NETWORK INTRUSION DETECTION ON KDD'99 DATASET USING VARIOUS MACHINE LEARNING TECHNIQUES

Internship Report

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF TECHNOLOGY

By

Nitin Kumar Gangwar
(16IS14F)

Under the Guidance of
Dr. N Balakrishnan

DEPARTMENT OF SUPERCOMPUTER EDUCATION AND RESEARCH CENTRE

INDIAN INSTITUTE OF SCIENCE

BENGALURU – 560012

July 2017

D E C L A R A T I O N

I certify that the report on “NETWORK INTRUSION DETECTION ON KDD’99 DATASET USING VARIOUS MACHINE LEARNING TECHNIQUES” which is being submitted as record of my internship is a bonafide report of the work carried out by me. The material contained in this report has not been submitted to any University or Institution for the award of any degree.

Nitin Kumar Gangwar

C E R T I F I C A T E

This is to certify that the report titled "NETWORK INTRUSION DETECTION ON KDD'99 DATASET USING VARIOUS MACHINE LEARNING TECHNIQUES" submitted to the Department of S.E.R.C., Indian Institute of Science, by Nitin Kumar Gangwar is accepted as a record of the work carried out by him as part of Summer Internship.

N. Balakrishnan
(GUIDE)

ABSTRACT

Network Intrusion is an attack from outside the organization and Intrusion Detection is a type of security management system for computers and networks. An ID system gathers and analyze the information from various area within a network or a computer to identify possible security breaches. In this paper, I have performed some machine learning techniques to identify Network Intrusion on KDD'99 dataset which is publicly available. All techniques are applied to 10% of KDD'99 original and filtered dataset. The data is filtered by me to remove the duplicates.

I have performed mainly three ML techniques : NaiveBase, J48 decision tree and Support Vector Machine. After getting the results of each classifiers I have performed some Ensemble techniques to get the better results than that of individual classifiers. I have performed Majority Voting(basic ensemble technique) and Stack Generalization to get the better results.

After these tasks I have also done an expertiment with Nueral Network like I have taken the outputs of individual classifiers and put them into a neural network to achieve some better results, its like ensembling only but the thing is I used a Neural Network to combine the results.

Finally I got to know that individually J48 and SVM are giving the good results compared to NaiveBase. If I talk about Ensembling, Stack Generalization and Neural Netwrok are far better than Majority voting.

Key Terms : Intrusion Detection, ML Classifiers, Ensemble technique, KDD'99(Knowledge Discovery and Data Mining)

INTRODUCTION

Because of heavy growth of technology, we saw that the applications running on various types of computer networks are increasing drastically. So this is the reason why security for the networks is becoming more important as well as complex task. So, we can use Intrusion detection systems (IDS) to detect anomalies and attacks in the network. These systems are dynamic in nature that is they collect and analyzes information from various areas within a computer or a network to identify possible security breaches.

Intrusion function includes[1]:

- Monitoring and analyzing both system and user activity.
- Analyze system configuration and vulnerabilities.
- Accessing system and file integrity.
- Ability to recognize the patterns typical to attack.
- Analysis of abnormal activity pattern.
- Tracing user policy violation

Basically there are three elements that are central to intrusion detection systems :
1. Resources that are to be protected in the target system.
2. Model that characterises the behaviour of the system to be normal or illegal.
3. The techniques that compares the actual system activities with the established model[2].

The goal of such a system is to have a high detection rate while keeping the false alarm rate as minimum as possible. There are two types of intrusion detection systems which are Host-Based (HIDS) and Network-Based (NIDS) [3].

- **Host Based Intrusion Detection System :** These systems reside on the host. It gains the knowledge of user activities and give alarm when it encounters any deviation from the regular user activity that is learned by the system and report to the admin.
- **Network based ntrusion Detection System :** It analyzes and monitors network traffic. It reads all the incoming packets and searches for suspicious patterns.

Furthermore there are two intrusion detection techniques : Misuse-Based and Anomaly-Based intrusion detection technique.

- **Misuse-Based intrusion detection scheme :** It maintains patterns or signatures that represent known attacks. It examines the network traffic for

such patterns in order to detect attacks. It fails to detect attacks whose patterns are not known.

