A NOVEL APPROACH FOR NETWORK INTRUSION DETECTION USING EXTREME GRADIENT BOOSTING AND LONG SHORT-TERM MEMORY RECURRENT NEURAL NETWORKS

R. E. RATNAYAKE

Department of Statistics and Computer Science, Faculty of Science,
University of Peradeniya.

1. Introduction

Recently, Extreme Gradient Boosting (XGBoost), which is an optimized distributed gradient boosting library, has been receiving increasing attention in many domains since the algorithm is highly effective in reducing the computing time and provides ideal use of memory assets. It can take care of missing values, supports parallel structure in tree construction, and has the unique ability to perform boosting on added data already on the trained model. It has also been used in the field of network intrusion detection and achieved very high accuracies. The fact that the importance scores for each attribute can be retrieved straight away after the boosted trees are constructed is another advantage of using ensembles of decision tree methods like gradient boosting.

Considering deep learning-based approaches, Recurrent Neural Network (RNN) models have also given high accuracy values in network intrusion detection. Several studies have justified the modelling of network traffic as a time series to improve network intrusion detection.

Bringing the effects of both of the above models together, this paper applies XGBoost on NSL-KDD dataset for calculating feature importance scores to create 4 minimal feature sets and then studies the effects of using them in an LSTM RNN classifier model in order to examine the feasibility of applying XGBoost for feature selection in network intrusion detection performed using LSTM models.

2. Materials and Methods

Data collection and pre-processing

This study uses the NSL-KDD dataset which is a benchmark dataset used for network intrusion detection. It consists of a separate train set (KDDTrain+) and a test set (KDDTest+) with 125973 and 22543 records respectively. The records contain 38 numeric features and 3 nonnumeric features. First, the categorical data are numericalized by encoding them as binary vectors. Secondly, the data are normalized to bring the values to a common scale. Each record is sampled individually to unit norm.

Model implementation

The main goal of this study is to test the feasibility of using XGBoost as a feature selection technique for network intrusion detection and to compare the performances with LSTM RNN. XGBoost is applied on the dataset at different learning rates (0.001, 0.01, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8 and 0.9) and feature importance scores are calculated in each case. 4 minimal feature sets are created after careful observation on the importance scores. Then, a LSTM RNN classifier model with 80 hidden nodes, Sigmoid as the activation function, learning rate set to 0.5 and number of epochs set to 45 is used. Inputs to the network are the data columns corresponding to network data and the output represents the predicted value: either 0 or 1 denoting whether it is a normal record or a record of an attack. The LSTM RNN model is experimented on with the full

feature set and with the minimal feature sets created using the feature importance scores calculated by XGBoost.
The implementations are done using Jupyter Notebook version 5.7.4.

3. Results

Results used from the XGBoost implementation

Table 1: Results of XGBoost implementation

Learning rate	Accuracy (%)
0.001	79.70
0.01	77.52
0.1	79.50
0.2	79.77
0.3	79.33
0.4	79.85
0.5	79.96
0.6	79.03
0.7	79.83
0.8	81.40
0.9	78.64

Table 1 shows that XGBoost gives the highest accuracy at learning rate 0.8 and the second highest at 0.5. Therefore, feature importance values calculated in those cases are mainly used for minimal feature set creation.

Table 2: Feature importance scores at learning rate 0.8

	columns	importances
56	ecr_i	0.283210
65	http	0.219024
1	src_bytes	0.158658
6	hot	0.052231
2	dst_bytes	0.047035
30	dst_host_same_srv_rate	0.036526
90	private	0.020444
19	count	0.020093
32	dst_host_same_src_port_rate	0.016791

Table 3: Feature importance scores at learning rate 0.5

	columns	importances
56	ecr_i	0.376655
1	src_bytes	0.144783
65	http	0.140390
2	dst_bytes	0.063815
19	count	0.030076
90	private	0.026545
29	dst_host_srv_count	0.025657
30	dst_host_same_srv_rate	0.024534
32	dst_host_same_src_port_rate	0.023836

The minimal feature sets created mainly upon careful observation of feature important scores at learning rates 0.8 (Table 2) and 0.5 (Table 3) are:

- 8-feature set: 1, 2, 6, 19, 30, 56, 65, 90
  Feature reduction ratio: 122:8
- 6-feature set: 1, 2, 6, 19, 56, 65
  Feature reduction ratio: 122:6
- 4-feature set: 1, 2, 56, 65
  Feature reduction ratio: 122:4
- 3-feature set: 1, 56, 65
  Feature reduction ratio: 122:3

Results of LSTM RNN using all the 122 features in the pre-processed dataset

Results show that the LSTM RNN classifier model, without feature selection, achieves an accuracy of 78.71% and a F1 score of 0.7968

Results of LSTM RNN using the minimal feature sets created

Results, as seen in Table 4, show that all the models created using the minimal feature sets have achieved accuracy values much close to the accuracy value achieved by the model created using all the 122 features. The minimal feature set created using 4 features has shown the closest accuracy (76.84%) and the F1 score (0.7745) to those obtained without feature selection. Further, it can be seen that according to the experimental results, the values for time taken to perform one classification by the different models show little difference when compared to each other.

Table 4: Results of LSTM RNN classifier using different feature sets

Feature set	Feature reduction ratio	Accuracy	F1 Score	Time taken for one classification (μs)
All 122 features	122:122	78.71	0.7968	20
8-features	122:8	73.82	0.7606	19
6-features	122:6	75.60	0.7715	21
4-features	122:4	76.84	0.7745	22
3-features	122:3	74.11	0.7663	25

4. Discussion

As seen in Table 4, compared to the accuracy and F1 score obtained by LSTM network trained with all features, it is notable that the values obtained by the models trained using minimal feature sets almost match with them and are acceptable.

The experimental results show only little difference in the time values for one classification when compared with each other. The reason for this, perhaps, could be the other processes running on the machine in parallel. However, more testing and optimization might give lower testing times for the models trained using minimal feature sets as expected.

As advantages of using the proposed method, it can be stated that reduction in number of features reduces the amount of processing involved and clearly saves up memory space required given the very high reduction ratio of the number of features.

5. Conclusions

Extreme Gradient Boosting (XGBoost) has shown to be very effective for feature selection in network intrusion detection. The classification accuracies achieved by using the minimal feature sets created using XGBoost are very close to that achieved using the original feature set. Using XGBoost for feature selection in designing an intrusion detection system using long short-term memory recurrent neural networks will reduce the complexity of the system whilst achieving an acceptable accuracy and a better performance in terms of memory consumption as very high feature reduction ratios can be achieved. This method can also be used in designing a first layer of defense for alerting users about a possible attack, until the complex higher layers complete more accurate classification using more features.