Subject : Distributed Data Management(CS750)

Title : Performance Enhancement of Anomaly based Intrusion Detection System using Deep Learning Techniques in the Big Data Environment.

Name-Soumyajit Bhattacharyya
Roll No.-192CS025
Regn. No.-192293
Course-M.Tech(2nd Semester)
Phone-8981472520
Email - soumyabhat.rkmv@gmail.com

Abstract : Hardwares or softwares that are used for monitoring and analyzing data flow between the hosts in a network to detect security threats are known as Intrusion Detection System(IDS) . Nowadays, because of the advanced communication system and increased number of networked devices, the network traffic data is considered as big data due to its volume . To predict and prevent the intrusion attacks, deep learning techniques are applied here on big data to come up with a model that can differentiate between normal and attacked network traffic flow and applied on the recently released CSE-CIC-IDS2018 dataset.

Keywords : Big Data , Intrusion Detection System , Deep Neural Network.

1. Introduction

The expanding nature of the use of the Internet and the rapidly growing nature of the volume of data originated from different sources are giving more opportunity to the hackers to initiate their harmful attacks and to make use of intelligent methods and gadgets to intrude in the network. On the contrary, researchers and developers aim to raise the efficiency of early harmful attack forecasting and detection. Intrusion refers to security violations that can threaten a system and Intrusion detection systems (IDSs) are the hardware or software that are used for monitoring and analyzing data flow between the hosts in a network to prevent intrusion.. Intrusion detection systems in general use two known methods to analyze flow and detect attacks: i.Signature based detection ii.Anomaly detection. Signature-based detection is used to detect previously known attacks whether anomaly detection is used to detect unseen attacks or unknown patterns related to abnormal behavior in network flow. Its work is based on building machine learning,deep learning and artificial neural networks where deviated patterns are detected and considered as abnormal or unusual behavior.

With the expansion of connectivity and networked systems , the network traffic flow is currently considered as big data. Big data has 7 ‘V’ properties , that are Volume, Variety, Velocity, Variability, Veracity, Visualization and Value.Volume is about how large is the data in size. Velocity is the speed of accessing data.Variety is about heterogeneous sources and unstructured nature of the data. Variability says that the same data can have different meanings. Veracity is about the quality of data. Visualization consists of graphics, charts and other plots that help to understand the meaning of data and retrieve more details. And finally Value is about how data processing can be done to produce a meaningful outcome.In this project mainly the Volume property of big data is utilised.The size of the data is nearly 7GB containing around 17 million records with more than 75 features. Also the Variety property came into help to some extent. Though, in most of the systems firewalls are used to avert the cyber attacks, Intrusion detection

systems (IDSs) play a notable role to enhance the system security. The intruders' works are to find new ways to bypass the system's security procedure.

In the current project I have done binary classification between normal and attacked type of packets and multi-class classification among attacked type of packets using deep learning technique on the recently released CSE-CIC-IDS2018 dataset because of the huge volume of data and compared with Random forest method.

2. Literature Review

Many of the researchers previously explored this domain using different approaches.

Osama Faker and Erdogan Dogdu[1] integrated Apache Spark and its machine learning(ML) and deep learning libraries for developing a deep learning model i.e. neural network. Then it is compared with Random forest classifier and Gradient Boosting Tree classifier(GBT). Comparison of performances is done on CICIDS-2017 and UNSW NB15 datasets for both binary and multiclass classification. The results show that on the UNSW NB15 dataset ,the highest accuracy of 99.16% and 97.01% is achieved by DNN for binary and multiclass respectively. On the other hand, for CICIDS-2017 dataset GBT classifier achieved the maximum accuracy(99.9%) and DNN achieved highest accuracy with 99.56% for binary and multiclass respectively.

X. Zhang and J. Chen[2] used Restricted Boltzmann Machine(RBM). Two hybrid algorithms are created by combining it with Support Vector Machine(SVM) and Deep Belief Network(DBN). They have compared the performances of two algorithms along with a traditional algorithm where Principal Component Analysis is combined with Back Propagation on KDDCup-99 dataset.RBM-DBN achieved highest accuracy with 97.16%.

