Intrusion Detection System

Batuhan Ozgur Basal, MSc Intelligent Systems & Robotics, De Montfort University, Leicester

This report describes the development of the neural network model that can identify network attacks using KDD Cup 1999 Dataset [1]. It covers the entire design and implementation from dataset pre-processing to model setup and finally testing the model. Python programming language and google colab (web IDE for python) are used in this study due to their versatility and ease of use in the field of machine learning libraries.

The main objective in this study is divided into three parts. First, the raw data must be pre-processed and be prepared for the model. We can't fit and apply machine learning methods on original dataset. We must convert the data to address the needs of each method to acquire the excellent efficiency on this predictive modelling assignment.

Next, the model must be built, and It needs to be ready for testing. Finally, during the testing phase, the parameters should be tested by trial and error method and the most suitable values should be selected.

Goals and Objectives

First, raw dataset uploaded to google colab to start pre-processing. To avoid confusion, each column is named according to the information obtained from the official website [1] from which the data was obtained. The KDD Cup 1999 dataset consists of 494021 rows and 42 columns in total. The last column consists of 23 network intrusion classifications. These are divided into 5 among themselves. Representative of: Denial-of-Service (DoS), Probe, Remote-to-Local (R2L), User-to-Root (U2R) and Normal. This classification has been reduced from 23 to 5 to make the dataset less complex. After examining the values of this dataset, 348435 duplicate records were determined in data set. Those records needed to be eliminated from the data set, since duplicate data consumes unneeded memory space and substantially delays computations. Then, 2 more columns that were detected as unnecessary were deleted from the data set.

In the next stage of data analysis, the categorical values were determined in the data set. We know that Machine Learning algorithms are based on numerical data. They use integers or floats as input source to predict an outcome. Therefore, categorical values must be converted to either integers or floats. For this, label encoding, and mapping methods are used. Then, data normalization was done by using z-score and MinMaxScaler function. Finally, the data was made ready for training by applying PCA (Principal Component Analysis).

All in all, network attack prediction was completed using 4 different classifier algorithms (Gaussian Naïve Bayes, Decision Tree, Artificial Neural Network, Support Vector Machine). The results obtained are given below.

Results

- A decrease was detected in the total number of rows and columns obtained as a result of the data pre-processing (145585 rows × 20 columns).
- As a result of the tests made on the ANN model, the parameters were optimized.
- The Decision Tree classifier algorithm had the best accuracy success among other algorithms including ANN.
- During the tests for ANN, it was seen that different activation and loss functions were effective in binary and multi-class classification (Sigmoid for binary and softmax for multi-class),(binary cross entropy for binary and categorical cross entropy for multi-class)
- Batch size had an opposite effect on the training speed.
- The PCA method had a great impact on the model success rate by reducing the columns (from 40 to 20) on the dataset.
- Adam optimizer had been successful and was the best optimizer in both binary and multi-class classifications.
- While the Gaussian Naive Bayes algorithm was successful in binary classification (0.9621), it was unsuccessful (0.4621) in multi-class classification.
- The success of all models used was more successful in binary classification than multi-class classification.

References

[1] KDD Cup 1999 [Online] : http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

[2] Pandas [Online] : https://pandas.pydata.org/

[3] Keras [Online] : https://keras.io/

[4] TensorFlow: [Online]. Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[5] Scikit-learn: Machine Learning in Python, Pedregosa *et al.*, JMLR 12, pp. 2825-2830, 2011.: https://scikit-learn.org/stable/about.html

[6] Van Rossum, G. & Drake Jr, F.L., 1995. *Python reference manual*, Centrum voor Wiskunde en Informatica Amsterdam.

[7]Google Colab [Online] : https://colab.research.google.com/?utm_source=scs-index

APPENDIX

Hyperparameter Tuning For ANN Model (Binary Classification)

(All values are optimized for ANN model binary classification)

Test Size	Training Loss	Training Accuracy	Recall Score	F1 Score	Precision Score
0.60	0.0232	0.9924	0.9901	0.9923	0.9946
0.50	0.0176	0.9947	0.9905	0.9939	0.9973
0.40	0.0170	0.9953	0.9934	0.9945	0.9955
0.30	**0.0151**	**0.9958**	**0.9928**	**0.9954**	**0.9979**
0.20	0.0162	0.9952	0.9892	0.9933	0.9974
0.10	**0.0172**	**0.9947**	**0.9893**	**0.9927**	**0.9961**

The number Columns in the data frame after PCA

Number of Dimensions	Training Loss	Training Accuracy	Recall Score	F1 Score	Precision Score
39	0.0158	0.9955	0.9941	0.99562	0.9970
30	0.0205	0.9946	0.9919	0.9934	0.9949
20	**0.0151**	**0.9958**	**0.9928**	**0.9954**	**0.9979**
10	0.0336	0.9880	0.9769	0.9851	0.9935
5	**0.0570**	**0.9825**	**0.9604**	**0.9779**	**0.9961**

