Intrusion Detection

by Abdul Rehman

Intrusion Detection Using Machine Learning on NSL-KDD dataset

Dataset

The proposed work used the NSL KDD dataset for the classification of intrusion attacks from Kaggle. NSL KDD dataset is the upgraded version of KDD 19 dataset. It is not included the unnecessary information and developed to solve the problems facing with KDD 19 dataset. As the NSL-KDD dataset not included the duplicated records and unnecessary information, the dataset only contains the 150,000 samples unlike the 5 million samples in KDD 19 dataset. The original dataset was also based on the 41 different features of 150,000 samples labeled with the associated intrusion attack. The labeled column contains the 40 unique values that represent the 40 types of intrusion attacks. Collectively, the NSL-KDD dataset is based on 40 types of 150,000 samples based on 41 features. The proposed work used the NSL KDD dataset for the classification of intrusion attacks using machine learning models.

Preprocessing

In the preprocessing of the dataset, firstly the 40 classes (attack types) were mapped to the 5 major attack types including the normal, U2R, R2L, Prob and DoS. Table 1 showed the mapping of the attack types on the majority class type.

Table 1: Criteria of Mapping attacks on Majority Class.

Attack Type	Majority Class	Attack Type	Majority Class	Attack Type	Majority Class
Back	DoS	Perl	U2R	Warez client	R2L
Land	DoS	Rootkit	U2R	Warez master	R2L
Neptune	DoS	ftp-write	R2L	Ipsweep	Prob
Pod	DoS	Guess-passwd	R2L	Nmap	Prob
Smurf	DoS	IMAP	R2L	Portsweep	Prob
teardrop	DoS	Multihop	R2L	Satan	Prob
Buffer overflow	U2R	PHF	R2L	Normal	Normal
Load module	U2R	Spy	R2L		

After the mapping of 22 attack types on the 5-majority class, all the samples of unnecessary classes were removed from dataset. The dataset also found by the 3 categorical features labeled as protocol-type, service and flag. As the machine learning models only interpreted the numerical value, the categorical features were converted into the numerical representation. The categorical features were converted into numerical features using the built-in label encoder function of scikit-learn library.

By following the preprocessing steps, we apply min max scaling on the input variables of the dataset. Min Max Scaling technique convert the all values of the dataset in the range of 0-1. For Min Max Scaling, we used the min-max-scaling function of scikit-learn library.

Train Test Split

By following the preprocessing of the dataset, the dataset was divided into two different groups labeled as training set and testing set. The dataset was divided into two different groups with the percentage of 80% and 20% respectively. The train test split built-I function of scikit-learn library was used for splitting the dataset into different groups. The newly training and testing set contained the 13694 and 5870 samples.

Machine Learning Models

For the classification of intrusion attacks, we used different machine learning models. We used the different variants of SVM, Decision Tree and K Nearest Neighbor (KNN). The variants and the parameters specific to that variant of machine learning models is presenting in Table 2.

Table 2: Machine Learning Models for attacks Classification.

Model Name	Model Variant	Parameter
SVM	Linear	Kernel: linear
SVM	Quadratic	Kernel: Poly, Degree: 2
SVM	Cubic	Kernel: Poly, Degree: 3
KNN	Fine	N-Neighbors: 1
KNN	Medium	N-Neighbors: 10
KNN	Cubic	N-Neighbors: 10, Metric: Cosine
DT	Fine	Max-leaf-nodes= 100
DT	Medium	Max-leaf-nodes= 20

We trained all the described model in Table 2 with the training set of NSL-KDD dataset. By the end of each model training, the models were evaluated using the testing set. For the evaluation of the models, different well known evaluation measures including the accuracy, precision, recall and f1-score were used. All the evaluation measures were calculated using the built-in function of scikit-learn library and all measures were calculated by using equation (1 – 4) respectively.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad \text{Eq. 1}$$

$$Precision = \frac{TP}{TP+FP} \quad \text{Eq. 2}$$

$$Recall = \frac{TP}{TP+FN} \quad \text{Eq. 3}$$

$$F1Score = \frac{2*(recall*Precision)}{Recall+Precision} \quad \text{Eq. 4}$$

Models Results

In the result section, the result of the trained models, classification reports and confusion matrices will be presented. Lastly, the result of proposed models and comparison study was performed.