Gozde Karatas et al.[3] has given a brief overview about the variants of IDS. But their major contribution is comparison of different available network intrusion dataset in terms of different attack types, number of records, features of the network packet etc. in detail. Also differences between machine learning and deep learning to use those datasets is provided.

G. C. Fernández et al.[4] proposed a robust DNN which is applied on ISCX IDS 2012 and CICIDS 2017 dataset and compared with other traditional methods like Decision Tree,Random Forest, MLP , Naive Bayes etc.IP addresses are here used as features. Considering and without considering IP address 99.93% and 96.77% accuracy was achieved for CICIDS 2017 dataset respectively.

Rahul Vigneswaran K et al.[5] compared DNN with 1,2,3,4 and 5 layers along with traditional ML techniques like Naive Bayes, Random Forest, Logical regression,SVM etc. on KDDCup-99 dataset and found that DNN with 3 layers showed maximum accuracy of 93%.

The summary of the literature review is summarized below in Table 1.

Paper	Dataset	Method	Accuracy
Osama Faker and Erdogan Dogdu [1]	CICIDS-2017 UNSW NB15	GBT(Binary) , DNN-4(Multi) DNN-4(Both)	99.9%,99.56% 99.16%,97.01%
X. Zhang and J. Chen [2]	KDDCup-99	RBM-DBN	97.16%
G. C. Fernández et al. [4]	ISCXIDS-2012 CICIDS-2017	DNN(with and without IP)	96.77%
Rahul Vigneswaran K et al. [5]	KDDCup-99	DNN-3	93%

Table 1 - Summary of literature review

3. Main Work

The explanatory details of the work is arranged in subsections below.

3.1. Proposed Work

The method proposed in the work is to classify between normal and attacked traffic flow obtained from the CSE-CIC-IDS2018 dataset and also to classify among different attack types . IDS has huge impact in the domain of cyber security.Previously due to lower number of attack types and without the availability of benchmark datasets machine learning techniques were used.But with the rapidly increasing size of network data ,deep learning techniques are inevitable to increase efficiency,accuracy and response time of IDS. In this work a deep neural network technique is used with 4 hidden layers along with one input and one output layer .With the large number of records along with a large number of features the deep learning technique must be more useful than traditional machine learning techniques.In order to show that it is compared with Random Forest w.r.t performance evaluation metrics like Accuracy,F1-score , Recall , Precision . The previous works on this domain consist of either machine learning techniques that are not scalable to big data or relatively old dataset like KDDCup 99, NSL-KDD, UNSW-NB15 or CICIDS2017. So the recently released CSE-CIC-IDS2018 dataset is explored with a deep learning technique that is scalable to big data.

3.2. Concepts used

i)Big Data - Here the 'Volume' property of big data is used. To simulate the real world scenario of network traffic, the size of the dataset is very large (nearly 7GB) with nearly 17 million records with more than 75 attributes. Without the help of deep learning this amount of large data cannot be handled properly. The 'Variety' property also came to help. Because the data is produced from heterogeneous sources and also a total of 10 days of traffic data is combined to train and evaluate the model.The Deep Neural Network technique used(which is discussed later) has obtained 98.83% accuracy in case of binary classification and 96.63% in case of multiclass classification.The deep learning technique generated a large impact on large sized dataset by obtaining a commendable 96%+ accuracy both way.

ii) Deep Learning - Deep learning is a specialized machine learning technique.It is basically a neural network. Deep learning algorithms operate using a number of levels/layers. The layers are interconnected and the output of the previous layer is fed as input to the next level,so it is called feed-forward network.Each neuron uses an activation function to systematize its output.The activation functions can be linear or nonlinear in nature. The number of hidden layers indicates how deep a deep learning model is. Epochs are continued with some constraint to get better accuracy.In every epoch in each step,the parameters are updated to achieve better accuracy and smaller loss. The typical activation functions are like ReLU,Sigmoid and Softmax. Image Processing, Natural Language Processing are two of the most popular applications of deep learning. In deep learning, a large amount of data in comparison with machine learning is needed to build the model which is ideal for big data analytics.Long time is taken in deep learning algorithms to train.But that large amount of complex data is not properly handled by machine learning and it takes more time.

