National College of Ireland

Network based Anomaly Detection: An ensemble approach

MSc Research Project
Data Analytics

Johan Bency
x16134753

School of Computing
National College of Ireland

Supervisor: Dr. Dominic Carr

National College of Ireland
Project Submission Sheet – 2017/2018
School of Computing

Student Name:	Johan Bency
Student ID:	x16134753
Programme:	Data Analytics
Year:	2016
Module:	MSc Research Project
Lecturer:	Dr. Dominic Carr
Submission Due Date:	11/12/2017
Project Title:	Network based Anomaly Detection: An ensemble approach
Word Count:	7057

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are encouraged to use the Harvard Referencing Standard supplied by the Library. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action. Students may be required to undergo a viva (oral examination) if there is suspicion about the validity of their submitted work.

Signature:	
Date:	13th December 2017

PLEASE READ THE FOLLOWING INSTRUCTIONS:

1. Please attach a completed copy of this sheet to each project (including multiple copies).
2. **You must ensure that you retain a HARD COPY of ALL projects**, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer. Please do not bind projects or place in covers unless specifically requested.
3. Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Network based Anomaly Detection: An ensemble approach

Johan Bency
x16134753
MSc Research Project in Data Analytics

13th December 2017

Abstract

Intrusion detection is a relatively mature domain. Network Intrusion Detection Systems have been used for over a long period of time for detecting potential intruders or attackers in a network or similar environment. Detecting and eliminating such threats is essential for the growth of any company in this cashless economy era where more and more users are moving to online banking, ecommerce and other related domains. With new technologies such as machine learning, neural networks, the advancements in computers such as processors and memory, and programming languages and libraries custom tailored for machine learning such as R, Python, matplotlib, scikit-learn etc. NIDS instead of being a vague Intrusion detection approach is now an accurate Intrusion detection system. The anomalies in network data are effectively predicted using such machine learning and neural network algorithms. In this project we are trying to create a model which works with a classic NIDS benchmark dataset in its backbone to create an accurate system which predicts network anomalies. Here we are using ensembled approaches for both sampling and for prediction phases for getting a superior result. For the sampling process we are using SMOTETomek algorithm which combine regular SMOTE for over-sampling and Tomek-link for under-sampling. For the classification approach we are using Support Vector Machines (SVM), K-Nearest Neighbors, Nave Bayes. The method is not claiming an absolute accuracy or performance, and it can be still be improved upon with better resources and tools.

1 Introduction

Hacking and network attacks are not a new term, the first known case of a network attack can be traced all the way back to the early 20th century when the now nicknamed Gentleman hacker sent inappropriate Morse code through a supposedly secure wireless telegraphy technology in a projector (Davidson and Hasledalen; 2014). From this incident to the even presently debated breaking of ENIGMA code during the second world war (Gaj and Orłowski; 2003), to the recent wannaCry attacks we can see that the attacks are becoming common and is targeting the common man along with big organizations and national Security. So, strengthening the defense is more important now than ever. Network Intrusion Detection Systems are tools which monitors network data for finding anomalies which signifies either policy violations or malicious content in a network (Jidiga

and Sammulal; 2014). There are different types of IDS tools ranging from anti-virus software to tools which monitor network traffic (Jidiga and Sammulal; 2014). IDS tools just detect the anomalies, basically it is a detection only system (Mehmood and Rais; 2016). There is intrusion prevention systems (IPS) which not only detects but also responds to anomalies (Mehmood and Rais; 2016). Anomalies could range from malwares, attacks, security policy violations to other common threats (Mehmood and Rais; 2016). In our research we are formulating a detection system not a prevention system. We are trying to detect anomalies which vary from the structure of good traffic. IDS can be categorized into two based on their detection approach namely Signature based IDS and Anomaly Detection based IDS (Mukkamala et al.; 2005). Signature based IDS is one such approach which uses pattern matching to finding known byte sequences or instruction sequences which can be linked to malware (Mukkamala et al.; 2005). This approach can be used to detect known attacks very easily however will fail to address the unknown attacks since it doesnt have any information regarding the new attack. The more recent approach uses machine learning algorithms to find the anomalies in the network and the anomalies corresponds to the network attack (Mukkamala et al.; 2005). Here a model is trained using known good traffic and is used to detect deviations from it. The major issue here is the false positive rates (Mukkamala et al.; 2005). This approach is called Anomaly Based IDS. In this research we focused on the Anomaly Based IDS approach. We are trying to improve the false positive issue we face in Anomaly Based approach by improving the dataset using sampling algorithms and then predicting the result using ensembled approaches.

Questions

1. How effectively can we reduce the false positive count if we are using SMOTETomek algorithm for sampling the dataset?
2. How much can we improve the accuracy of the Intrusion Detection System if we are using an ensembled approach of SVM, KNN and Nave Bayes?

In this cyber era where plastic currency, ecommerce and online banking controls the market new advanced threats such as advanced malwares, new types of network attacks and advanced crackers with high end hacking tools are also popping up to make the situation harder. So, the intrusion detection systems have to evolve beyond the regular approaches to account for all this. So that they can keep the internet safe. We are using the classic DARPA 98 datasets subset the KDD CUP 99 and is using python to code our solution. The reason for using the old dataset is justified in the previous works section below. We are using different libraries such matplotlib, pandas, scikit-learn and brew for different purposes. The pandas data frame is being used for holding the data and then for the ETL process as well. ETL being extract transform and load stage of the dataset. We are relying on matplotlib for plotting the data for helping with the ETL process. Scikit-learn is being used for the machine learning algorithms. Brew is used for ensemble algorithms. Using this we are trying to train a system using training data and then test the accuracy of the model with the test data.

In the scope of this research we will be covering the related work, methodology, implementation and finally we will evaluate the results.

