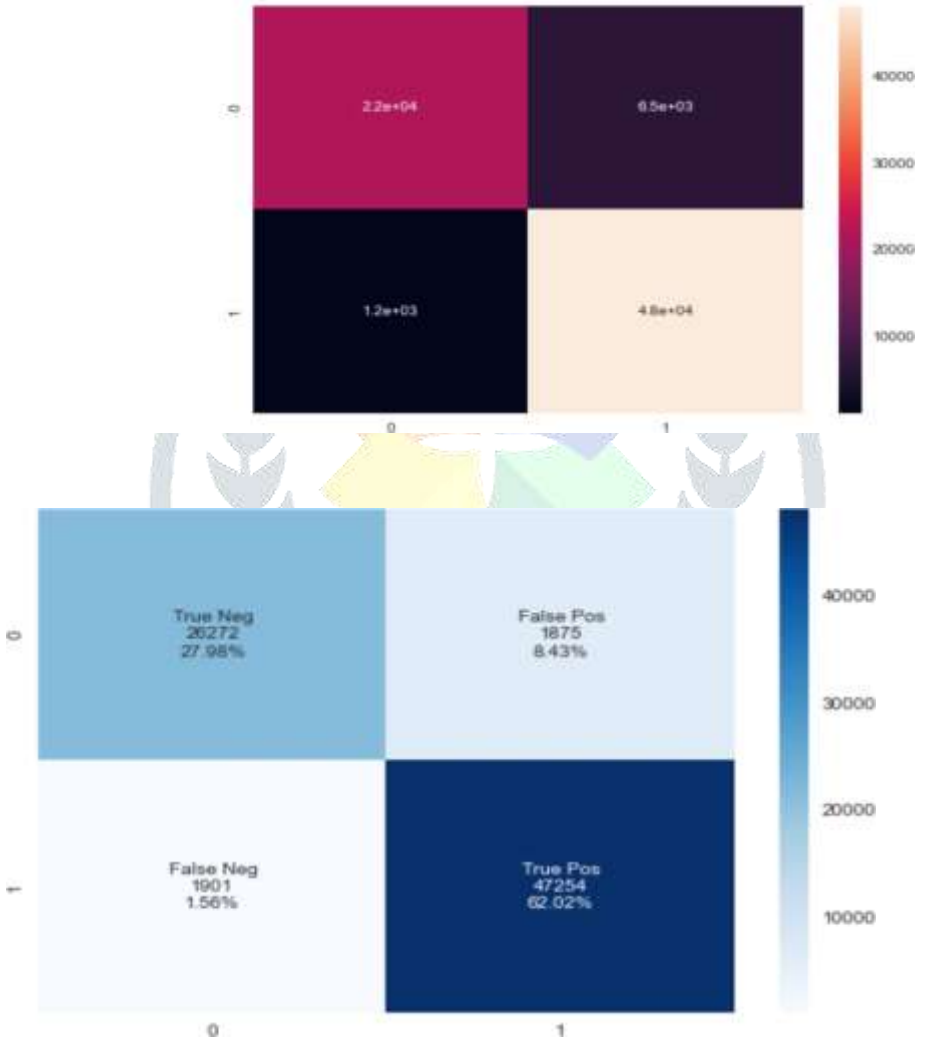
Out[54]: <AxesSubplot:>



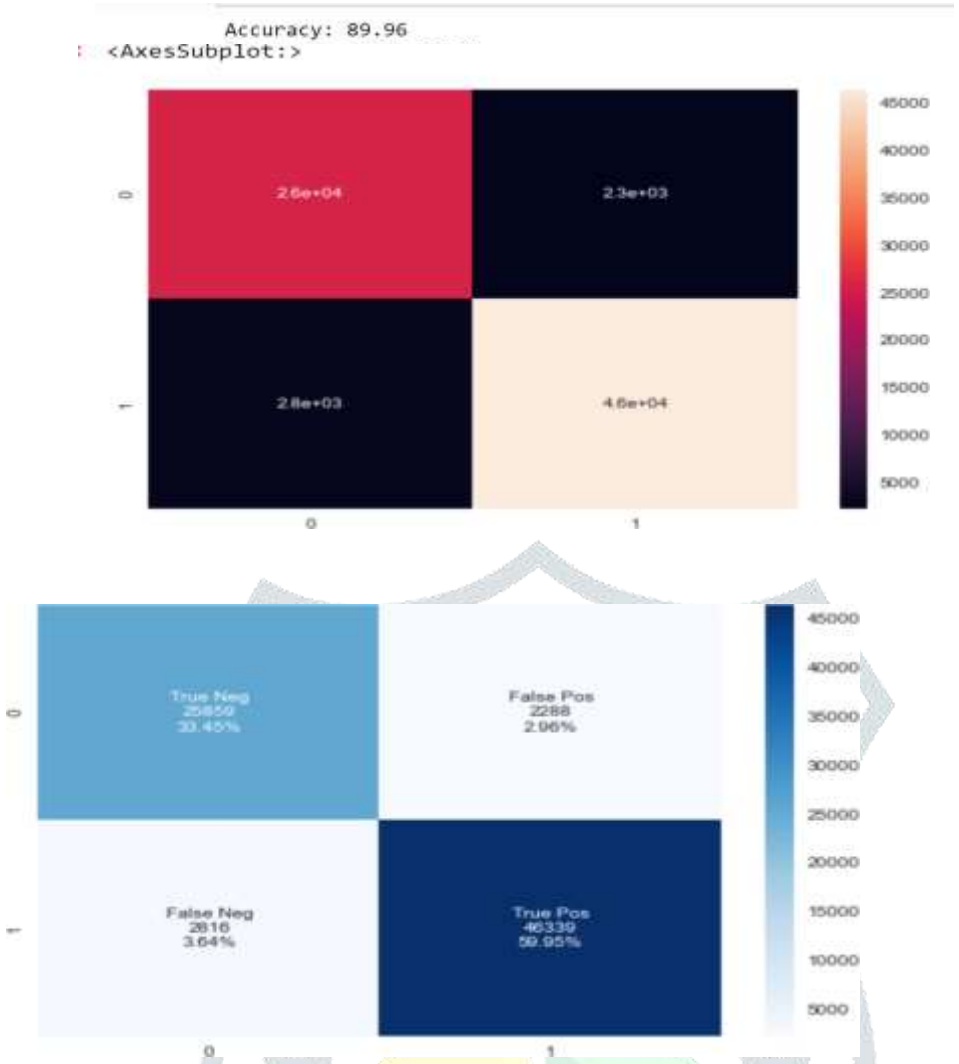
2. Neural Network: Confusion Matrix

```
Accuracy: 92.98
Accuracy of CV: 89.9
Execution time: 965.9254505634308
Confusion matrix

 [[25593  2554]
 [ 2571 46584]]
              precision    recall  f1-score   support

           0       0.91      0.91      0.91     28147
           1       0.95      0.95      0.95     49155

    accuracy                           0.93     77302
   macro avg       0.93      0.93      0.93     77302
weighted avg       0.93      0.93      0.93     77302
