Anomaly Detection in Network Traffic using Machine Learning Algorithms

Suvrorup Mukherjee
180907724, Roll No. 54
Section C, ECE
Manipal Institute of Technology
Manipal, India
rabi.rishi@gmail.com

Riddhi Garg
180907766, Roll No. 58
Section C, ECE
Manipal Institute of Technology
Manipal, India
riddhigarg26@outlook.com

Abstract— **Network Intrusion is a serious problem that is faced today. This project focuses on detection of anomalies in datasets containing data about network traffic from remote and local ISPs. Two datasets were obtained from different sources. An intrusion detection system is built using various Machine Learning algorithms. The project focuses on analysing various Supervised and Unsupervised Learning Algorithms , comparing them on the metric of their accuracy which deems them relatively fit to build an Intrusion Detection System.**

Keywords— TCP/IP, LAN, ASN, Intrusion Detection, Machine Learning, Network Analysis, Python, R

I. Introduction

A. Motivation

Network Traffics are always under threat of being vulnerable to malware, suspicious protocols and ciphers. In recent times it has become essential to monitor networks and optimize their performances to secure it's functioning and keep the thousands of routers and IP addresses connected with it safe. Network Traffic Analysis is a growing field which is devoted to such a cause. It involves collecting real-time and historical data to predict anomalies, troubleshoot a slow network, improve internal visibility and eliminate blind spots. Network-based intrusion detection systems (NIDS) are devices intelligently distributed within networks that passively inspect traffic traversing the devices on which they sit. The most common variants of NIDS are based on Signature Detection and Anomaly Detection. This project throws light on Anomaly Detection to monitor intrusions. We operate on two different datasets obtained from credible sources and analyse different Supervised and Unsupervised Learning Algorithms that are used to build these Intrusion Detection Systems. The results obtained from the study suggest what algorithms work in tracing connections to detect an anomaly of a particular instance.

B. Collecting Data

We obtained two datasets to conduct our study. Both of these data sets are quite different in structure based on the raw data that was collected.

The first Dataset was obtained from Stanford University's public repository on their statistical website. It consists of four columns of data with a summary of some real network traffic data from the past. The data covers 10 local workstation IPs over a three month period and their connections to remote ASNs over different periods of every day. However it doesn't contain any prediction variable to point whether a connection is anomalous or not.

The second dataset was obtained from Kaggle, which consists of a wide variety of intrusions simulated in a military network environment. It created an environment to acquire raw TCP/IP dump data for a network by simulating a typical US Air Force LAN. For each TCP/IP connection, 41 quantitative and qualitative features are obtained from normal and attack data. The predictor variable "class" contains two values : normal and anomalous to indicate the nature of the connection.

C. Softwares and Tools

To conduct the proposed research on this project we made the use of the programming languages R and Python. R was mainly used for exploratory data analysis on the two datasets and for implementing few of the algorithms. Certain libraries like 'ggplot2', 'caret', 'randomForests' and 'MASS' were used to complete the tasks. Python libraries like NumPy, Pandas, Keras, Scikit-Learn were used to retrieve the datasets, formulate the anomalies and to implement various Machine Learning algorithms.

II. Literature Review

In the datasets used, LAN and Remote ASNs are utilised in two different ways. While in the first dataset, there are two main categorical features- one feature is based entirely on local IPs and the second one is based on remote ASNs. These local IPs are 10 computer systems connected to a Local Area Network (LAN), i.e. a network of connected devices that exist within a specific location. An IP (Internet Protocol) address is a unique address that identifies a device on the aforementioned local network.
Whereas, each ASN is a set of Internet routable IP prefixes belonging to a network, that are all managed, controlled and supervised by a single entity or organization. Multiple such remote ASNs have been taken as a feature in the 1st dataset.

In the second dataset, a LAN was blasted with multiple intrusion attacks. A huge number of *connections* were built between the attacker and the LAN. A connection is a sequence of *TCP* packets starting and ending at some time duration between which data flows to and from a source IP address to a target IP address under some well-defined protocol. The Transmission Control Protocol (TCP) is a transport protocol that is used on top of IP to ensure reliable transmission of packets. TCP includes mechanisms to solve many of the problems that arise from packet-based messaging, such as lost packets, out of order packets, duplicate packets, and corrupted packets. Since TCP is the protocol used most commonly on top of IP, the Internet protocol stack is sometimes referred to as TCP/IP.