2 Related Work

For this section we are making an assumption that the reader knows the basics of machine learning, datasets and classifiers. The detailed explanation of the terms and the technologies used are given in the next section.
In this section initially, we are covering the area of this project then which is network intrusion detection. Next, we will be explaining the existing machine learning approaches that has been done in this domain. Further, we will be covering the dataset KDD-Cup 99 and its source the DARPA 98 in detail. We are then covering the sampling algorithms and the need for sampling algorithms.

2.1 Area: Network Intrusion Detection

Network intrusion detection systems is the process of finding intruders in a network (Jidiga and Sammulal; 2014). Network Intrusion Detection System is the core of Intrusion Detection systems (Mehmood and Rais; 2016). Intrusion Detection System or IDS placed in a network will work just like network sniffer, basically finding the network attacks and then reporting to the network administrator (Kim et al.; 2014).
Generally, there are two types of IDS namely misuse detection and anomaly detection (Kim et al.; 2014). Misuse detection also called signature based network intrusion detection systems are using existing network signatures to find network attacks (Kim et al.; 2014). The signature based Intrusion System has one major advantage which is its low false positive rate, this makes the system a reliable option to find all the attacks which are previously known (Patcha and Park; 2007). Even though this appears as a viable option if the attack is not a previously known attack type it will be impossible to identify it (Kim et al.; 2014). This major disadvantage of regular signature based techniques are tackled in the anomaly based network intrusion detection systems (Kim et al.; 2014). The main advantage of Anomaly based NIDS over misuse detection is its ability to detect zero day attacks, which are essentially network attacks caused by vulnerabilities in the system which hasnt been noticed by the vendor and patched and also any other known attacks (Patcha and Park; 2007). Also, another advantage is that the profile for normal activity varies from system to system and so it is hard for an attacker to guess to tackle the security measure in disguise easily (Patcha and Park; 2007). Even with all the advantages this system has it has once alarming disadvantage which is it is high false positive rate. (Patcha and Park; 2007). Also, the system is highly complex and it is hard to predict which specific event triggered the alarm (Patcha and Park; 2007) Over the years a wide variety of data mining algorithms were used to implement an IDS, however the algorithms based on computational intelligence have found to have higher performance (Elhag et al.; 2015).
Anomaly based network intrusion detection is a relatively mature approach and different researchers have been trying to implement strategies which can outsmart attackers from early as 1997 where computer policy violations were considered as the anomaly (Lane and Brodley; 1997).

2.2 Previous approaches

As mentioned before a lot of approaches were attempted in Anomaly Based NIDS till date using machine learning. In one of such early approaches in 1997 the computer se-

curity policy violations were identified as the anomaly (Lane and Brodley; 1997). In this approach a mathematical function is used to find the similarity score for different sequences of data with the help of a mathematical function (Lane and Brodley; 1997). There have been approaches of finding anomalies even before this, however most of them were using pattern matching to equate two sequences byte to byte (Lane and Brodley; 1997). Since the approach presented in this research was revolutionary at the time we are considering it as the earliest NIDS approach (Lane and Brodley; 1997). In another approach signature based Ids is combined with anomaly based NIDS for superior results (Elbasiony et al.; 2013). The classic KDD 99 dataset was used for this approach and uses random forest for supervised learning for the signature base IDS module (Elbasiony et al.; 2013). For the anomaly based approach K-means clustering is employed as an unsupervised trained model (Elbasiony et al.; 2013). The research found out that owing to class imbalance issue there is an issue with the sensitivity of the model to minority instruction (Elbasiony et al.; 2013). For the anomaly based IDS approach using K-Means clustering we observed a very high detection rate and very high false positive value (Elbasiony et al.; 2013). In the case of signature based approach comparatively lower value of detection rate with very low false positive rates (Elbasiony et al.; 2013). The major disadvantage of this research being the false positive rate which could be tackled to a limit using sampling algorithms.

In another such approach the author is combining SVM classification and K-means Clustering for getting a better accuracy (Chitrakar and Chuanhe; 2012). The whole concept here is to use SVM to eliminate the requirement of a large dataset and then use K means for getting the best results (Chitrakar and Chuanhe; 2012). The Kyoto2006 dataset was used for training and testing purposes and received a better accuracy than its predecessor approach (KNN and Naive Bayes) in terms performance, accuracy and detection rate (Chitrakar and Chuanhe; 2012). SVM or Support Vector Machine is a very popular approach when dealing with NIDS (Shon and Moon; 2007). The linear SVM and other basic approaches which follows a supervised learning approach fails to trigger when dealing with zero day attacks and other novel attacks (Shon and Moon; 2007). In this research we are using unlabeled data and an unsupervised learning approach, hence the data can be unlabeled (Shon and Moon; 2007). However, this model cannot be used in a real-world network owing to its very high false positive rates (Shon and Moon; 2007). So a genetic algorithm along with filtering and packet profiling approaches are used to reduce the false rates to make it comparable to real world systems (Shon and Moon; 2007). In one of the latest implementations one class Support Vector Machine was implemented along with K-Mean Clustering approach (Nandkishor and Sunil; 2016). The tcpdump files which carries the network packet data of DARPA 99 dataset was analyzed for feature extraction (Nandkishor and Sunil; 2016). The extracted features are then used to predict anomalies in the network (Nandkishor and Sunil; 2016). A backtracking approach where the IDS is reshaped over and over is implemented to get the best results (Nandkishor and Sunil; 2016). The system was not able to identify mailbomb attacks from the TCP header (Nandkishor and Sunil; 2016). In both the researches the DARPA 99 dataset was used, which is similar to DARPA 98, the source dataset of our dataset KDD cup 99. Using unsupervised learning for both the approaches had bad results on the research which ended up with high false positive values.