- **In anomaly-Based detection scheme :** Any action that is different from the normal behaviour is said to be anomaly. It checks for the normal and abnormal behaviour of the system [4].It classifies using rules.
  In this, I have used some machine learning classification algoriths to classify the labeled data that we called is supervised learning. Basically there are two phase in every machine learning classifier :

- Training phase : In this phase , the model is being trained based on the labeled data.

- Testing phase : In this phase, the model is being tested and we get to know that how many instances are correctly classified by comaparing the model with original labeled dataset.

This paper analyzes three classifiers i.e. NaiveBase, J48 and SVM. I have used 10% of KDD data for traning as well as testing purpose. I have used preprocessed filter that is an unsupervised filter to remove the duplicates and then all three classifiers are experimented with this filtered data and I have compared the results of these classifiers with original 10% KDD data and filtered 10% KDD data.

LITERATURE SURVEY

A following survey is based on the KDD dataset. The training data set consists of seven weeks of traffic with around 5 million connections and the testing data consists of two weeks of traffic with around 300,000 connections. The data contains four main categories of Network Intrusions :

- Denial-of-service
- Remote-to-local (R2L)
- User to root (U2R)
- Probing

There are many attacks possible of each of above intrusion type. I am giving a small detail about each type of intrusion like attacks name, CVE-ID and comparison of accuracy of various ML techniques to detect the correct intrusion type.

S.No.	Attack Detected	Intrusion Type	CVE Index	Performace Evaluation using different Machine Learning Classifiers			
				Naive base	Decison Tree	KNN	Random Forest
1.	LAND	DOS	CVE-1999-0016	81.81	96.96	-	-
2.	SYN Flood	DOS	CVE-1999-0116				
3.	POD	DOS	CVE-1999-0128				
4.	SMURF	DOS	CVE-1999-0513				
5.	Teardrop	DOS	CVE-1999-0104				
6.	Satan	Probe	CVE-1999-1037	81.81	89.09	-	-
7.	Nmap	Probe	CVE-2001-				

			0896				
8.	InsideSniffer	Probe	CVE-2006-0182				
9.	SPY	U2R	CVE-2004-2676	97.07	-	99.89	-
10.	Buffer Overflow	U2R	CVE-2003-0652	99.694	99.938	99.969	99.969
11.	Load Module	U2R	CVE-1999-1586	99.983	100	99.979	100
12.	Perl	U2R	CVE-2016-1238	98.167	100	100	100
13.	Eject	U2R	CVE-1999-0027	94.431	99.892	99.921	99.924
14.	Guess_Password	R2L	CVE-2006-0834	98.389	99.062	98.98	98.98
15.	FTP_Write	R2L	CVE-2010-4096	97.797	99.96	99.96	99.96
16.	IMap	R2L	CVE-2002-0379	99.33	99.97	99.97	99.97
17.	Phf	R2L	CVE-1999-0067	99.927	99.98	99.98	99.98
18.	Netbus	R2L	CVE-2003-1475	90.89	99.93	-	99.90
19.	Httptunnel	R2L	CVE-2001-1087	83.414	99.45	99.3	-
20.	Neftp	R2L	CVE-1999-1333	95.75	-	-	-

Brief Idea about all four intrusion types :

- **DOS** - A denial of service attack is an attack in which the attacker makes some computing or memory resource too busy or too full to handle legitimate or geniune requests, or denies legitimate users access to a machine. These attacks directly target the server infrastructure. They make the online resources unavailable to the legitimate users.

Types of DOS attack : LAND(local area network denial), Neptune(SYN flood), POD(Ping of Death), Smurf, Teardrop, Back, etc.

- **PROBE** - The probe attacks are aimed at monitoring or gathering information about the vulnerability of a network or host. This information can later be used to exploit the privacy and security of the system.

  There are many number of programs have been distributed that can automatically scan a network of computers to gather information or find known vulnerabilities. These network probes are very useful to an attacker who wants to do an attack. An attacker who gets information like which machines and services are available on a network can use this information to look for weak points. Some of these scanning tools (Nmap, satan, saint, mscan) can quickly check hundreds or thousands of machines on a network for known vulnerabilities.

