International Journal of Science and Research (IJSR)
ISSN: 2319-7064
SJIF (2020): 7.803

Intrusion Detection using Machine Learning Techniques

Akshay Kaushik[1], Varun Goel[2]

[1]Department of Information Technology, Maharaja Agrasen Institute of Technology, (Affiliated to GGSIPU) New Delhi, India
kaushikakshay359[at]gmail.com

[2]Assistant Professor, Department of Information Technology, Maharaja Agrasen Institute of Technology, (Affiliated to GGSIPU) New Delhi, India
varungoel.cs[at]gmail.com

Abstract: ***An Intrusion is an uncredited access to a computer in your organization or a personal computer. As the world is becoming more internet-oriented and data leaks occur more than ever in our tech-savvy world, we need to know about these attacks so that they can be prevented hence coming into action Intrusion Detection System. IDS are systems that alert about the attack by analyzing the traffic on the network for signs of unauthorized activity. To identify the attack and alert about that possible attack, this system needs to be trained on some previous attacks data, for this study, the improved version of the KDD99 dataset, NSL-KDD dataset have been used for training the Machine Learning Model. In this analysis of Machine Learning algorithms, the algorithms under consideration are Logistic Regression, Support Vector Machine, Decision Tree, Random Forest. For comparison of the performance of the algorithms metrics like Accuracy Score, Confusion Matrix, and Classification Report were considered to find the best algorithm among them.***

Keywords: Machine Learning, Intrusion Detection, Algorithm, Dataset, NSL-KDD, Attacks

1. Introduction

In recent times, with the increase in the use of the Internet, progress in technology, and a large number of data breaches, Network Security has become an important topic of research. With the availability of data to a larger range of audiences, privacy and integrity of data have to be provided. Whenever it is attacked by something or someone, this action is often identified as an intrusion. The network intrusion or an attack can be classified into four classes:

1) Denial of Service Attacks (DoS) Attack
2) Probing Attack
3) U2R attacks(User to Root)
4) R2L attacks

To prevent these attacks, we need to know about them before they strike so, we need an Intrusion Detection System that alerts the owner of the attack. An intrusion detection system (IDS) goes through the activity on the network to find possible intrusions. Intrusion Detection is possible when we have the model to predict the possibility of an attack, for the model should be trained on data of previous attacks. In this study, the NSL-KDD dataset has been used for training different Machine Learning models.

NSL-KDD dataset is the improved version of the KDD99 dataset, from which duplicate values were removed to get rid of biased results of classification.

Machine Learning Algorithms that are considered for this study are Logistic Regression, Support Vector Machine, Decision Tree, and Random Forest. In this paper, research is more intended to find the best algorithm among mentioned algorithms to classify the attack. The selection of an algorithm to predict the result is the most pivotal role in coming up with a sound solution for the intrusions. The study gives an idea about which machine learning algorithm should be used by the Intrusion Detection System that will best identify the deviation in the network.

The paper is organized as follows Section II discussed work of other authors on Intrusion Detection System and Machine Learning Algorithms.

Section III consists of the dataset details and methodology followed to get desired results.
Section IV has the discussion about different Metrics to measure the performance of the model.
Section V explains the results of the implementation of classifiers and shows the results using metrics discussed in Section IV.
Section VI discussed the conclusion of this study and what future work can be done in this paper to get better results.

2. Literature Review

Network Security is one of the vital research topics, several other authors have worked on this topic and found different insights.

Most of the studies on the Intrusion Detection System use the KDD99 dataset, The KDD99 dataset consists of 41 features obtained by preprocessing from the DARPA dataset in 1999. It consists of almost 5 M and 2 M instances for training and testing respectively.

The author [1] studied the effectiveness of the dataset, reviewed the datasets and performance evaluation methods on these datasets.

Author [2] have also utilized the NSL-KDD dataset and studied a new model that can be used to estimate the intrusion scope threshold degree based on the network transaction data's optimal features that were made available for training

Author [3] have featured a combined 2 data mining algorithms Decision Tree and SVM in their paper and the main target was to combine the advantages of both the algorithms.