Also in a different research Genetic Algorithms were used along with Kernel Principal Component Analysis or KPCA which is a nonlinear approach of extracting principal components using a kernel (Kuang et al.; 2014). In this approach a smaller subset of the

KDD CUP 99 dataset was used (Kuang et al.; 2014). Once the KPCA is used to extract the features of the dataset, multi class SVM was employed for determining whether it is an attack or not and the GA was used for selecting parameters for the classifier to prevent overfitting and under fitting (Kuang et al.; 2014). The proposed process was found to have a higher performance than the SVM approach which is what that has been rapidly been used with higher accuracy (Kuang et al.; 2014). In the current scenario network data comes in huge volumes and the traditional machine learning approaches and tools wont be able to create a successful IDS with the current data flow (Juvonen and Sipola; 2014). So, this research proposes an independent framework which relies on rule extraction where rules are just easy to understand conditions that can be used to classify data points (Juvonen and Sipola; 2014). The main disadvantages of this system are that the rules have to be presented in the precise structure for it to work and the training is a huge process and will create performance issues for the model (Juvonen and Sipola; 2014). Also with the latest growths in big data tools which handles machine learning well like Apache Spark this issue can be tackled in the near future. In another approach a case study on the real-time bank data was done using the new Rule Based Decision Tree machine learning approach for classification (Jidiga and Sammulal; 2014). The main motive behind this approach is to train a model for handling real time banking data, however the training data is considerably small (less than 7000) and the redundant data from main dataset is used as the test data (Jidiga and Sammulal; 2014). So, the performance of this system might not be reproduced on another dataset.

One major research done in this domain uses 13 different machine learning classifier models for detecting anomalies (Gulenko et al.; 2016). The models of Nave Bayes, SMO, part, jrip, oner, random forest, decision stump, decision table, random tree, logistic model tree, hoeffding tree, J48 and rep tree was discussed in this paper (Gulenko et al.; 2016). The aim here is to find anomalies in the cloud infrastructure (Gulenko et al.; 2016). Initially an accuracy of average of 92.4% was observed which diminished to 70% once the training and test data were fetched in a short time gap (Gulenko et al.; 2016). So, this model also couldnt satisfy the necessary requirements in a real-world scenario, A constantly learning training model could be a solution for this. A similar survey was done on various machine learning approaches for measuring up their effectiveness as an Anomaly based NIDS (Mehmood and Rais; 2016). Out of the four algorithms namely, Nave Bayes, Support vector Machine, Decision table and J.48 tested using the KDD CUP 99 dataset, none had a satisfying detection rate (Mehmood and Rais; 2016). The true positive rate or TPR wasnt considerably high (Mehmood and Rais; 2016). The issue with this can be identified as not identifying the potential class imbalance issue. We are making an attempt to resolve this issue by taking the class imbalance issue into perspective. In another survey, different Hybrid and normal anomaly based intrusion detection systems are compared with regarding to their speed, false alarm rate, scalability and other domains as well as their capability in detecting zero day attacks (Patcha and Park; 2007). Hybrid techniques are the ones which combines traditional approaches of misuse detection and anomaly based IDS to tackle the false positive rates of anomaly based IDS and to detect zero attacks limitation of the misuse detection IDS (Patcha and Park; 2007). Like in an ensembled approach we combine the different machine learning approaches in Hybrid NIDS entirely different IDS systems are integrated (Patcha and Park; 2007). These combinations cannot be done in random as it is not always the case that we might get a better system than the existing one, so we need to make sure these are compatible with each other and then decide on whether we should opt for a hybrid of these (Patcha and Park;

2007). Another new approach blends in a part of visualization for the anomaly detection part (Luo and Xia; 2014). A 4 -angle star where 5 anomaly classes are represented is in the heart of this approach (Luo and Xia; 2014). However, owing to this fixed number of classes that can be represented in this algorithm this is not a scalable approach (Luo and Xia; 2014).
Even though not a lot of researches have been done regarding ensembled approaches a very few of them was done in this regard with good success. Such a project combines the models of SVM, Artificial Neural Network or ANN and Multivariate Adaptive Regression Splines or MARS using the DARPA 98 intrusion detection dataset (Mukkamala et al.; 2005). This approach was found to outperform all the individual models and out of the individual models SVM had the highest accuracy (Mukkamala et al.; 2005). Another inference was a cent percent accuracy is theoretically achievable if the selection of attributes is done perfectly like using a Genetic Algorithm (Mukkamala et al.; 2005). Another similar research combines Regression Trees (RT) and Bayesian Networks (Chebrolu et al.; 2005). Even here the ensemble outperformed the individual models (Chebrolu et al.; 2005).

2.3 KDD CUP 99 and DARPA 98

DARPA datasets by MIT and its subsidiary KDD CUP 99 are the benchmark datasets for all anomaly based IDS approaches (Mehmood and Rais; 2016)(Kuang et al.; 2014)(Nandkishor and Sunil; 2016). Due to the increased threats in the cyber space in that era Defense Advanced Research Projects Agency or DARPA decided to create datasets for researches in the domain of IDS (Brugger and Chow; 2007). They used a simulated network atmosphere to create a dataset and it was named DARPA 98 (Brugger and Chow; 2007). Following years DARPA 99 and DARPA 2000 was also generated for the same applications (Brugger and Chow; 2007). By processing the network traffic files of DARPA 98 the KDD CUP 98 was generated (Elbasiony et al.; 2013). Since the DARPA dataset is created in a simulated environment and has very old attacks it cant be considered as the best solution for a new IDS (Brugger and Chow; 2007). However, two factors force the research to use KDD CUP 99 dataset. The first one being that if a model doesnt work well with the DARPA subsidiary then it wouldnt work well with the current network attacks as well (Brugger and Chow; 2007). The second and the most important one being the absence of any good reliable dataset rather than this (Brugger and Chow; 2007). So, in the current circumstances our best bet to create an anomaly based IDS is still KDD CUP 99.