III. Methodology

A. *Getting and Cleaning the data*

The first dataset as mentioned in the introduction did not contain a predictor variable and hence required to design one. The data contained 'date', 'l_ipn', 'r_asn' and 'f count' as the four columns. The data description indicated four specific dates when a certain number of the 10 local IPs were compromised. Comparing these instances with the normal communication flow we devise an algorithm to create the prediction variable which is to qualify any connection as 'anomalous' if the f_count is above the Q3 median. Once we obtain the NaN values, we obtain our complete and tidy dataset.

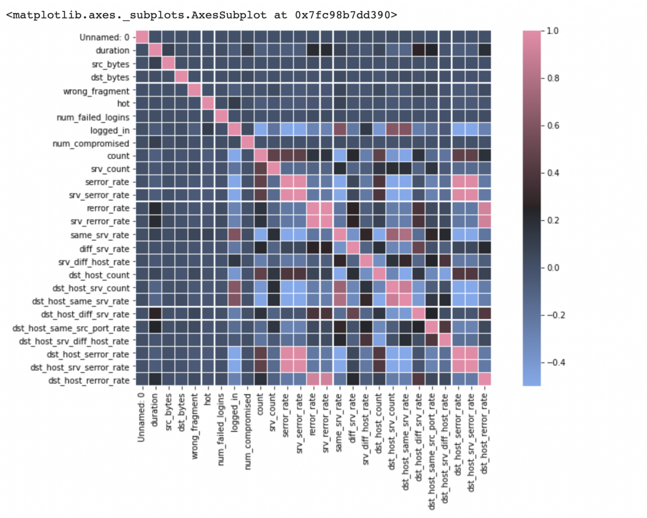
The second dataset already contains a categorical variable indicating whether a connection is normal or an anomaly. The set of 41 features have 3 qualitative and 38 quantitative features. We drop the qualitative features as they add no value to the following Machine Learning analysis. We further observe the rest of the quantitative features by making an importance chart signifying the importance of a variable to the predictor variable. We drop the last four features as they add null value and evidently so since they do not have any variation in data and hence have unusually large correlation. The presence of such data would give inaccurate results.

B. *Exploratory Data Analysis*

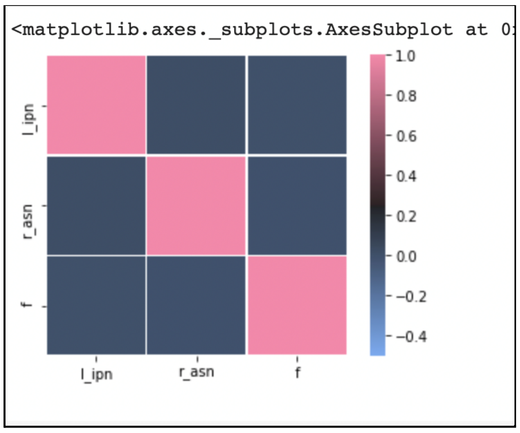
We plot the various anomalous instances as is mentioned in the data description of the first dataset with the f_count against their corresponding r_asn. To get a basis of reference we analyse the communication activity of the three months separately for each of the 10 IPs. The communication plot for July has been shown. We note very high or very low outliers indicate an anomalous connection. We further analyse the day-to-day activities of each of the local IPs. Having three levels of comparison we devise an algorithm to create a predictor variable. Since different r_asn groups show different activity , we divide the entire remote ASN range into five subgroups and obtain their respective third quantile count as the cutoff to qualify a connection as normal or anomalous. Any f count lying higher than the quantile signifies an anomaly. The third quantile is a justified basis for cutoff as the major chunk of the data lies within this limit and any f_count above the Q3 is so high that it biases the mean value to an above normal range. We thus obtain the final dataset to implement the algorithms.

We draw an importance chart on the second dataset owing to it's large number of features as shown. Accordingly we drop the last four features as they add no metric or basis to the predictor variable and/or have constant value thus adding unusually high correlation which would give incorrect results on further analysis.

The correlation graphs for both the datasets have been shown. The graphs clearly state that all the features in either of the datasets are completely independent hence we need not use bootstrapping algorithms to optimise feature selection.



```
<matplotlib.axes._subplots.AxesSubplot at 0x7fc98b7dd390>
```



```
<matplotlib.axes._subplots.AxesSubplot at 0
```

Figure 1. a) Correlation Chart for Dataset 2
b.) Correlation Chart for Dataset 1

C. *Anomaly Detection*

Since both the datasets qualify as classification problems having to predict either of two alternatives (normal or anomaly; or equivalently 0 and 1 respectively) we limit the analysis to using Supervised and Unsupervised Learning Algorithms. Supervised learning algorithms are trained using labeled data hence taking direct feedback to predict data. Unsupervised learning algorithms are trained using unlabeled data and hence do not use feedback.