Optimizers	Training Loss	Training Accuracy	Recall Score	F1 Score	Precision Score
ADAM	**0.0151**	**0.9958**	**0.9928**	**0.9954**	**0.9979**
SGD	0.0968	0.9739	0.9453	0.9658	0.9872
RMSProp	0.0255	0.9932	0.9877	0.9919	0.9961
AdaDelta	**0.1712**	**0.9534**	**0.9032**	**0.9376**	**0.9747**
AdaGram	0.0767	0.9751	0.9508	0.9687	0.9874

Learning Rate	Training Loss	Training Accuracy	Recall Score	F1 Score	Precision Score
0.1	**0.6737**	**0.6013**	**0.0**	**0.0**	**0.0**
0.01	0.0261	0.9902	0.9912	0.9926	0.9941
0.001	**0.0151**	**0.9958**	**0.9928**	**0.9954**	**0.9979**
0.0001	0.0339	0.9912	0.9802	0.9862	0.9923

Batch Size	Training Loss	Training Accuracy	Recall Score	F1 Score	Precision Score
8	0.0201	0.9944	0.9898	0.9935	0.9971
16	0.0177	0.9948	0.9936	0.9946	0.9956
32	0.0164	0.9949	0.9925	0.9946	0.9968
64	**0.0151**	**0.9958**	**0.9928**	**0.9954**	**0.9979**
128	0.1893	0.9869	0.9640	0.9701	0.9763
256	0.0219	0.9937	0.9916	0.9931	0.9945
512	**0.1027**	**0.9882**	**0.9864**	**0.9816**	**0.9767**

Loss Function	Training Loss	Training Accuracy	Recall Score	F1 Score	Precision Score
Binary cross-entropy	**0.0151**	**0.9958**	**0.9928**	**0.9954**	**0.9979**
Categorical cross-entropy	0.0	0.6043	0.0	0.0	0.0
kl_divergence	**9.7014e-07**	**0.3955**	**1.0**	**0.5709**	**0.3995**

Output Layer Activation Function	Training Loss	Training Accuracy	Recall Score	F1 Score	Precision Score
softmax	**0.0194**	**0.3969**	**N/A**	**0.5674**	**0.3961**
sigmoid	**0.0151**	**0.9958**	**0.9928**	**0.9954**	**0.9979**
tanh	0.2490	0.9775	0.9518	0.9593	0.9669
relu	**9.2040**	**0.3964**	**N/A**	**0.5687**	**0.3973**
softplus	0.0705	0.9849	0.9697	0.9826	0.9959

The Number of Neurons in the Input Layer	Training Loss	Training Accuracy	Recall Score	F1 Score	Precision Score
10	0.0171	0.9948	0.9929	0.9952	0.9974
20	0.0183	0.9949	0.9923	0.9945	0.9967
32	**0.0151**	**0.9958**	**0.9928**	**0.9954**	**0.9979**
40	0.0163	0.9950	0.9931	0.9935	0.9938

Binary Classification Algorithms	Accuracy Score	Multiclass Classification Algorithms	Accuracy Score
Decision Tree	**0.9974**	Decision Tree	0.9973
Gaussian Naïve Bayes	0.9621	Naïve Bayes	**0.4621**
Support Vector Machine	0.9519	Support Vector Machine	0.9483
ANN	0.9958	ANN	0.9948

Gaussian Naïve Bayes Binary Classification Accuracy

	precision	recall	f1-score	support
Attack	0.96	0.98	0.97	26460
Normal	0.96	0.94	0.95	17216

Gaussian *Naïve Bayes Multi-class Classification Accuracy*

	precision	recall	f1-score	support
normal	0.94	0.14	0.25	26380
dos	0.44	0.96	0.60	16315
probe	0.21	0.88	0.33	659
r2l	0.53	0.41	0.46	303
u2r	0.02	0.79	0.05	19

Decision Tree Binary Classification Accuracy

	precision	recall	f1-score	support
Normal	1.00	1.00	1.00	26341
Attack	1.00	1.00	1.00	17335

Decision Tree Multi-class Classification Accuracy

	precision	recall	f1-score	support
normal	1.00	1.00	1.00	26427
dos	1.00	1.00	1.00	16340
probe	0.97	0.97	0.97	616
r2l	0.94	0.94	0.94	279
u2r	0.42	0.57	0.48	14

Support Vector Binary Classification Accuracy

	precision	recall	f1-score	support
Normal	0.93	0.99	0.96	26373
Attack	0.98	0.89	0.94	17303

Support Vector Multi-class Classification Accuracy

	precision	recall	f1-score	support
normal	0.93	0.99	0.96	26373
dos	0.98	0.92	0.95	16360
probe	0.67	0.30	0.41	640
r2l	0.00	0.00	0.00	289
u2r	0.00	0.00	0.00	14