2.4 Class Imbalance

The routine approach for an Anomaly Based IDS that runs using any machine learning algorithm is to get the network data with sufficient features to predict the result and then to categorize them using a target class (attack or normal) and then using the suitable algorithm(s) train the model over the data and validate it with the validation data (Qazi and Raza; 2012). However, in the majority of the cases of dataset due to an unequal number of instances of each class a problem which affects the performance (false positive rate) of the system arises and is called class imbalance problem (Qazi and Raza; 2012). For tackling this problem different sampling algorithms are in practice namely advanced sampling, over-sampling, under-sampling and Synthetic Minority Over-Sampling

Technique or SMOTE (Qazi and Raza; 2012). SMOTE is an approach of over-sampling the minority class using synthetic samples and in theory is more efficient than any of its counter parts (Qazi and Raza; 2012). SMOTE even though perfect in theory causes problems by creating samples around the border which could create noises in the data (Guo et al.; 2008). The Tomek link is an under-sampling approach which is a cleaning approach (Guo et al.; 2008). In this approach values when two samples of different target classes are found in the same region they are taken as a pair and one will negated as noise based on certain parameters (Guo et al.; 2008). When the data is skewed more than a limit approaches combining under-sampling and over-sampling are used and one such example is combining SMOTE and Tomek link which create the SMOTETomek sampling approach (Guo et al.; 2008). So due to our data being skewed it is a wise approach to use SMOTETomek for our sampling purposes.

3 Methodology

In this section we cover the various modules that are required for executing this project. The section starts with the process flow diagram of required modules and discusses the list of methods and technologies used for the acquisition of data, technologies used, data flow between various modules, data sampling approaches and the prediction models used to predict the network anomalies for detecting the network attacks.

3.1 Data background and acquisition

For building a good IDS the dataset used is equally important as the approach and algorithms used for the prediction model. The major constraint in the world of Anomaly Based IDS systems is the lack of good datasets. We are relying on the DARPA 98 dataset which is still considered the benchmark dataset and the best option for research in this domain by the current standards. The dataset is available at the DARPA 98 website. [1] This data set consists of 7 weeks of training data and 2 weeks of test data. The dataset mainly consists of tcpdump files which are nothing but the present-day cap files with all packet data and TCP header data in them. A Python package for http request was used for downloading all the files.
However, since we are focusing only on the tcpdump files of the DARPA 98 dataset we decided to choose the KDD 99 dataset which is generated by processing the tcpdump parts of the dataset. The dataset and its description were fetched from KDD CUP 99 archive.[2] This dataset has 42 attributes. Also, the dataset has 48,98,431 rows in the training dataset. Each row corresponds to a single packet. The dataset has details regarding 23 types of attacks.
The KDD 99 cup datasets training data is 708 MB in size. The dataset description is also available in the link along with different subsets of the major dataset. The data files are in CSV formats. Due to the limitations of our system we will be using the 10 percent sample of the file which is also available in the website which has the same ratio of records for each network attacks.

[1]DARPA 98 website: `https://www.ll.mit.edu/ideval/data/1998data.html`
[2]KDD CUP 99: `http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html`

3.2 Python

For the development purposes we had a wide variety of programming languages and GUI tools to choose from. We chose Python from a very crowded pool consisting of R, Java, MATLAB. Python is a very mature programming language which is used for a wide range of applications. Python is an open source language and has a very strong open source community for support. It is very fast in its machine learning capabilities compared to its major replacement R and has a very strong set of third-party modules and extensive libraries to choose from. The data frame data structure from pandas is very user friendly, fast, scalable and was designed for the machine learning and data science modules in mind. Different modules are available for different purposes we will look into the major libraries we use in detail.

3.3 Anaconda

Anaconda is the most popular python data science package out there. The package consists of different IDEs such as spyder, jupyter-notebook and packages such as scikit-learn, numpy, scipy. For our purposes of data science, we dont have any other alternative and Anacondas meet all our necessary requirements.

3.4 Jupyter-Notebook

For the python IDE we had an option to choose other normal IDEs such as visual studio code and spyder. However, Jupyter notebook has significant advantages over other regular IDEs. This range from line by line execution to stylized comments and an attractive UI. Also, this is open source and is fairly customizable as far as the keyboard short cuts and events are considered.

3.5 Pandas

We use pandas package of python for handling data. We use pandas dataframe for this purpose. The advantage of using pandas dataframe is its high speed, transformability and its long list of inbuilt functions focused on data science. Since our data set is very huge and we require a fair amount of data pre-processing pandas will be ideal for our scenario.

3.6 Scikit-learn

Scikit-learn is one of the most prominent Python machine learning libraries out there. Scikit-learn supports various machine learning algorithms for different purposes such as sampling, prediction algorithm and so on. In our project we require both sampling algorithms and predictor algorithms so scikit-learn could satisfy most of our requirements.

3.7 Matplotlib

Matplotlib is an open source plotting library of Python. Plotting is very useful for our project as we require this in our data preprocessing stage to plot correlations and to plot any graphs. Plots are always more useful than a table because human eyes always perceive visual graphics better than texts or regular tables. Also, some data which are

not obvious while viewing normal data will be more perceivable when looking at it as a graph or similar object.

3.8 Brew

Python brew is another open source python project used for machine learning. This package is mostly used for ensembling, stacking and blending. We are using this package mostly for ensembling the predictor algorithms and for ensembling the sampling algorithms.

3.9 Machine Learning

Machine learning is the domain of Computer science which deals with the ability of computers to learn scenarios without getting explicitly programmed (Livieris et al.; 2016). Machine learning deals with the development of algorithms for such tasks. Typically, machine learning cases include function learning, clustering and predive pattern (Livieris et al.; 2016). Data which is collected through previous observations will be used to learn the tasks (Livieris et al.; 2016). The addition of the observations will improve the learning process in the long run (Gulenko et al.; 2016). The whole idea behind the usage of machine learning is to automate the process more and more. This takes humans out of the equation. Machine learning can be related to other similar domains such Artificial Intelligence(AI), Knowledge Discovery Data(KDD), Statistics and Neural Networks. Machine Learning even though started more of as a research focused technology is moving more into the real world and commercial applications (Gulenko et al.; 2016).
In our project we rely on machine learning for training systems for predicting the anomalies in the network data as it can provide an environment which is self-taught and not entirely dependent on previous types of anomalies. We use various machine learning algorithms to find the outliers in the data to predict the anomalies.