We take the cleaned dataset of the local IP activity against the remote ASN connections. We split the dataset into a training dataset and testing dataset in the ratio 7:3. We apply four different Machine Learning Algorithms on each of the datasets. Algorithms used are Logistic Regression, Naive Bayes, Neural Networks, k-means, Random Forest and Linear discriminant analysis (LDA).

1. *Logistic Regression*

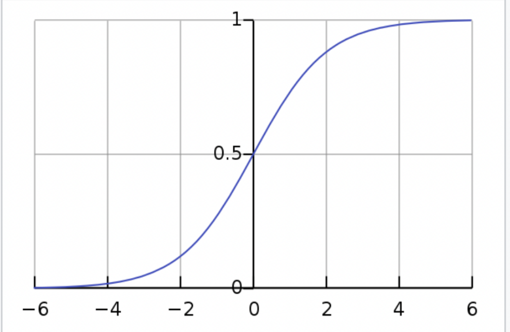
Logistic regression is a supervised learning classification algorithm used to predict the probability of a target variable. The nature of target or dependent variable is dichotomous, which means there would be only two possible classes. In simple words, the dependent variable is binary in nature having data coded as either 1 (stands for success/yes) or 0 (stands for failure/no).

Mathematically, a logistic regression model predicts P(Y=1) as a function of X.

A logistic function or logistic curve is a common S-shaped curve (sigmoid curve) with an equation:

$$f(x) = \frac{L}{1 + e^{-k(x-x_0)}},$$

where x_0 is x value of the sigmoid's middle point, L is the curve's max value and k is the steepness.



Standard logistic sigmoid function where $L = 1, k = 1, x_0 = 0$

2. *Naive Bayes Classifier*

A Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.

The Naive Bayes is called so because it is a) Naive, that is, it assumes that the occurrence of a certain feature is independent of the occurrence of other features and b) it depends on the principle of Bayes' Theorem. Bayes' theorem is used to determine the probability of a hypothesis with prior knowledge. It depends on the conditional probability. The formula for Bayes' theorem is given as,

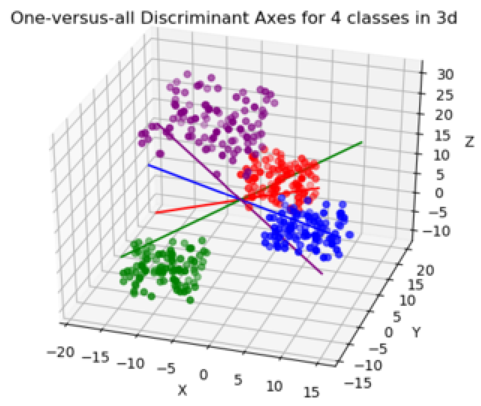
$$P(A \mid B) = \frac{P(B \mid A) \cdot P(A)}{P(B)}$$

A, B	= events
$P(A\|B)$	= probability of A given B is true
$P(B\|A)$	= probability of B given A is true
$P(A), P(B)$	= the independent probabilities of A and B

3. *Linear discriminant analysis (LDA)*

Linear Discriminant Analysis or LDA is a dimensionality reduction technique. It is used as a pre-processing step in Machine Learning and applications of pattern classification. The goal of LDA is to project the features in higher dimensional space onto a lower-dimensional space in order to avoid the curse of dimensionality and also reduce resources and dimensional costs.

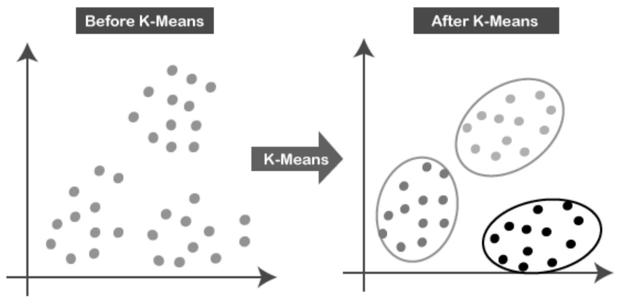
In the case where there are more than two classes, the analysis used in the derivation of the Fisher discriminant can be extended to find a subspace which appears to contain all of the class variability.



4. *k- means clustering*

The K-means algorithm in data mining starts with a first group of randomly selected centroids, which are used as the beginning points for every cluster, and then performs iterative (repetitive) calculations to optimize the positions of the centroids. It halts creating and optimizing clusters when either: a. The centroids have stabilized — there is no change in their values because the clustering has been successful. b. The defined number of iterations has been achieved.