```

Out[60]: <AxesSubplot:>



3. SVM : Confusion Matrix

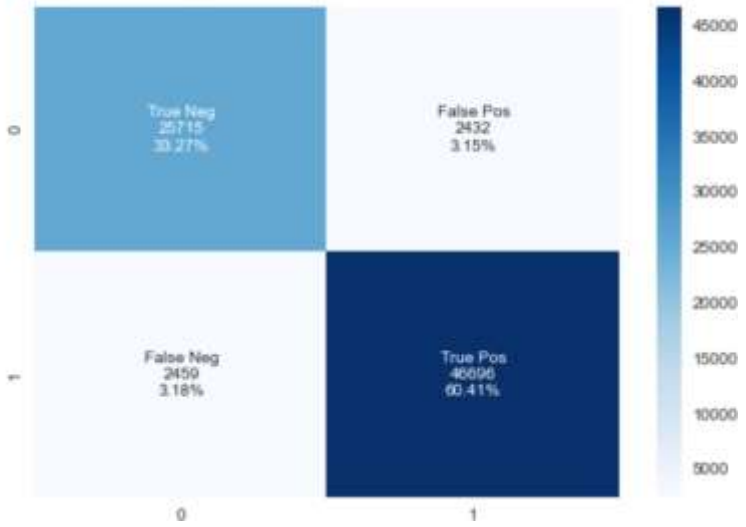


4. Decision Tree: Confusion Matrix

```
Accuracy: 97.95
Accuracy of CV: 91.11
Execution time: 34.18578052520752
Confusion matrix

 [[25715  2432]
 [ 2459 46696]]
              precision    recall  f1-score   support

           0       0.91      0.91      0.91     28147
           1       0.95      0.95      0.95     49155

    accuracy                           0.94     77302
   macro avg       0.93      0.93      0.93     77302
weighted avg       0.94      0.94      0.94     77302
```