3.10 Data Mining Algorithms

Data mining is the process of using machine learning or similar techniques for finding patterns in large datasets for predicting data (Gouda and Chandrika; 2016). Since we are using machine learning algorithms for data mining. As per examining the previous works it will be the best solution to use Support Vector Machine, K-Nearest Neighbor and Nave Bayes Algorithm. This is due to the higher accuracy we found for these algorithms in the previous works.

4 Implementation

In this project implementation we are trying to predict network anomalies for detecting network intrusions with a high accuracy and less false positive rate. Here we are dealing with 11 types of network attacks in our KDD 99 dataset. We are trying to predict whether the network packet or the network access can be tagged as a network attack or not and are not trying to identify what sort of attack does this actually can be categorized into.
In this section we are trying to explain the project implementation in detail.
The major stages of the implementation of this project is mentioned below.
1. Technical Environment

2. Data source
3. Python and libraries: Data cleaning and pre-processing
4. Create validation data set
5. Use sampling algorithms to create a synthetic sample
6. Building Machine Learning models for prediction using scikit-learn and brew
7. Ensemble the classifiers using brew
8. Make predictions using individual classifiers and the ensembled approach calculate different performance metrics

4.1 Technical Environment

1. Windows 10 Ultimate Edition
2. Anaconda 3.5
3. Python 3.6
4. Spyder IDE
5. Jupyter-notebook
6. Matplotlib library
7. Numpy library
8. Pandas library
9. Scikit-learn library

4.2 Data source

As mentioned above in the methodology section the KDD data source was fetched from the KDD cup website. Due to the limitation of the Machine used the nearly 5 million row datasets with 42 attributes is hard to process. So, we are opting for the smaller version of the dataset available at the small link which is 10 percent of the total dataset. In this dataset the normal and each type of attacks are carefully selected from main dataset to keep the proportions the same.

4.3 Python and libraries: Data cleaning and pre-processing

The dataset had its share of discrepancies and redundant attributes as expected of any other huge dataset. We had to do our fair share of cleaning and preprocessing before the dataset could be categorized as useful. We are using Python along with strong third-party libraries such as numpy, pandas, matplotlib for the process. We clear all null values and convert all string values to integer values. Using a custom plotting function which acts as a binder of matplotlib and the pandas corr() function a correlation graph was plotted. Using this graph all fields with a very high correlation value is removed for improving the speed of the learning and sampling algorithms.

We are only showing the correlation graph of 8 attributes. The bigger plot is too large for the document with 42 attributes.

4.4 Create validation data set

Validation data set is the portion of dataset that is held back for validating the trained model. This is also called test dataset. We are splitting 20% of the dataset for the

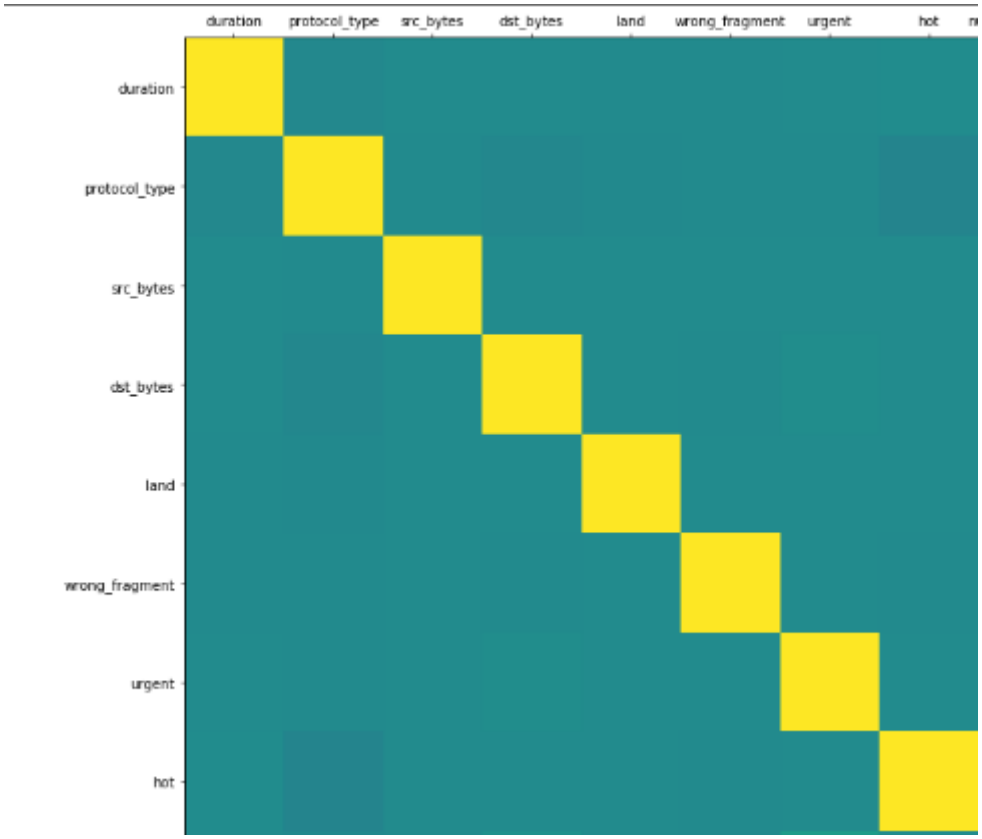

Figure 1: Plot of the correlation between different attributes in the dataset, yellow stands for 1 to 1 correlation

validation purpose. We are using the "train_test_split()" function in scikit-learn package to split the data. In the split data the percentage of each classes will remain the same as the major dataset aswell.

