Introduction to Machine Learning
Project Topic: Intrusion Detection Using Machine Learning Algorithms.

Priyanka Ashok Sapkal

August 14, 2019

Problem Statement

Implemented Intrusion Detection System (IDS) to detect various types of attacks such as Denial of Service attack (Dos), Probing Attack, User to Root Attack (U2R) and Root to Local Attack (R2L) using three different machine learning algorithms.

Motivation

Traditional IDS are capable of detecting known attacks. However, it is important to detect new and unknown type of attacks without having any prior knowledge about it. To develop such a system, various machine learning techniques are being used. Here, I have implemented a model that predicts 'good' and 'bad' connections using Logistic Regression, Neural Network and Convolutional Neural Network.

Data Analysis

For this purpose, I have used NSL KDD Cup 1999 data set. It is a refined version of KDD Cup1999 data set. It consist of TCP connection data where each connection is labeled as 'Normal' or 'Attack' with exactly one specific type of attack. The attacks are categorized as follows:

1. DOS: Denial of Service Attack. Example: syn flood.
2. R2L: Root to Local Attack. Example: guessing password.
3. U2R: User to Root Attack. Example: various "buffer overflow" attacks
4. Probing. Example: port scanning.

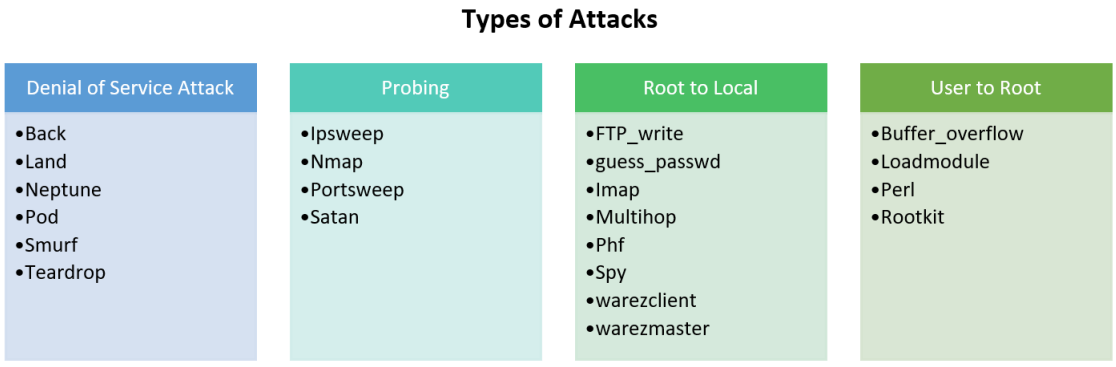

Figure 1: Categories and sub categories of attacks.

The data set consist of total of 24 Training attack types (belonging to one of the four categories mentioned above). Along with this, the test data consist of additional 14 attack types that are not included in the training data. These attack types are present in the test data only. This makes the task more realistic. A complete description of the data set can be found here [1]. The original KDD Cup 1999 data set has few shortcomings such as number of redundant records, uneven distribution of attacks etc. Including redundant records would have made the learning algorithm biased towards the frequent records and using data with unevenly distributed data of attacks would make cross validation difficult. Hence, I have used the refined version of this data set that handles these shortcomings.

Feature selection

Features of each connections can be classified into three categories.

1. Basic Features: Features extracted from the TCP connections.
2. Traffic Features: Features computed with respect to window interval.
3. Content Features: Features extracted from the data portion of connections. Example: Number of failed logins.

There are total 41 features in this data set that are categorized into three categories as mentioned above. Since, including all the these 41 features can make the model over fit, I have used Principal Component Analysis to reduce the number of features and include only those features that will help in determining the type of connection.

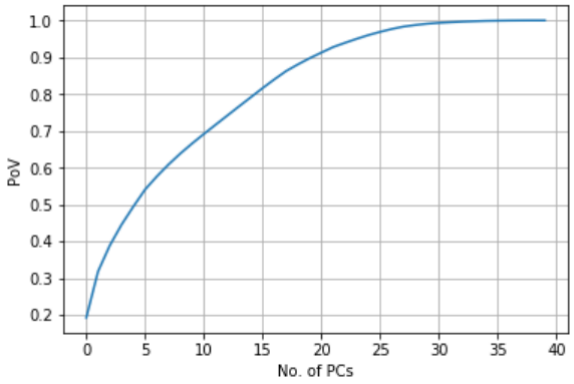

Figure 2: Proportion of Variance

As shown above, approximately 90% of variance is given by first 20 PCs. Thus, I have used ncomp = 20 in the following models.