SVM Model

SVM is very efficient for the classification of high dimensional data. In the proposed study, we trained the three variants of SVM including the Linear, Quadratic, and Cubic SVM on the training set of NSL-KDD dataset. After the full training of the model, the assessment of the model was evaluated on the 5870 samples of testing set. We got the 96.47%, 97.51%, and 97.80% accuracy score for linear SVM, quadratic SVM and Cubic SVM respectively. For the calculation of precision, recall and f1-score, we also calculate the complete classification report and plot the confusion metrices of the model. The classification report of all SVM variants is below (Table 3):

Table 3: Intrusion Attacks Classification – SVM Report

Classification Report of Linear SVM Classifier				
	Precision	Recall	F1-score	Support
1 - Normal	0.99	0.98	0.98	3501
2 - DoS	0.90	0.86	0.88	302
3 - Prob	1.00	0.08	0.15	12
4 - R2L	0.91	0.42	0.57	24
5 - U2R	0.93	0.97	0.95	2031
accuracy			0.96	5870
Avg (macro)	0.95	0.66	0.71	5870
Avg (weighted)	0.97	0.96	0.96	5870
Classification Report of Quadratic SVM Classifier				
	Precision	Recall	F1-score	Support
1 - Normal	0.99	0.98	0.99	3501
2 - DoS	0.97	0.92	0.94	302
3 - Prob	0.00	0.00	0.00	12
4 - R2L	0.91	0.42	0.57	24
5 - U2R	0.95	0.99	0.97	2031
accuracy			0.98	5870
Avg (macro)	0.76	0.66	0.69	5870
Avg (weighted)	0.97	0.98	0.97	5870
Classification Report of Cubic SVM Classifier				
	Precision	Recall	F1-score	Support
1 - Normal	0.99	0.98	0.99	3501
2 - DoS	0.97	0.94	0.95	302
3 - Prob	0.50	0.08	0.14	12
4 - R2L	0.90	0.38	0.53	24
5 - U2R	0.95	0.99	0.97	2031
accuracy			0.98	5870
Avg (macro)	0.86	0.67	0.72	5870
Avg (weighted)	0.98	0.98	0.98	5870

K Nearest Model

K Nearest Neighbor was also used for the classification of intrusion attacks using NSL-KDD dataset. In the proposed study, we trained the three variants of KNN including the KNN Fine, KNN Medium, and KNN Cubic on the training set of NSL-KDD dataset. After the full training of the model, the assessment of the model was evaluated on the 5870 samples of testing set. We got the 98.89%, 98.01%, and 98.11% accuracy score for KNN Fine, KNN Medium, and KNN Cubic respectively. For the calculation of precision, recall and f1-score, we also calculate the complete classification report and plot the confusion metrices of the model. The classification report of all KNN variants is below (Table 4):

Table 4: Intrusion Attacks Classification – KNN Report

Classification Report of Fine KNN Classifier				
	Precision	Recall	F1-score	support
1 - Normal	1.00	1.00	1.00	3501
2 - DoS	0.96	0.96	0.96	302
3 - Prob	0.73	0.67	0.70	12
4 - R2L	0.69	0.75	0.72	24
5 - U2R	0.99	0.98	0.99	2031
accuracy			0.99	5870
Avg (macro)	0.87	0.87	0.87	5870
Avg (weighted)	0.99	0.99	0.99	5870
Classification Report of Medium KNN Classifier				
	Precision	Recall	F1-score	Support
1 - Normal	0.99	1.00	0.99	3501
2 - DoS	0.91	0.94	0.92	302
3 - Prob	0.20	0.08	0.12	12
4 - R2L	0.91	0.42	0.57	24
5 - U2R	0.98	0.97	0.97	2031
accuracy			0.98	5870
Avg (macro)	0.80	0.68	0.72	5870
Avg (weighted)	0.98	0.98	0.98	5870
Classification Report of Cubic KNN Classifier				
	Precision	Recall	F1-score	Support
1 - Normal	0.99	1.00	0.99	3501
2 - DoS	0.91	0.94	0.93	302
3 - Prob	0.33	0.08	0.13	12
4 - R2L	0.91	0.42	0.57	24
5 - U2R	0.98	0.97	0.97	2031
accuracy			0.98	5870
Avg (macro)	0.82	0.68	0.72	5870
Avg (weighted)	0.98	0.98	0.98	5870

Decision Tree

For the classification of the intrusion attacks, the decision tree model was used on NSL-KDD dataset. The variant of decision tree including the Fine DT and Medium DT was trained on the training samples of the dataset. Following the training process of DT models, 5870 samples of test set was used for the evaluation of the model. All the selected evaluation measures were calculated on the test samples. We got the 98.82% and 98.16% accuracy score for Fine DT and Medium DT respectively. Rest of the evaluation measures were also calculated on the test set. The complete classification report of the variants of DT trained models is shown in below Table 5.