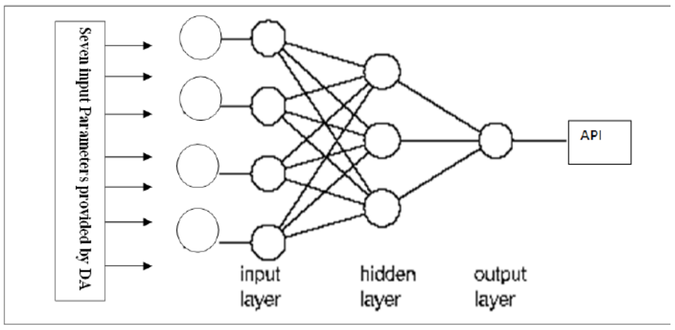
Here, a cluster refers to a collection of data points aggregated together because of certain similarities.



5. *Artificial Neural Networks*

Artificial neural networks (ANNs), usually simply called neural networks (NNs), are computing systems vaguely inspired by the biological neural networks that constitute animal brains. A neural network is an oriented graph. It consists of nodes which in the biological analogy represent neurons, connected by arcs. Each arc is associated with a weight while at each node. Apply the values received as input by the node and define the Activation function along the incoming arcs, adjusted by the weights of the arcs. We can apply Neural networks not only for classification but also for regression of continuous target attributes. A neural network usually contains the following 3 layers:

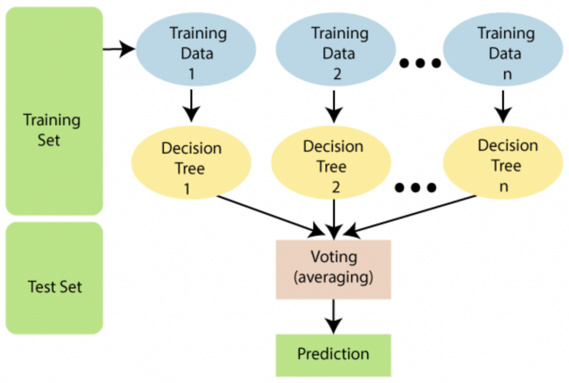
- Input layer – The activity of the input units represents the raw information that can feed into the network.
- Hidden layer – To determine the activity of each hidden unit. The activities of the input units and the weights on the connections between the input and the hidden units. There may be one or more hidden layers.
- Output layer – The behavior of the output units depends on the activity of the hidden units and the weights between the hidden and output units.



6. Random Forest

Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset. Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output. The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

The below diagram explains the working of the Random Forest algorithm:

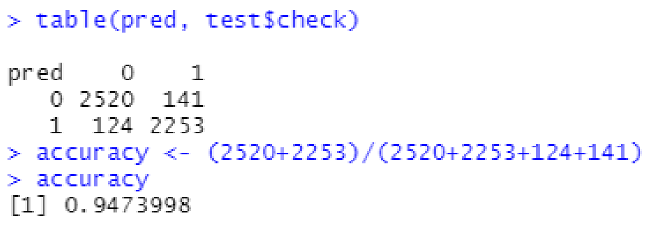


D. Calculating Respective Accuracies

After a model has been developed based on a specific algorithm on the training set, we produce the contingency table, which is a table between the predicted values and the true values. It enlists the number of true positives, true negatives, false positives, false negatives. The false positives are called Type I Error and the false negatives are called Type II Errors. Type I Error count is inversely proportional to the Type II Error count. The accuracy is calculated as follows :

Accuracy = (TP + TN)/(TP + TN + FP + FN)

This calculation is done across all the different models we have implemented. The following figure shows the same :



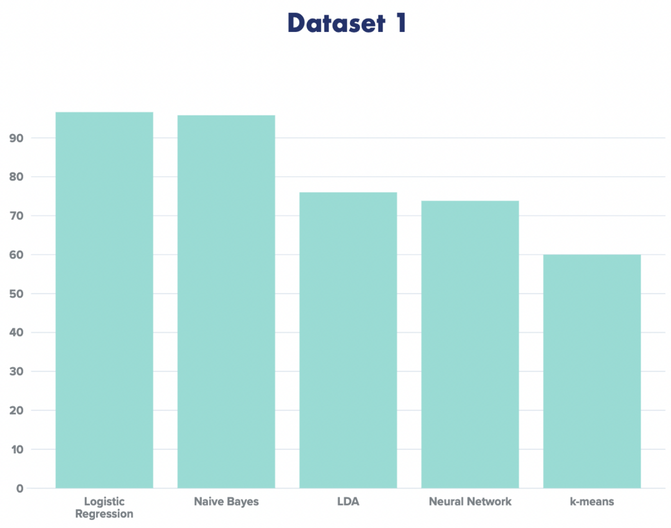
```
> table(pred, test$check)

pred    0    1
   0 2520  141
   1  124 2253
> accuracy <- (2520+2253)/(2520+2253+124+141)
> accuracy
[1] 0.9473998
```