Machine Learning Algorithms

Logistic Regression

One of the initial classification models, Logistic Regression is useful in classifying binary as well as multi-class data. I have used this model since it is easy and simple to implement and will act as a base line for other complex models. First, the data is scaled and then I have applied PCA. When trained on train data and tested on test data, this model gives an accuracy of approximately 73% on test data and approximately 95% on training data. One of the reasons why Logistic Regression does not perform well is that, it is not suitable for non-linear problems.

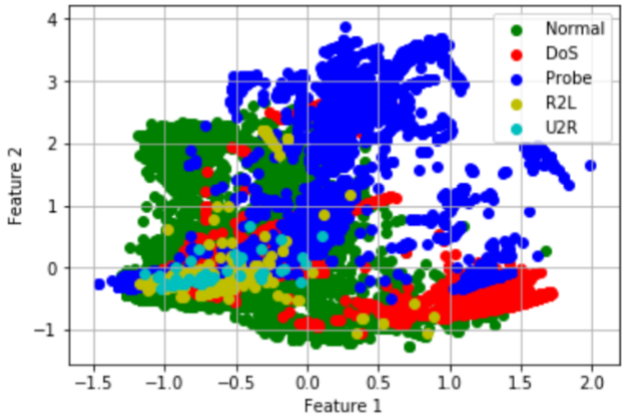

Figure 3:

By plotting various classes of attacks as shown in figure above, I found out that most of the classes overlapped and were not linearly separable. This makes it difficult for Logistic Regression to perform without significant loss.

Neural Network

Since, neural networks work well even on non-linear data, I have used it as my next algorithm in the IDS implementation. I have implemented a neural network that consist of 1 input layer, 2 hidden layers and 1 output layer as shown in the figure below. The hidden layers 1and 2 consist of 100 and 50 units respectively. Before feeding the data to the neural network, the data is normalized. Also, PCA is applied on the normalized data. I have used sigmoid activation function for hidden units and softmax activation function for output units. Along with this, I have used 'Adam' optimizer with a learning rate of 0.001. I tried using different learning rates with different number of epochs and batch sizes. I finally settled on learning rate of 0.001 with 30 epochs and batch size 100. With this configuration, the neural network gives an accuracy of approximately 78% on test data. It is not the best, yet better than logistic regression.

Convolutional Neural Network

As neural network also gave an accuracy of approximately 78%, it was tricky to understand the underlying reason behind the low accuracy on test data despite the high accuracy on train data. Perhaps, one of the main reasons behind low accuracy on test data was the fact that the test data consisted of 14 new attacks that were not present in the training data. Since, the signature of these attacks were not learnt by the model, it couldn't classify the unknown attacks. Some intrusion experts believe that most novel attacks are variants of known attacks and the "signature" of known attacks can be sufficient to catch novel variants. The motivation to use Convolutional Neural Network was to find out patterns in connections that are labeled as attacks.

I have implemented a convolutional neural network consisting of 1 input layer, 1 hidden layer and 1 output layer. I have used 1D convolutional layer with 1D Maxpooling and a Dense layer. As mentioned in [2], I have used 64 filters, each of length 5, learning rate of 0.001 with 5 epochs and batch size of 100. With this configuration, the convolutional neural network was able to perform better than neural network, with an accuracy of approximately 85% on the test data. Thus, with an even better combination of convolutional layers, filters and learning rates, CNNs can be used in identifying the unknown attacks.

References

[1] http://kdd.ics.uci.edu/databases/kddcup99/task.html

[2] Vinayakumar R, Soman KP and Prabaharan Poornachandrany, 'Applying Convolutional Neural Network for Network Intrusion Detection'

[3] Mahbod Tavallaee, Ebrahim Bagheri, Wei Lu, and Ali A. Ghorbani, 'A Detailed Analysis of the KDD CUP 99 Data Set'

[4] S. Devaraju, S. Ramakrishnan,'Detection Of Accuracy For Intrusion Detection System Using Neural Network Classifier'

[5] L.P. Dias, J. J. F. Cerqueira, K. D. R. Assis, R. C. Almeida Jr, 'Using Artificial Neural Network in Intrusion' Detection Systems to Computer Networks

[6] https://www.unb.ca/cic/datasets/nsl.html