4.5 Sampling algorithms for creating synthetic sample

Sampling algorithms are used to evade the issues caused by class imbalance issues. We are attempting SMOTETomek algorithm which is an improvement of regular SMOTE algorithm. This improvement combines regular SMOTE with Tomek algorithm which essentially creates hard borders across 2 classes. If the two classes have shared borders it basically makes it hard to predict a value in that range. Tomek algorithm rectifies this. So, this algorithm is a good fit for our use.

4.6 Building Machine Learning models for prediction using scikit-learn

For predicting the network attacks, we are using various machine learning algorithms. We are using Support Vector Machine, K Nearest Neighbor, Nave Bayes individually to train separate machine learning models. We are using scikit-learn for the model training purpose. For each of these classifier algorithms scikit-learn has functions which can be used to train, predict the model and to even modify the parameters used in the algorithm to a certain extend. Each model is trained using the same dataset which was the output

of the SMOTETomek algorithm in the previous stage.

4.7 Ensemble the classifiers using brew

Python brew library is essentially used for the purpose of ensembling classifiers. We are combining all three classifier algorithms we tried out in the last step namely, KNN, SVM and Nave Bayes using the brew ensemble function. The ensembled approach is aiming for improving the performance of the Anomaly Detection System as a whole. In theory the machine learning model will have a higher specificity and sensitivity value due to this approach.

4.8 Make predictions using individual classifiers and the ensembled approach and calculate performance

For any machine learning model created the usefulness of the model can be assessed only after using it for predicting actual results. We have created a test dataset or validation dataset for this purpose previously. This will come into play in this stage. Using the predict function of scikit-learn we will be predicting how each model predicts whether each entry can be categorized to an attack or not. Finally, we will be doing the same with the ensembled approach. Once we get the predicted value we will be comparing this with the labels for that class to find accuracy of the model using the scikit-learn metrics module. Using the same module, we will be creating the confusion matrix and then calculate the true positive rate and true negative rates based on this inference.

5 Evaluation

In this section we will be discussing the results we have obtained for the machine learning approaches and how it has given us new insights regarding previous researches as well.
In our research we ran SVM, KNN, Nave Bayes algorithm for finding anomaly based NIDS. Afterwards we are running an ensembled approach of the three classifiers ran before.
As this is an anomaly detection system even though we have very high accuracy it wouldn't necessarily imply that the model is performing correctly. As with any other intrusion detection systems we are more concerned regarding the false positive value of the model. After prediction using trained data the confusion matrix generated is used find the true positive, true negative, false positive and false negative values. Also using equations for determining true positive rate and true negative rate we are determining the values of all four models.

TPR or True Positive Rate = TP / (TP + FN)

TNR or True Negative Rate = TN / (TN + FP)

For fetching the true positive, true negative, false positive and false negative values we need to obtain the values from the confusion matrix. The confusion matrix structure is as follows.

	ATTACK	NORMAL
ATTACK	True Positive	False Negative
NORMAL	False Positive	True Negative

Figure 2: Different performance metrics observed for the models when SMOTETomek was used for sampling.

5.1 Running models without sampling algorithm

Initially we are running all the 4 models, namely SVM, KNN, Naive Bayes and the ensemble of the three without using SMOTETomek for sampling the dataset.
We obtained models with very high accuracy.

Test Data = 100000	SVM	KNN	Naïve Bayes	Ensemble
TN	19396	19677	19451	19639
TP	79928	80111	78716	79966
FP	346	65	291	103
FN	960	155	1542	292
TNR	0.9824	0.9976	0.9852	0.9947
TPR	0.988	0.998	0.9807	0.996
Accuracy	0.98694	0.9978	0.9816	0.996

Figure 3: Different performance metrics observed for the models when sampling algorithms were not used.

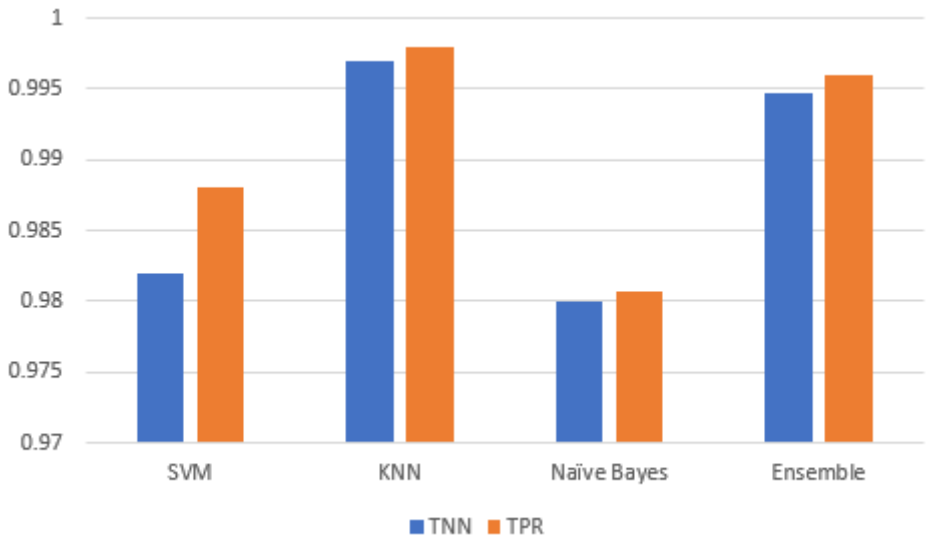

Figure 4: True Positive Rate and True Negative Rate of the four models when no sampling algorithm was employed.

5.2 Running models after Sampling using SMOTETomek

We are initially running the sampling algorithm SMOTETomek for handling the possible class imbalance issue. After this using the new synthetic dataset we are running all the four models again.
The sampling algorithm helped the models in decreasing the false positives. The ensembled approach improved overall.