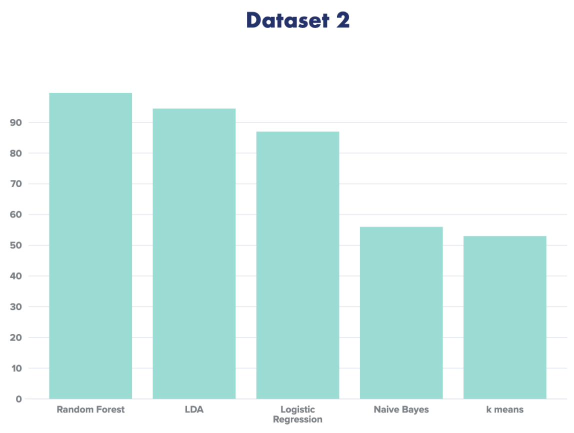
IV. Results

In the first dataset, it was observed that Logistic Regression gave the best output out of all the five algorithms used.

It can also be observed that supervised Machine Learning gave better output than Unsupervised Machine Learning(K means Clustering) and deep learning(Neural Networks).



In the second dataset, Random Forest gave an almost perfect accuracy score (more than 99%) followed by LDA.



V. Discussion

In our first dataset, all of the variables had a high degree of independence and were very less correlated, hence models like Logistic Regression and Naive Bayes performed really well.
Both the datasets and the problem statement have posed what is known as a classification problem in Machine Learning - when the prediction is between two alternatives. Logistic Regression and Linear Discriminant Analysis are two known linear methods to deal with classification problems and we notice the differing accuracies from both of them.
LDA is generally used to project the features in higher dimension space into a lower dimension space, since in the first dataset, the dimensions (features taken) were very less, it could not properly reduce them into subclasses and hence performed below average. Whereas in the second dataset, since the dimensions were high, its performance was extremely good.
Probable reasons why the Neural Network gave an average result in the first dataset could be the class imbalance (less number of anomalies in the predictor variable) or underfitting of the network (since the network doesn't have a large size due to a low number of continuous features).
We can see that k-means' performance was below average on both the datasets, this could have resulted from unequal division of clusters, variables having different variance or the distribution being non-spherical. Scaling of current datasets into larger ones might've improved the results however with limited computational power of home computers, that is out of the scope.
We know that Random Forest is intrinsically suited for multiclass problems and works well with numerical and interdependent features, which is exactly what we had in our second dataset. Instead of searching for the most important feature while splitting a node, it searches for the best feature among a random subset of features. This results in a wide diversity that generally gives a better model.

VI. Conclusion

This project aims to develop a Network traffic analysis system which performs really well in all the scenarios. From the results obtained after comparison of more than 5 algorithms in two dataset we can conclude that - if the features taken are independent of each other, a simple model like Logistic Regression can give a high rate of intrusion detection.
If the features are of various continuous and categorical types, having a good level of dependency- then a decision tree model like that of a Random Forest is a good choice.

Accuracy table for the first dataset(%):

Logistic Regression	Naive Bayes	Linear Discriminant Analysis	Artificial Neural Network	K Means
96.6	95.8	74	73.82	60

Accuracy table for the second dataset(%):

Random Forest	Linear Discriminant Analysis	Logistic Regression	Naive Bayes	K Means
99	94.5	87	56	53

VII. References

[1] http://statweb.stanford.edu/~sabatti/data.html

[2] https://www.kaggle.com/sampadab17/network-intrusion-detection

[3] Mohammed, Abdullahi. (2013). Network Traffic Analysis: A Case Study of ABU Network. Intelligent Systems Engineering. 4. 10.5120/2222-2863.

[4] B. Mukherjee, L. T. Heberlein and K. N. Levitt, "Network intrusion detection," in *IEEE Network*, vol. 8, no. 3, pp. 26-41, May-June 1994, doi: 10.1109/65.283931.

[5] https://kdd.ics.uci.edu/databases/kddcup99/task.html

[6] Markus Ring, Sarah Wunderlich, Deniz Scheuring, Dieter Landes and Andreas Hotho, "A Survey of Network-based Intrusion Detection Data Sets" .July 2019.