Out[67]: <AxesSubplot:>

Out[70]: <AxesSubplot:>

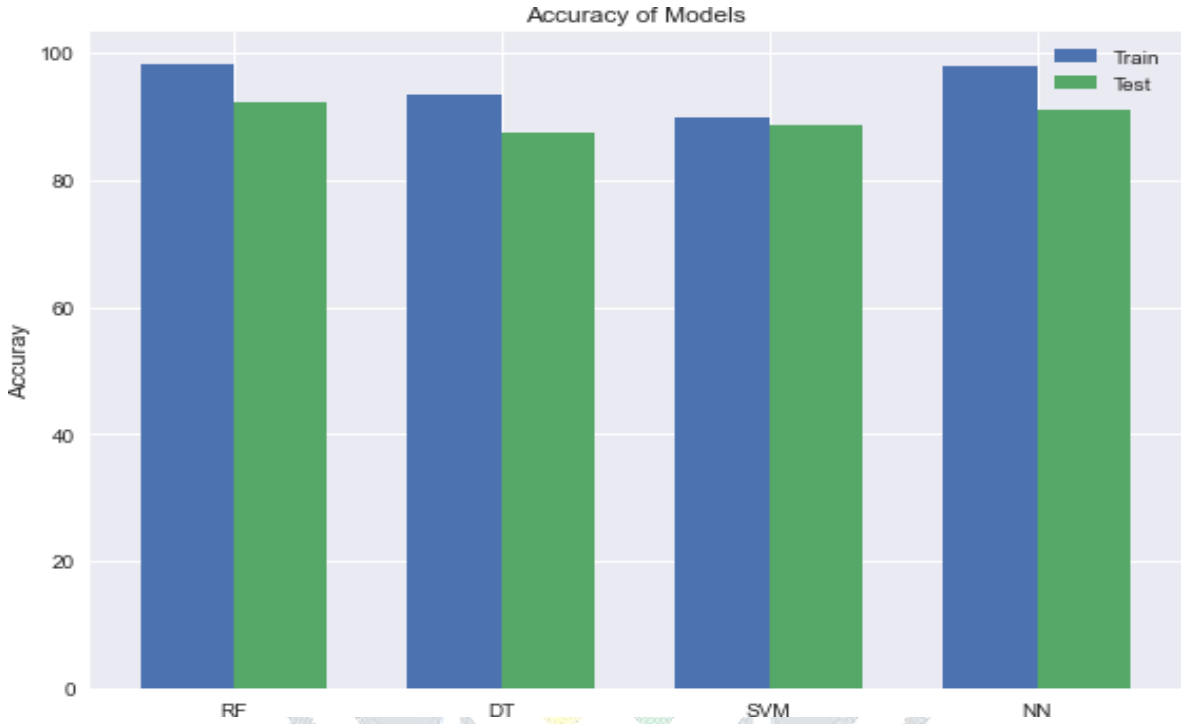


The following screenshot shows the overall confusion matrix for four different ML algorithms.

Out[93]:

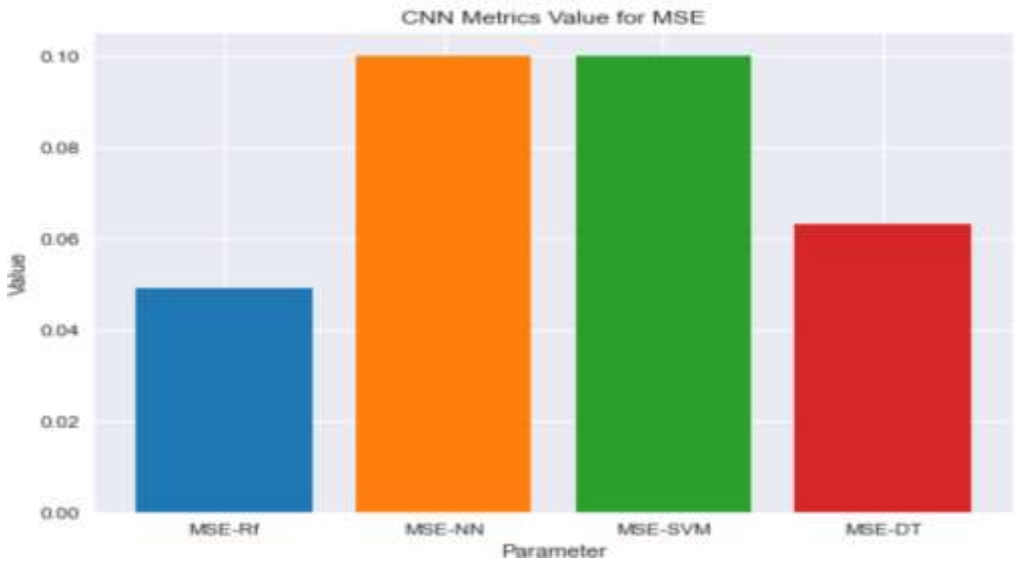
	Algorithm	Accuracy %	Execution Time	Model Accuracy	Recall[0]	Recall[1]	Precision Negative	Precision Positive
0	RandomForest	92.19	193.389472	0.951049	0.932781	0.961510	0.932781	0.961510
1	Decision Tree	91.11	31.412771	0.936729	0.913596	0.949975	0.912721	0.950497
2	Neural Network	89.90	965.925451	0.933702	0.909262	0.947696	0.908713	0.948024
3	SVM	88.66	45.769536	0.900041	0.768430	0.975404	0.947062	0.880325

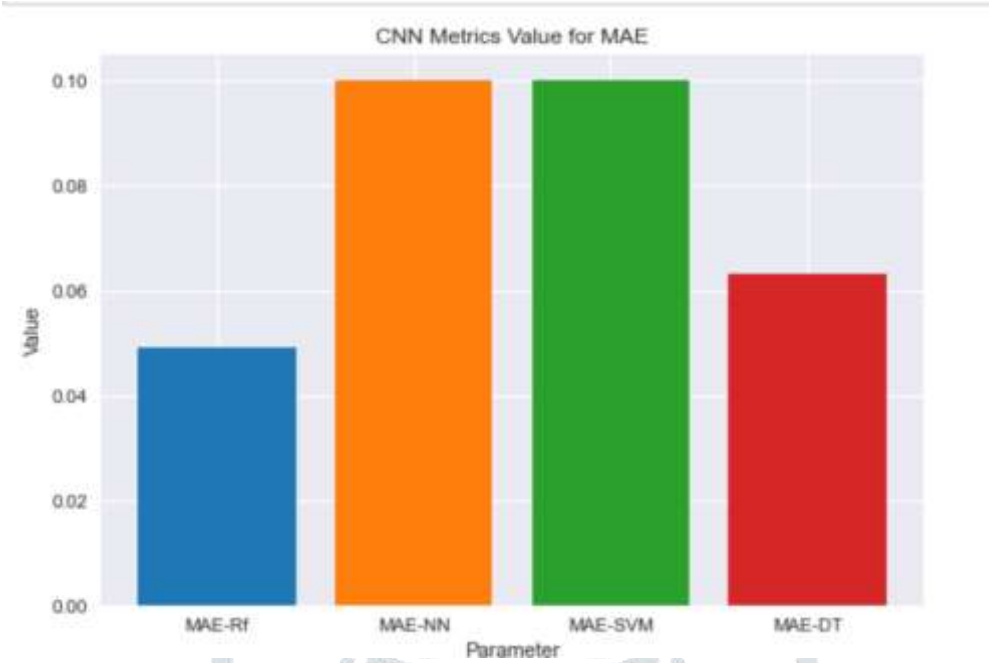
The following screenshot shows the results of Accuracy and Confusion Matrix of ML algorithm. Here we used Random Forest, SVM, Neural Network, Decision Tree for classification process and showed that Random Forest is the best classifier in terms of accuracy detection.