Author [4] had an experiment aiming to understand the implications of using supervised machine learning techniques on intrusion detection and results showed that Random Forest Classifier worked best for that dataset. Similar Studies have been done by many other researchers also.

3. Research Methodology

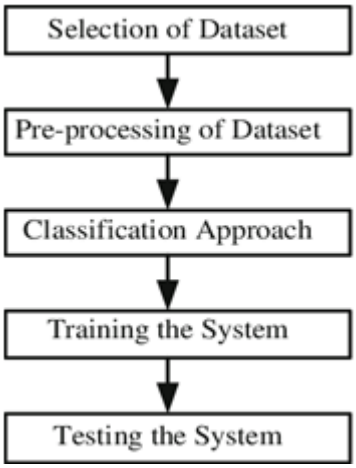
a) Dataset

The NSL-KDD dataset is collected, NSL-KDD is the improved version of KDD99.The NSL-KDD dataset has various version available on the internet, the version we have used have number of instances in the training dataset: 125973, number of instances in the test dataset: 22544

This dataset has the following advantages:

- It does not consist of recurring instances in the train set, which makes the classifier less biased towards some attacks.
- There are no null values available in the dataset
- It does not consist not necessary instances in the training set, so the classifiers will not be partial towards more duplicate records.

b) Methodology



The process started with the collection of the dataset, after collection pre-processing on the dataset is performed in which data is checked for null values, missing values, out of domain values. There were none of the above anomalies in the dataset. The distribution of different types of attacks was checked and found that attacks like a spy, Perl, phf, multihop, ftp_write, load module, have instances less than 10, so were moved these since there will not be sufficient training data for the Machine Learning Model.

In the dataset, there were three data type attributes int(22 attributes), float(15 attributes), and object(3 attributes). Int, a float is ready for training the model while object (categorical values) type attributes needed encoding to numerical values so it could be used for the training of the model also. For encoding, Label Encoder and One Hot Encoding are used in this study. After conversion, the newly created attributes are concatenated with the rest of the columns.

There were NaN(Not a Number) values in a few attributes after concatenation, they were replaced with the mode(statistical mode: Higher number of occurrences in the column) of that column because the values were only 0 and 1 in those attributes taking mean or standard deviation is not right, and removing the NaN(Not a Number) values would have decreased the training dataset significantly.

Now dataset was free from anomalies and the attributes were either integer or float, ready to be normalized for training the models. Min-Max and Standard Scaler were used to normalize the data in two different instances and models were trained; it was found that the dataset normalized with Standard Scaler produced output with more accuracy score.

$$Standard\ \ Scaler\ (z) = \frac{(x - u)}{s}$$

x=current value
u=mean value
s=standard deviation

As our dataset is normalized and ready for training, the Implementation of the Machine Learning algorithms mentioned above is done using python library scikit-learn.

1) Logistic Regression

```
clf = LogisticRegression()
clf.fit(X_train, Y_train)
pred = clf.predict(X_test)
```

2) Support Vector Machine(SVM):

```
clf1=LinearSVC(random_state=0)
clf1.fit(X_train, Y_train)
predsvm = clf1.predict(X_test)
```

3) Decision Tree Classifier:

```
clf2=DecisionTreeClassifier(random_stat
e=0)
clf2.fit(X_train, Y_train)
predDT = clf2.predict(X_test)
```

4) Random Forest Classifier:

```
clf3 =
RandomForestClassifier(random_state=40,
n_estimators=300,n_jobs=-1)
clf3.fit(X_train, Y_train)
predRF = clf3.predict(X_test)
```

After implementation, the performance of all the classifiers was measured using various metrics like accuracy score, confusion matrix, and classification report.