iii)Backpropagation - Backpropagation is a method to train neural networks.Based on the loss in the previous epoch the weights of the connections and biases of the neurons are modified based on some rule. The algorithm works by calculating the gradient of the loss function w.r.t every weight using chain rule one layer at a time.The algorithm propagates backward from the last layer to avoid unnecessary calculations and is an example of dynamic programming.

iv)Random forest - Random forest classifier is an ensemble classifier made up of multiple decision trees. Each decision tree is trained on different data samples and the final class is decided by majority election/voting.

3.3. Model Used

The deep neural network used here is made up of 4 hidden layers between 1 input and 1 output layer which are fully connected with each other.ReLU function is used in hidden layers.Sigmoid and Softmax functions are used in the output layer in case of binary and multi class classification respectively.Input layer contains 77 neurons. For binary classification the output layer contains 1 neuron. For multiclass classification the output layer contains 13neurons i.e the number of output labels. The hidden layers contain 256,128,64 and 32 neurons respectively from first to fourth.The neuron count in each hidden layer decreases by half to ensure accurate output and reduce cost. To prevent overfitting and to make the model more robust regularization technique is used. Between every two fully connected layers dropout of 0.1 and between output layer and last hidden layer dropout of 0.5 is used for the purpose of speeding up.Dropout removes neurons with their connection in a random manner to prevent overfitting and to speed up and reduce cost. Binary cross entropy function and categorical cross entropy function is used in binary and multi class classification respectively.

The brief description of the model is given below in Table 2 and Table 3.For binary classification the Adam optimizer is used with default learning rate 0.001. Number of epochs is set as 20 for binary and 100 for multiclass because after that accuracy is more or less stable.The batch size is set as 1024. For multiclass classification the Adam optimizer is used with learning

rate 0.01. The batch size is set as 256.The random forest classifier is used with 100 iterations as default value.

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_1 (Dense)              (None, 256)               19968
_________________________________________________________________
dropout_1 (Dropout)          (None, 256)               0
_________________________________________________________________
dense_2 (Dense)              (None, 128)               32896
_________________________________________________________________
dropout_2 (Dropout)          (None, 128)               0
_________________________________________________________________
dense_3 (Dense)              (None, 64)                8256
_________________________________________________________________
dropout_3 (Dropout)          (None, 64)                0
_________________________________________________________________
dense_4 (Dense)              (None, 32)                2080
_________________________________________________________________
dropout_4 (Dropout)          (None, 32)                0
_________________________________________________________________
dense_5 (Dense)              (None, 1)                 33
=================================================================
Total params: 63,233
Trainable params: 63,233
Non-trainable params: 0
```

Table 2 - Model Summary for Binary classification