Table 5:Intrusion Attacks Classification – DT Report

Classification Report of Fine DT Classifier				
	Precision	Recall	F1-score	Support
1 - Normal	0.99	1.00	1.00	3501
2 - DoS	0.95	0.98	0.96	302
3 - Prob	0.86	0.50	0.63	12
4 - R2L	0.88	0.58	0.70	24
5 - U2R	0.99	0.98	0.98	2031
accuracy			0.99	5870
Avg (macro)	0.93	0.81	0.85	5870
Avg (weighted)	0.99	0.99	0.99	5870
Classification Report of Medium DT Classifier				
	Precision	Recall	F1-score	Support
1 - Normal	0.99	1.00	0.99	3501
2 - DoS	0.93	0.93	0.93	302
3 - Prob	0.00	0.00	0.00	12
4 - R2L	0.83	0.42	0.56	24
5 - U2R	0.98	0.98	0.98	2031
accuracy			0.98	5870
Avg (macro)	0.75	0.66	0.69	5870
Avg (weighted)	0.97	0.98	0.97	5870

Proposed Solution

As the accuracy of the ML models is high but the precision score of mostly models in not significant. For the significant score of accuracy, precision, recall and f1-score, we proposed an ensemble learning based voting classifier. Voting classifier mainly based on several standalone machine learning models that learn individually and makes its own prediction. Voting classifier take the prediction result from each model and announce the final perdition based on the majority votes. The proposed voting classifier is based on three different classifiers including the Fine Decision Tree, Fine KNN and Medium KNN. The class that gains majority votes will be predicted as the final class by the voting classifier. As the voting classifier make prediction on the experience of multiple classier, hence the predicting result by the voting classifier will be more robust and accurate.

The voting classifier was trained on the training samples of the NSL-KDD dataset for intrusion attacks classification. By following the training of the model, the proposed model was evaluated using the test samples of the dataset. All the chosen evaluation measures were calculated on test samples and got the 98.96% accuracy. For the calculation of remaining evaluation measures, complete classification report was also reported with test data in Table 6.

Table 6: Intrusion Attacks Classification – Voting Classifier Report

Classification Report of Linear Voting Classifier				
	Precision	Recall	F1-score	Support
1 - Normal	1.00	1.00	1.00	3501
2 - DoS	0.95	0.97	0.96	302
3 - Prob	0.86	0.50	0.63	12
4 - R2L	0.93	0.54	0.68	24
5 - U2R	0.99	0.99	0.99	2031
accuracy			0.99	5870
Avg (macro)	0.94	0.80	0.85	5870
Avg (weighted)	0.99	0.99	0.99	5870

Comparative Study

Lastly comparative study of all the trained models was performed for the fair comparison of the models. All the evaluation score of the models on the test set was compare using tabular and bar chat data. In the below table, the proposed voting classifier for attack prediction showed the highest (98.96%) relative to the other machine learning models. The complete comparison of all the models is shown in Table 7.

Table 7: Intrusion Attacks Classification – Comparative Results

Metrics	**Linear SVM**	**Quad SVM**	**Cubic SVM**	**Fine KNN**	**Medium KNN**	**Cubic KNN**	**Fine DT**	**Medium DT**	**Proposed**
Accuracy	0.9647	0.9751	0.9780	0.9889	0.9800	0.9810	0.9882	0.9816	0.9896
Precision	0.9467	0.7628	0.8633	0.8723	0.7969	0.8240	0.9316	0.7460	0.9426
Recall	0.6617	0.6613	0.6731	0.8716	0.6808	0.6818	0.8086	0.6639	0.7998
F1-score	0.7082	0.6936	0.7167	0.8715	0.7157	0.7196	0.8548	0.6911	0.8516

By using the comparison data of all the trained models, bar chat was plotted for the graphical representation of the evaluation scores.