Test Data = 100000	SVM	KNN	Naïve Bayes	Ensemble
TN	19592	19670	19425	19694
TP	78983	80111	78735	79298
FP	125	47	292	23
FN	1300	172	1548	985
TNR	0.9936	0.9976	0.9851	0.9988
TPR	0.9838	0.9978	0.9807	0.9877
Accuracy	0.9857	0.9978	0.9816	0.9899
Avg. Precision Recall	1	1	0.99	1

Figure 5: Different performance metrics observed for the models when SMOTETomek was used for the sampling.

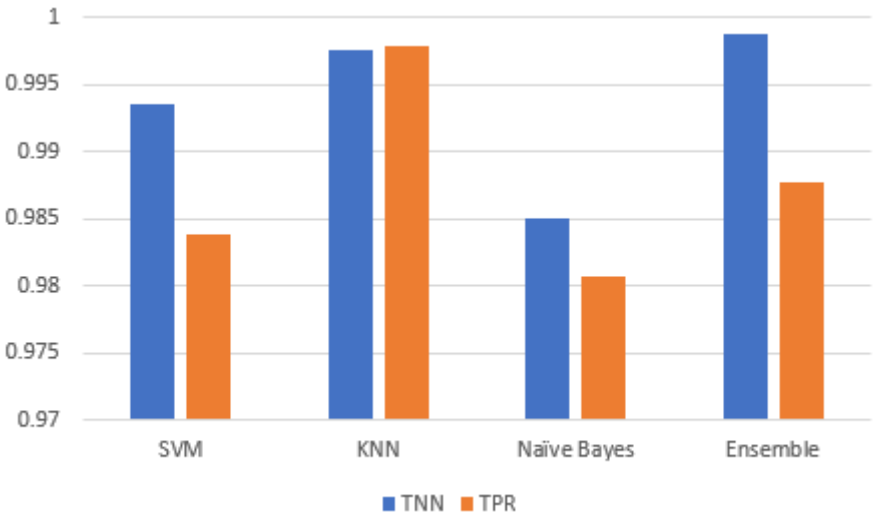

Figure 6: True Positive Rate and True Negative Rate of the four models when SMOTETomek was used for sampling.

5.3 Discussion

The model training and predicting process was run in two cycles. First using the regular sample and then using the SMOTETomek algorithm for handling class imbalance issue. We ran linear SVM, KNN, Naive Bayes and an ensemble of all three approaches.
The figure(3) shows the different performance metrics for each the of models when trained using normal dataset. We observed very high values of accuracy, true positive rate, true negative rate (98% +) for all the four models. The KNN approach out performed all the other three models in terms of all the three mentioned performance metrics. The accuracy of this model is 99.78%. Nave Bayes model has the lowest percentage of all four models at 98.1%, however the true negative rate of the model is slightly better than the SVM model which has the worst case of TPR of all the four. The ensembled approach has slightly higher accuracy than all approaches except KNN. The 99.6% accuracy and 99.6% TPR. As far as the counts are concerned out of the 100,000-sized validation set, 99,605 were correctly predicted were correctly predicted by the ensembled model. Figure (4) shows the true positive rate, true negative rate comparison of all the four approaches. The figure(5) shows the performance metrics of the 4 models which are trained using the SMOTETomek approach. SMOTETomek as explained removed any instances if two values qualify as a T link. Since after SMOTETomek the number of instances in each class is the same the result is the same as regular SMOTE. The accuracy of the model has decreased after the sampling algorithm ran by a small value. This is due to the increase happened in the false negative counts. However, the false positive value decreases by many folds after the sampling algorithm. The change in the metrics happened only for SVM, KNN and Ensembled approach. In Nave Bayes the performance metrics are the same before and after the sampling. The false positive value improved heavily for the ensembled approach. The false positive count decreased by 4-fold after the sampling approach. The ensembled approach the best false negative values of all the 8 instances in the research. However, the false negative value increased. Also, another important attribute precision has higher values after the ensembled approach. The figure (6) shows the comparison of TPR and TNR rates for these 4 models.

We have seen a drop in the accuracy after the ensembled approach. However, the most discussed attribute of the Anomaly based IDS is false positives. Wee found that the false positive count has decreased tremendously after the sampling approach. We can infer that the model did improve after sampling in that regard. The high accuracy of the model is due to two major reasons. The first one being the high number of attributes in the dataset and the second one being the number of redundant values in the KDD CUP 99 dataset. A large number of data in the dataset is redundant.
As part of the research papers we referred for this project we were observing SVM based classifiers had the highest performance. However, in our research we found that the KNN outperformed all other approaches even outperforming the ensembled one which was not what we anticipated.

6 Conclusion and Future Work

In the scope of this project we built an anomaly based network intrusion detection system. Network attacks are becoming more and more common these days and the network space is filled with cyber attackers. So, the IDS have to be shrewd and smart. For this approach we considered the class imbalance issue to be a major hurdle in achieving good results so we are using SMOTETomek sampling algorithm for create a synthetic dataset which is free from class imbalance issues. Further down the road we chose three machine learning models namely SVM, KNN and Nave Bayes by looking at the previous works done in the domain and choosing the ones with the best accuracy. Also, finally we tried to decrease the false positive count by attempting an ensembled approach of all the three classifiers. The results show that the sampling approach using SMOTETomek could provide the model with a minor performance boost and that too by sacrificing another performance metric. The research question which was under observation was whether we will be able to decrease the false positive count with such an approach. It was able to decrease the false positive counts for 3 out of 4 approaches including the ensembled approach as well. Now when we are looking at the ensembled approach the research was trying to determine whether we can increase the accuracy and other performance metrics comparing to the individual models. However, the ensembled approach out performed SVM and Nave Bayes in all aspects of performance, it could outsmart KNN only in the false count rates. The ensembled approach has very low false positive rates however when it comes to false negative rates it couldnt outperform KNN. So for the second research question we can conclude that it was able to outperform two models and had better false positive rates than KNN. The accuracy of the ensembled model is 99.6% which is the best we can get however KNN outperformed it in the long run.
The reason for this very high accuracy could be owing to the number of parameters we are using for training the model. Also, the dataset itself has a lot of redundant values and this could also be a major reason for the high accuracy.
In the future instead of improving on the machine learning approaches further research has to be put to create a new up to date dataset, as we are following an assumption that the model which doesnt perform well with DARPA dataset wouldnt be able to do the same with modern datasets. However, the inverse might not always be true. Even if it performs well with DARPA it doesnt necessarily mean that it will perform well in a modern environment. So, a new dataset which is not simulated and will have real network along with anonymous network data should be created to give some breathing space for