```
Model: "sequential_2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_6 (Dense)              (None, 256)               19968
_________________________________________________________________
dropout_5 (Dropout)          (None, 256)               0
_________________________________________________________________
dense_7 (Dense)              (None, 128)               32896
_________________________________________________________________
dropout_6 (Dropout)          (None, 128)               0
_________________________________________________________________
dense_8 (Dense)              (None, 64)                8256
_________________________________________________________________
dropout_7 (Dropout)          (None, 64)                0
_________________________________________________________________
dense_9 (Dense)              (None, 32)                2080
_________________________________________________________________
dropout_8 (Dropout)          (None, 32)                0
_________________________________________________________________
dense_10 (Dense)             (None, 13)                429
=================================================================
Total params: 63,629
Trainable params: 63,629
Non-trainable params: 0
```

Table 3 - Model Summary for Multiclass classification

3.4. Dataset Description

The dataset used here is CSE-CIC-IDS2018 dataset which is the result of a collaborative project between the Communications Security Establishment (CSE) & the Canadian Institute for Cybersecurity (CIC). The dataset has 16137183 records after removing the null and infinity values.There are total 13390249 normal packet flows and 2746934 attacked packet flows.1 is represented as normal and 0 as attacked packet in case of binary classification.There are total 13 attack types with frequencies 862382, 611, 230, 686012, 1730, 41508, 461912, 139890, 10990, 193354, 160639 , 87, 187589 respectively for multiclass classification.There are 77 features that are used for classification.Those are like Flow Duration, Total number of Forward and Backward packets, Minimum,maximum,mean length of packets,different flag information,protocol,segment size etc.

The other 6 features like source and destination ip address and port information,flow id and timestamp are dropped in the preprocessing stage. There is a label for every record like Benign (i.e normal flow) or the name of the attack.The dataset includes seven different kind of attack like Bruteforce attack, Infiltration attack, DoS attack, DDoS attack, Botnet attack, Web attack, and Heartbleed attack and their different variants. The dataset is hosted in AWS and can be downloaded from there using the command line interface.

3.5. Preprocessing of Data

To supply more relevant data to the deep neural network classifier, a series of preprocessing operations are done on the data. These operations are as follows :

• Original dataset includes the source IP address,destination IP address,source port number and destination port number, which must be removed to ensure unbiased classification. Because of that information the model may result in overfitting. The characteristics of the packet helps in the formation of the model , so that any flow with alike packet information is classified regardless of its ip and port information.Also timestamp information is removed as temporal information of the packet flow is redundant.

• The multi-class labels are basically the name of the attacks which are encoded numerically, and after that hot encoding is used to create a vector.This is only used in case of multiclass classification.

• The numerical data in the dataset in every attribute span heterogeneously.So normalization is done on the dataset in each attribute between 0 to 1. This helps in providing more analogous values to the classifier.

• Every tuple with null or infinity value is dropped.

• For multiclass classification, packets representing normal network flow are ignored as they make up the large part of the data and only the packets with attack information are kept for evaluation purposes.

3.6. Hardware and Software Specifications

The entire code is hosted using Google Colab with the help of Tensor Processing Unit(TPU).The TPU is used to achieve more speed within a short time and also to get around 35 GB RAM and more than 100 GB memory. The full code is written using Python language and executed using Google Colab platform.The dataset is saved in drive after certain preprocessing and also the required numpy arrays are saved to reduce computation overhead in future. The Keras and Tensorflow libraries are heavily used.

3.7. Experimental Details

At first the data is downloaded using AWS CLI described by the official website about the dataset(https://www.unb.ca/cic/datasets/ids-2018.html).10 .csv files are downloaded with total size around 7 GB.There is a file around 4 GB and all other files are less than 400 MB.At first, timestamp information is removed manually from all the .csv files.The large .csv file had extra 4 columns called flow id, src ip , dst ip and src port which were also removed manually.The files are then kept in a folder in google drive. But due to the large size of the dataset the whole data cannot be loaded into a numpy array at once.So 9 small files were kept in a folder and the large file in another folder. Then the 9 files were stacked in a numpy array and the big file was stacked in another numpy array. Separate preprocessing is then done on both the numpy array like label encoding,removing missing values,data normalization etc. After that the numpy arrays containing data and labels from both parts were stored in google drive.Again they are concatenated because otherwise all the available RAM had been used.Same thing is done for multiclass classification also.In multiclass classification we have transformed the labels into categorical attribute.After concatenation, finally the column named dst port is removed from the data.After that the data is split into train and test data into 80%and 20% respectively.The data is shuffled randomly.After the training using Deep Neural Network and Random Forest,the performance of the classifiers are compared w.r.t different evaluation metrics.The confusion matrix and classification report are also computed using scikit-learn library.

3.8. Results and Evaluation

The performance of the Deep Neural Network classifier and Random Forest are evaluated using several evaluation metrics like accuracy,f1-score,recall and precision .Also the confusion matrix and classification are shown.The time taken for binary classification using DNN takes around 40 minutes whether for Random Forest is around 140 minutes. The multiclass classification takes around 20 minute for DNN and more or less the same for Random Forest.A confusion matrix is a representation of predicted results for a classification problem by providing the prediction of the samples separated as shown in Table-4. Suppose class 0 is positive and class 1 is negative.

	Class 0(Predicted)	Class 1(Predicted)
Class 0(Actual)	TP	FN
Class 1(Actual)	FP	TN

Table 4 : Confusion matrix

True Positive (TP) : Actually the sample is positive, and prediction is also positive.

False Negative (FN) : Actually the sample is positive, but prediction is negative.

True Negative (TN) : Actually the sample is negative, and prediction is also negative.

False Positive (FP) : Actually the sample is negative, but prediction is positive.

Then in terms of TP,FP,TN,FN the accuracy, f1-score ,recall and precision is defined as :

Accuracy = (TP + TN)/(TP + FP + TN + FN) implies the fraction of correctly classified samples.

Precision = (TP)/(TP + FP) implies the probability that a sample classified as positive is indeed positive.

Recall = (TP + TN)/(TP + FN) implies the probability that the class is correctly recognized.

F1-Score = 2*Recall*Precision/(Recall + Precision) which uses harmonic mean instead of arithmetic mean.Here we are using the weighted average for precision,recall and f1-measure for every class.

	Accuracy	Precision	Recall	F1-score
DNN	98.84%	98.85%	98.84%	98.82%
Random Forest	98.67%	98.67%	98.67%	98.66%

Table 5 : Binary Classification performance comparison

	Accuracy	Precision	Recall	F1-score
DNN	96.63%	96.71%	96.63%	96.47%
Random Forest	96.66%	96.71%	96.66%	96.52%

Table 6 : Multiclass Classification performance comparison

The results and comparisons of DNN and Random Forest in terms of different evaluation metrics for binary and multiclass classification are given in table 5 and table 6.We are able to see that as the binary classification is operated on large data, DNN performs slightly better over Random Forest. Similarly in case Multiclass classification is operated on relatively small

data,both classifiers' performances are comparable. The performance data for binary and multiclass classification with DNN and Random Forest is presented in Table 7,8,9 and 10 below.