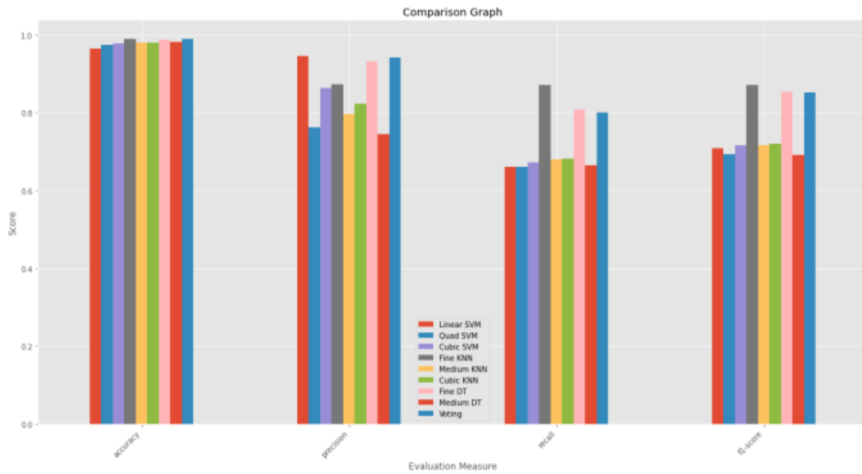

Figure 1: Evaluation Scores of trained models

Moreover, the accuracy score of all the trained models was also compare with the published study. We found the slightly different result due to different number of samples for different classes. Although we get the same number of samples for normal, DoS, R2L and Prob class, but we only get the 89 samples of U2R class after processing of dataset. While the paper used the 3086 samples of U2R class. The mismatching of the samples for U2R attack may be happen due to the different mapping criteria that we used by following the Field-Name file. We also assume that the slightly difference of evaluation score is also due to the different number of samples against the attack types. However, by following the given mapping criteria and by using the available samples of different attacks, we got the highest accuracy with our proposed model.

Table 8: Comparison of our and paper results.

Model Name	Our Score	Published Score
SVM Linear	0.9647	0.9847
SVM Quadratic	0.9751	0.9932
SVM Cubic	0.9780	0.9946
KNN Fine	0.9889	0.9964
KNN Medium	0.9800	0.9915
KNN Cubic	0.9810	0.9909
TREE Fine	0.9882	0.9992
TREE Medium	0.9816	0.9992
Proposed Model	0.9896	####

Intrusion Detection

ORIGINALITY REPORT

10%	2%	6%	6%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to University of Hertfordshire Student Paper	4%
2	Submitted to University of East London Student Paper	2%
3	"Intelligent Data Engineering and Automated Learning – IDEAL 2018", Springer Science and Business Media LLC, 2018 Publication	1%
4	Rajasekhar Chaganti, Vinayakumar Ravi, Tuan D. Pham. "Image-based malware representation approach with EfficientNet convolutional neural networks for effective malware classification", Journal of Information Security and Applications, 2022 Publication	1%
5	www.mdpi.com Internet Source	<1%
6	Suvankar Banerjee, Sk. Abdul Amin, Nilanjan Adhikari, Tarun Jha. "Essential elements regulating HDAC8 inhibition: a classification	<1%

based structural analysis and enzyme-inhibitor interaction study of hydroxamate based HDAC8 inhibitors", Journal of Biomolecular Structure and Dynamics, 2019
Publication

7	Sowndarya Krishnamoorthy, Luis Rueda, Sherif Saad, Haytham Elmiligi. "Identification of User Behavioral Biometrics for Authentication Using Keystroke Dynamics and Machine Learning", Proceedings of the 2018 2nd International Conference on Biometric Engineering and Applications - ICBEA '18, 2018 Publication	<1 %
8	gssrr.org Internet Source	<1 %
9	Vinayakumar Ravi, Harini Narasimhan, Tuan D. Pham. "Chapter 9 EfficientNet-Based Convolutional Neural Networks for Tuberculosis Classification", Springer Science and Business Media LLC, 2021 Publication	<1 %
10	"Service-Oriented Computing", Springer Science and Business Media LLC, 2018 Publication	<1 %
11	Md. Zainal Abedin, Kazy Noor-e-Alam Siddiquee, M. S. Bhuyan, Razuan Karim, Mohammad Shahadat Hossein, Karl Andersson. "Performance Analysis of	<1 %

Anomaly Based Network Intrusion Detection Systems", 2018 IEEE 43rd Conference on Local Computer Networks Workshops (LCN Workshops), 2018
Publication

Intrusion Detection

GRADEMARK REPORT

FINAL GRADE

/0

GENERAL COMMENTS

Instructor

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7