4. Metrics and Performance Evaluation

Before moving to the measure of the performance of model we need to know a few terms, terms are:

1) True Positive (tp):True Positive means when model predicted the instance positive and it was positive in y_true also.
2) True Negative (tn): True negative is when model correctly predicts the negative class of the dataset.
3) False Positive (fp): False positive is when model incorrectly predicts the positive class.
4) False Negative (fn): False negative is when model incorrectly predicts the negative class.

Confusion Matrix is a table that is used to measure the performance of the model (classification model)

After implementation, the performance of the model was measured using the following metrics:

1) Accuracy score: It is defined as the ratio of the sum of true positive and true negative to all predictions. The Calculation formula is given below:

$$Accuracy = \frac{(tp+tn)}{tp+tn+fp+fn}$$

2) Precision: Precision is the ratio of true positive to the sum of true positive and false positive. Formula is given below:

$$Precision = \frac{tp}{tp+fp}$$

3) Recall: Recall is the ratio of true positive to the sum of true positive and false negative. Formula is given below:

$$Recall = \frac{tp}{tp+fn}$$

4) Support: The support is the number of occurrences of each class in y_true.

Classification Report:
The classification report shows the values of precision, recall, F1-score, and support scores of the classifier.

5. Results

After implementation and performance measurement of the four classifiers used in this paper, their results are as follows:

1) Logistic Regression:

```
Accuracy of Logistic Regression is 0.777.
Confusion Matrix is:
 [[8897 3909]
 [1113 8598]]
Classification Report is:
              precision    recall  f1-score   support

      attack       0.89      0.69      0.78     12806
      normal       0.69      0.89      0.77      9711

    accuracy                           0.78     22517
   macro avg       0.79      0.79      0.78     22517
weighted avg       0.80      0.78      0.78     22517
```

2) Support Vector Machine:

```
Accuracy of SVM is 0.7581.
Confusion Matrix is:
 [[8763 4043]
 [1404 8307]]
Classification Report is:
              precision    recall  f1-score   support

      attack       0.86      0.68      0.76     12806
      normal       0.67      0.86      0.75      9711

    accuracy                           0.76     22517
   macro avg       0.77      0.77      0.76     22517
weighted avg       0.78      0.76      0.76     22517
```

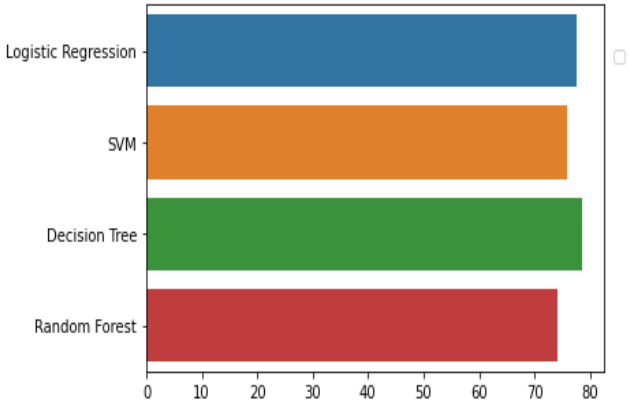
3) Decision Tree Classifier

```
Accuracy of DT is 0.7859.
Confusion Matrix is:
 [[8641 4165]
 [ 657 9054]]
Classification Report is:
              precision    recall  f1-score   support

      attack       0.93      0.67      0.78     12806
      normal       0.68      0.93      0.79      9711

    accuracy                           0.79     22517
   macro avg       0.81      0.80      0.79     22517
weighted avg       0.82      0.79      0.79     22517
```