```
The accuracy score is 0.9883638317339735
The precision score is 0.9884547367456414
The recall score is 0.9883638317339735
The f1-score is 0.9882185494582253

[[ 514446   35378]
 [   2177 2675436]]

=== Classification Report ===
              precision    recall  f1-score   support

           0       1.00      0.94      0.96    549824
           1       0.99      1.00      0.99   2677613

    accuracy                           0.99   3227437
   macro avg       0.99      0.97      0.98   3227437
weighted avg       0.99      0.99      0.99   3227437
```

Table 7 : Binary classification report using DNN

```
The accuracy score is 0.986718253524391
The precision score is 0.9867128304513948
The recall score is 0.986718253524391
The f1-score is 0.9865768177786151

[[ 514151   35673]
 [   7193 2670420]]

=== Classification Report ===
              precision    recall  f1-score   support

           0       0.99      0.94      0.96    549824
           1       0.99      1.00      0.99   2677613

    accuracy                           0.99   3227437
   macro avg       0.99      0.97      0.98   3227437
weighted avg       0.99      0.99      0.99   3227437
```

Table 8 : Binary classification report using Random Forest

```
The accuracy score is 0.9663370265404897
The precision score is 0.9671175309001078
The recall score is 0.9663370265404897
The f1-score is 0.9647334369645175


=== Classification Report ===
               precision    recall  f1-score   support

           0       1.00      1.00      1.00    172249
           1       0.94      0.60      0.74       134
           2       1.00      0.48      0.65        42
           3       1.00      1.00      1.00    137180
           4       1.00      1.00      1.00       333
           5       1.00      0.99      1.00      8398
           6       1.00      1.00      1.00     92422
           7       0.76      0.51      0.61     27852
           8       1.00      0.99      1.00      2206
           9       0.72      0.89      0.79     38966
          10       1.00      1.00      1.00     32029
          11       0.00      0.00      0.00        17
          12       1.00      1.00      1.00     37559

    accuracy                           0.97    549387
   macro avg       0.88      0.80      0.83    549387
weighted avg       0.97      0.97      0.96    549387
```

Table 9 : Multiclass classification report using DNN