the developers.
Also owing to the big data explosion and the traditional systems migrating over to the big data side of things to stay in the pole position of the race, an anomaly based IDS should be formulated which can handle data of volumes high volumes in real time. With the new evolution in big data tools such as Apache Stream over Hadoop which has machine learning capabilities and the big data streaming frameworks such as Apache Spark streaming and Apache Storm and with the recent improvements in the processing capabilities this is possible now. Ideally this system should be able to handle the huge flow of network flow in real time and should train the model in a constantly evolving fashion periodically. Such a model will be able to handle with the constantly changing patterns and will give a near perfect accuracy in predicting the anomalies.

Acknowledgements

I would like to take this moment to thank my supervisor Dr Dominic Carr for his guidance and support all through my research work with his patience, motivation, and immense knowledge.

References

Brugger, S. T. and Chow, J. (2007). An assessment of the darpa ids evaluation dataset using snort, *UCDAVIS department of Computer Science* **1**(2007): 22.

Chebrolu, S., Abraham, A. and Thomas, J. P. (2005). Feature deduction and ensemble design of intrusion detection systems, *Computers & security* **24**(4): 295–307.

Chitrakar, R. and Chuanhe, H. (2012). Anomaly detection using support vector machine classification with k-medoids clustering, *Internet (AH-ICI), 2012 Third Asian Himalayas International Conference on*, IEEE, pp. 1–5.

Davidson, P. and Hasledalen, K. (2014). Cyber threats to online education: A delphi study, *ICMLG2014 Proceedings of the 2nd International Conference on Management, Leadership and Governance: ICMLG 2014*, Academic Conferences Limited, p. 68.

Elbasiony, R. M., Sallam, E. A., Eltobely, T. E. and Fahmy, M. M. (2013). A hybrid network intrusion detection framework based on random forests and weighted k-means, *Ain Shams Engineering Journal* **4**(4): 753–762.

Elhag, S., Fernández, A., Bawakid, A., Alshomrani, S. and Herrera, F. (2015). On the combination of genetic fuzzy systems and pairwise learning for improving detection rates on intrusion detection systems, *Expert Systems with Applications* **42**(1): 193–202.

Gaj, K. and Orłowski, A. (2003). Facts and myths of enigma: Breaking stereotypes, *Advances in CryptologyEUROCRYPT 2003* pp. 643–643.

Gouda, K. and Chandrika, M. (2016). Data mining for weather and climate studies, *Int. J. Eng. Trends Technol* **32**: 29–32.

Gulenko, A., Wallschläger, M., Schmidt, F., Kao, O. and Liu, F. (2016). Evaluating machine learning algorithms for anomaly detection in clouds, *Big Data (Big Data), 2016 IEEE International Conference on*, IEEE, pp. 2716–2721.

Guo, X., Yin, Y., Dong, C., Yang, G. and Zhou, G. (2008). On the class imbalance problem, *Natural Computation, 2008. ICNC'08. Fourth International Conference on*, Vol. 4, IEEE, pp. 192–201.

Jidiga, G. R. and Sammulal, P. (2014). Anomaly detection using machine learning with a case study, *Advanced Communication Control and Computing Technologies (ICACCCT), 2014 International Conference on*, IEEE, pp. 1060–1065.

Juvonen, A. and Sipola, T. (2014). Anomaly detection framework using rule extraction for efficient intrusion detection, *arXiv preprint arXiv:1410.7709* .

Kim, G., Lee, S. and Kim, S. (2014). A novel hybrid intrusion detection method integrating anomaly detection with misuse detection, *Expert Systems with Applications* **41**(4): 1690–1700.

Kuang, F., Xu, W. and Zhang, S. (2014). A novel hybrid kpca and svm with ga model for intrusion detection, *Applied Soft Computing* **18**: 178–184.

Lane, T. and Brodley, C. E. (1997). An application of machine learning to anomaly detection, *Proceedings of the 20th National Information Systems Security Conference*, Vol. 377, Baltimore, USA, pp. 366–380.

Livieris, I. E., Mikropoulos, T. A. and Pintelas, P. (2016). A decision support system for predicting students performance, *Themes in Science and Technology Education* **9**(1): 43–57.

Luo, B. and Xia, J. (2014). A novel intrusion detection system based on feature generation with visualization strategy, *Expert Systems with Applications* **41**(9): 4139–4147.

Mehmood, T. and Rais, H. B. M. (2016). Machine learning algorithms in context of intrusion detection, *Computer and Information Sciences (ICCOINS), 2016 3rd International Conference on*, IEEE, pp. 369–373.

Mukkamala, S., Sung, A. H. and Abraham, A. (2005). Intrusion detection using an ensemble of intelligent paradigms, *Journal of network and computer applications* **28**(2): 167–182.

Nandkishor, B. and Sunil, S. (2016). Analyzing network traffic to detect anomaly based intrusion using svm, *International Journal of Advanced Research in Computer and Communication Engineering* **5**(5).

Patcha, A. and Park, J.-M. (2007). An overview of anomaly detection techniques: Existing solutions and latest technological trends, *Computer networks* **51**(12): 3448–3470.

Qazi, N. and Raza, K. (2012). Effect of feature selection, smote and under sampling on class imbalance classification, *Computer Modelling and Simulation (UKSim), 2012 UKSim 14th International Conference on*, IEEE, pp. 145–150.

Shon, T. and Moon, J. (2007). A hybrid machine learning approach to network anomaly detection, *Information Sciences* **177**(18): 3799–3821.