4) Random Forest Classifier

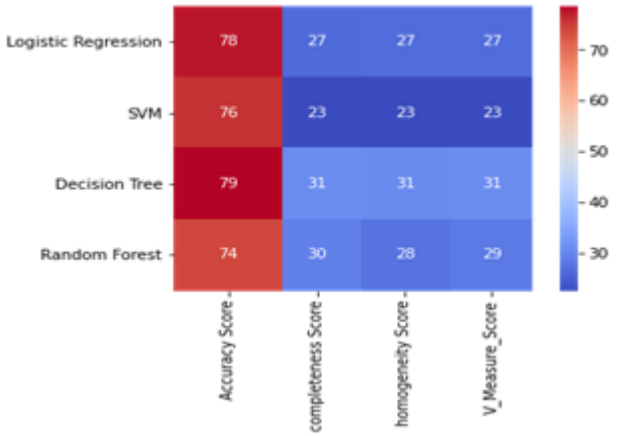
```
Accuracy of RF is 0.7418.
Confusion Matrix is:
 [[7310 5496]
 [ 319 9392]]
Classification Report is:
              precision    recall  f1-score   support

      attack       0.96      0.57      0.72     12806
      normal       0.63      0.97      0.76      9711

    accuracy                           0.74     22517
   macro avg       0.79      0.77      0.74     22517
weighted avg       0.82      0.74      0.74     22517
```

6. Conclusions and Future Work



1Bar graph showing Accuracy Score

The analysis of multiple classification models like Support Vector Machine, Logistic Regression, Decision Tree, Random Forest for anomaly intrusion detection system is done. The performance of these models has been observed and studied on the basis of their accuracy and precision on the test data. The experiments proved that the classifiers are capable of handling high-dimensional data and still produce accurate results. The results indicate that the ability and accuracy of the Decision Tree classifier outperform that of Others. The Random Forest and SVM took higher time in training and testing of the dataset as compared to Logistic Regression and Decision Tree. The accuracy in the results produced using Random Forest is lowest amongst all classifiers.



2 Heat map showing all metrices for all four algorithms

This work can be continued by finding the best features to select before feeding it to the model using various Feature Selection Techniques such as Wrapper Methods, Filter Methods, etc. Also, the analysis can be continued on newer datasets that have cutting-edge attacks information and get the best classifier to predict the possibility of a network intrusion.

References

[1] A. A. Aburomman and M. B. I. Reaz, "A survey of intrusion detection systems based on ensemble and hybrid classifiers," Comput. Secur., vol. 65, pp. 135–152, 2017

[2] Aljawarneh, M. Aldwairi, and M. B. Yassein, "Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model," J. Comput. Sci., vol. 25, pp. 152– 160, 2018.

[3] P.Sangkatsanee, N. Wattanapongsakorn and C. Charnsripinyo,, "Practical Real-Time Intrusion Detection Using Machine Learning Approaches, Computer Communications" , vol. 34, no. 18, pp. 2227–2235, (2011).

[4] J. Manjula C. Belavagi and Balachandra Muniyal, "Performance Evaluation of Supervised Machine Learning Algorithms for Intrusion Detection" Twelfth International Multi-Conference on Information Processing- 2016.

[5] Ibrahimi, K., Ouaddane, M.: Management of intrusion detection systems based-KDD99: analysis with LDA and PCA. In: 2017 International Conference on Wireless Networks and Mobile Communications (WINCOM), pp. 1–6. IEEE (2017).

[6] Modi, U., Jain, A.: An improved method to detect intrusion using machine learning algorithms. Inf. Eng. Int. J. (IEIJ) 4(2), 17–29 (2016). https://doi.org/10.5121/ieij.2016.4203

[7] Moustafa, N.; Slay, J. The evaluation of Network Anomaly Detection Systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set. Inf. Syst. Secur. 2016, 25, 18–31.

[8] Al Tobi, A.M.; Duncan, I. KDD 1999 generation faults: A review and analysis. J. Cyber Secur. Technol. 2018, 2, 164–200.

[9] Nadiammai, G.V.; Hemalatha, M. Effective approach toward Intrusion Detection System using data mining techniques. Egypt. Inf. J. 2014, 15, 37–50

[10] Patcha, A., Park, J.M.: An overview of anomaly detection techniques: existing solutions and latest technological trends. Comput. Netw. 51(12), 3448–3470 (2007)