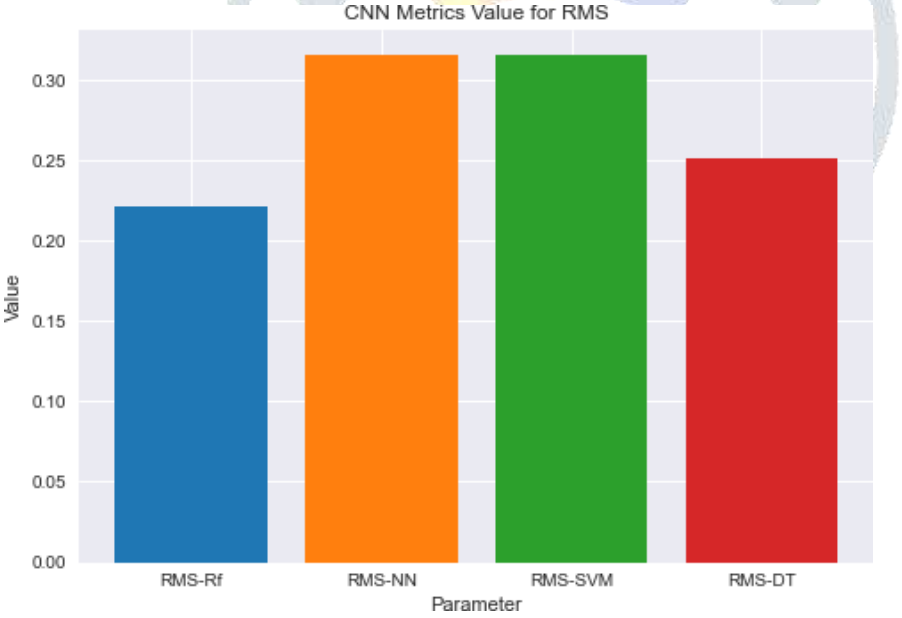


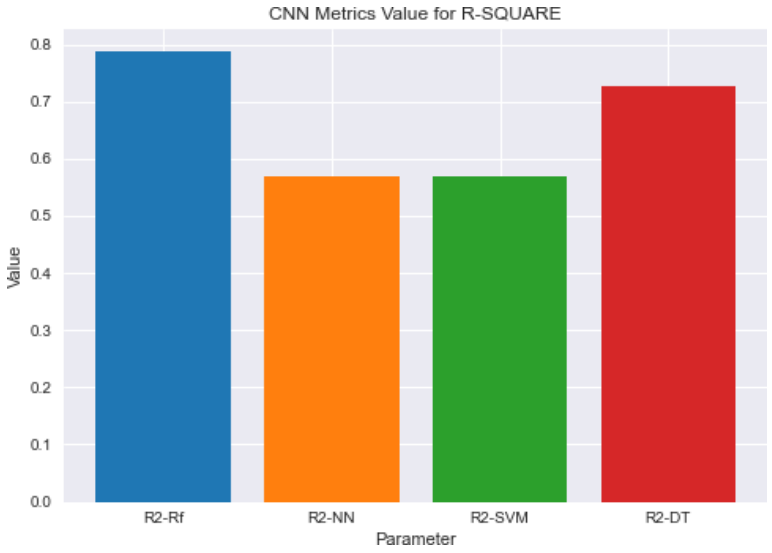
Evaluation Metrics:

The following screen shows the evaluation metrics using four different ML algorithms. Here we used MAE, MSE, R-Square, and RMSE metrics to determine the algorithm performance.




CNN Metrics Value for MAE
Value
0.10
0.08
0.06
0.04
0.02
0.00
MAE-Rf
MAE-NN
MAE-SVM
MAE-DT
Parameter


CNN Metrics Value for RMS
Value
0.30
0.25
0.20
0.15
0.10
0.05
0.00
RMS-Rf
RMS-NN
RMS-SVM
RMS-DT
Parameter



The following screenshot shows the overall evaluation metrics in tabular form for four different ML algorithms.

Out[81]:

	Algorithm	Accuracy %	MSE	MAE	R-SQUARE	RMSE
0	RandomForest	92.16	0.048847	0.048847	0.789029	0.221014
1	Decision Tree	91.11	0.063271	0.063271	0.726732	0.251538
2	SVM	88.66	0.099959	0.099959	0.568280	0.316162
3	Neural Network	87.43	0.063271	0.063271	0.726732	0.251538

The following screenshot shows overall error values of the ML algorithms.