  Examples of Probe : Satan, Nmap, InsideSniffer, Ipsweep, ect.

- **U2R(User to Root)** - The attacker begins out as a normal user to the system. It then exploits various vulnerabilities of the system to gain the root access of the system. There are several different types of User to Root attacks. The most common is the buffer overflow attack. Buffer overflows occur when a program copies too much data into a static buffer without checking to make sure that the data will fit.

  Types of U2R attacks : Buffer Overflow, Spy, LoadModule, Perl, Eject, etc.

- **R2L(Remote to Local)** - In this type of attack the attacker gains unauthorized access to a local account on a remote machine on which it can send packets through a network.

  Types of R2L attacks : Dictionary, Ftp write, Imap, Phf, Netbus, Httptunnel, Neftp, etc.

One can study in detail about each and every attack from [5].

Jalil and Masrek [6] in their paper evaluated the performance of J48 classification algorithm and compared its result to two other machine learning algorithm that are Neural Network and support Vector Machine based on the detection rate, false alarm rate and accuracy of classification based on attack type.es.

Patil and Sherekar [17] have evaluated Naïve Bayes and J48 classification algorithm in the context of bank data set. Their focus was on measuring the performance of classification algorithm based on True Positive rate and False Positive rate.

Chandolikar and Nandavadekar [8] have evaluated the performance of J48 classification algorithm based on the correctly classified instances, Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), Root relative squared error and kappa statistics measures. They have applied feature selection on KDD cup data set before evaluating the performance of the algorithm.

Lee, Stolfo and Mok [9] have proposed a data mining framework to build intrusion detection models. According to them, learning rules that precisely capture the normal and intrusive behavior of activities can be used for detecting intrusions.

This paper compares the performance of Naïve Bayes, J48 and SVM algorithms on KDD'99 data set and studies the effects of removing redundant data from the dataset by applying preprocessing filter (i.e. RemoveDuplicate) present in Weka version 3.8.1 . After I have applied some ensemble techniques like majority voting , stacked generalization using meta classifier as J48 and stacked generalization using meta classifier as Neural Network and compare those results also.

EXPERIMENTAL ARROACH

A. Algorithms :-

NaibeBase : Bayesian classifiers are statistical classifiers. They can predict class membership probabilities, such as the probability that a given sample belongs to a particular class. Bayesian classification is based on Bayes theorem. Studies comparing classification algorithms have found a simple Bayesian classifier known as the naïve Bayesian classifier to be comparable in performance with decision tree and neural network classifiers. Bayesian classifiers have also exhibited high accuracy and speed when applied to large databases. [10]. Naïve Bayesian classifiers assume that the effect of an attribute value of a given class is independent of the values of the other attributes. This assumption is called class conditional independence. It is made to simplify the computations involved and, in this sense, is considered “naive”.

According to Bayes rule, the expression [11] for probability that class Y will have value Y i given the value of feature vector (X 1 ...X n), is calculated using this :

$$P(Y = y_k|X_1 \ldots X_n)=\frac{P(Y=Y_i)*P(X_1 \ldots X_n|Y=y_i)}{\sum_j P(Y=y_j)*P(X_1 \ldots X_n|Y=y_j)}$$

Since Naive Bayes assumes that all feature values are independent of each other so above equation can be written as :

$$P(Y = y_k|X_1 \ldots X_n)= \frac{P(Y=y_i)\prod_k(X_k|Y=y_i)}{\sum_j P(Y=y_j)* \prod_k P(X_k|Y=y_j)}$$

We need to find the most probable value of class attribute that is the value of variable Y, which can be found using following equation :

$$Y \leftarrow \arg\max P(Y = y_k)\prod_i P(X_i|Y = y_k)$$

J48 : It decides the target value of the new sample based on attribute value of available data. It recursively partitions the data.

There are internal nodes , edges or branches and leaf nodes. Different attributes are there on internal nodes. Edges tell us about the possible values of these attributes can have in observed sample. Leaf nodes tell us about the final value of the dependent variable.