```
The accuracy score is 0.9666064176982709
The precision score is 0.9670922150258797
The recall score is 0.9666064176982709
The f1-score is 0.9652494620412682


=== Classification Report ===
              precision    recall  f1-score   support

           0       1.00      1.00      1.00    172249
           1       1.00      0.66      0.80       134
           2       0.94      0.69      0.79        42
           3       1.00      1.00      1.00    137180
           4       1.00      1.00      1.00       333
           5       1.00      1.00      1.00      8398
           6       1.00      1.00      1.00     92422
           7       0.75      0.52      0.62     27852
           8       1.00      0.99      0.99      2206
           9       0.72      0.88      0.79     38966
          10       1.00      1.00      1.00     32029
          11       1.00      0.59      0.74        17
          12       1.00      1.00      1.00     37559

    accuracy                           0.97    549387
   macro avg       0.95      0.87      0.90    549387
weighted avg       0.97      0.97      0.97    549387
```

Table 10 : Multiclass classification report using Random Forest

3.9. Conclusion and Future Works

This project aims to provide a model to simulate an IDS to correctly classify the normal and attacked packet flows. Comparison between two different models are done using evaluation metrics and we can see the DNN performs better in case of binary classification whether with multiclass classification the random forest has done better. Big data is used here with deep learning to make the most of the model.But there are several limitations of the model which can be overcomed in future.Like due to availability of only standalone Apache Spark , the integration of Spark cannot be done with deep learning. But using cloud computing platforms like AWS , the Apache Spark can be integrated to get a better result for a very short response time.More number of spark clusters give short response time and accurate computation. Due to the disconnective nature of Google Colab we can not use cross validation split , which indeed gives better training. Also the number of epochs can be increased for fine tuning to get more accurate results. But each epoch takes a long time due to which the google colab gets disconnected,so the usage of spark clusters are highly recommended. Other hyper parameters can be fine tuned to get more accuracy. Also adding or removing some layers in case of multiclass classification can improve accuracy. Other recently released dataset can also be examined by keeping the skeleton of the model and fine-tuning it. Performances of other machine algorithms like Support Vector Machine(SVM) , Gradient Boosting Tree(GBT) etc. can be measured. In this experiment , the session is getting crashed with the use of SVM and GBT does not provide multiclass classification. So I have not used them. And finally all the features of the dataset are used for training the model. But by the means of better feature selection algorithms , the redundant features can be dropped and overhead can be reduced.

3.10. References

[1] Osama Faker and Erdogan Dogdu, “Intrusion Detection Using Big Data and Deep Learning Techniques,” In Proceedings of the 2019 ACM Southeast Conference (ACM SE ’19). Association for Computing Machinery, New York, NY, USA,2019, pp. 86–93.

[2] X. Zhang and J. Chen, "Deep learning based intelligent intrusion detection," 2017 IEEE 9th International Conference on Communication Software and Networks (ICCSN), Guangzhou, 2017, pp. 1133-1137.

[3] G. Karatas, O. Demir and O. Koray Sahingoz, "Deep Learning in Intrusion Detection Systems," 2018 International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism (IBIGDELFT), ANKARA, Turkey, 2018, pp. 113-116.

[4] G. C. Fernández and S. Xu, "A Case Study on using Deep Learning for Network Intrusion Detection," MILCOM 2019 - 2019 IEEE Military Communications Conference (MILCOM), Norfolk, VA, USA, 2019, pp. 1-6.

[5] R. K. Vigneswaran, R. Vinayakumar, K. P. Soman and P. Poornachandran, "Evaluating Shallow and Deep Neural Networks for Network Intrusion Detection Systems in Cyber Security," 2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Bangalore, 2018, pp. 1-6.