- Step1 : first create a decision tree based on attribute value of available training data. So whenever it gets a set of items, it identifies the attributes which discriminates the instances more clearly. This attribute has the highest information gain. Among all values of a feature, if there is any value for which there is no ambiguity means same value for the target variable then terminate that branch.
  Eg : suppose if weather condition is forecast then 'A' team will lose the match , all other attribute's value does not matter than it will be terminated.

- Step2 : Then we look for another attribute which has highest information gain and continue in this manner.

SVM : Support Vector Machines are the supervised leaning models that are associated with some learning algoriths. These models analyze the data and can be used for classification or regression analysis.

- Goal of SVM : Finf the optimal seperating line which maximize the margin of traning data.
- Margin : Given a particular hyperplane, we can find the distance between the hyperplane and the closest datapoint, if we double it, we will get margin.
- These closest data points are called the SUPPORT VECTORS.
- Basically margin is no man's land.
- There will never be any data point in the margin.

Since I have multi class data and SVM is bydefault a binary classifier and there is no standard way for dealing with multiclass problems. So the basic idea to apply multiclassification to svm is to decompose the multiclass problem into several two class problem that can be addressed directly using several svm's. This approach only I have done wih svm.

Ensemble Techniques :

Heterogeneous classifiers – different learning algorithms over the same data
- Voting or rule-fixed aggregation
- Stacked generalization or meta-learning

Voting : Voting is the easiest ensemble methods. It is easy to understand and implement. Voting is used for classification. The first step is to create multiple classification/regression models using some training dataset. Each base model can be created using different splits of the same training dataset and same algorithm, or using the same dataset with different algorithms, or any other method.

Majority voting : Every model makes a prediction (votes) for each test instance and the final output prediction is the one that receives more than half of the votes. If

none of the predictions get more than half of the votes, we may say that the ensemble method could not make a stable prediction for this instance.

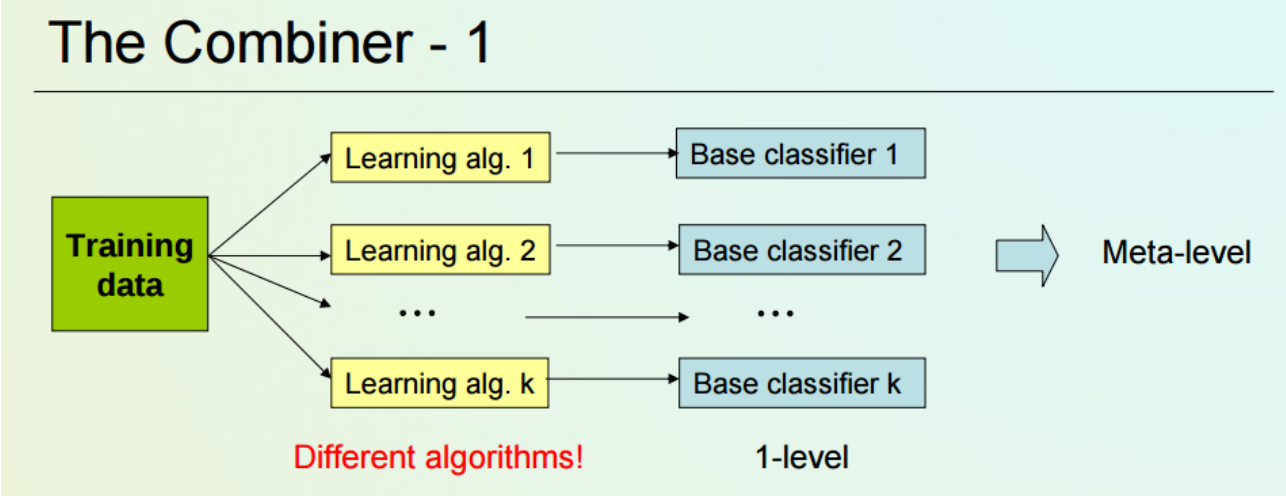
Stack Generalization :

I have performed stack generalization in two ways :

1. first I have chosen J48 as a meta classifier.
2. second I ahve chosen MultilayerPerceptron(Neural Network) as a meta classifier.

Now lets try to understand how stack genralization works :

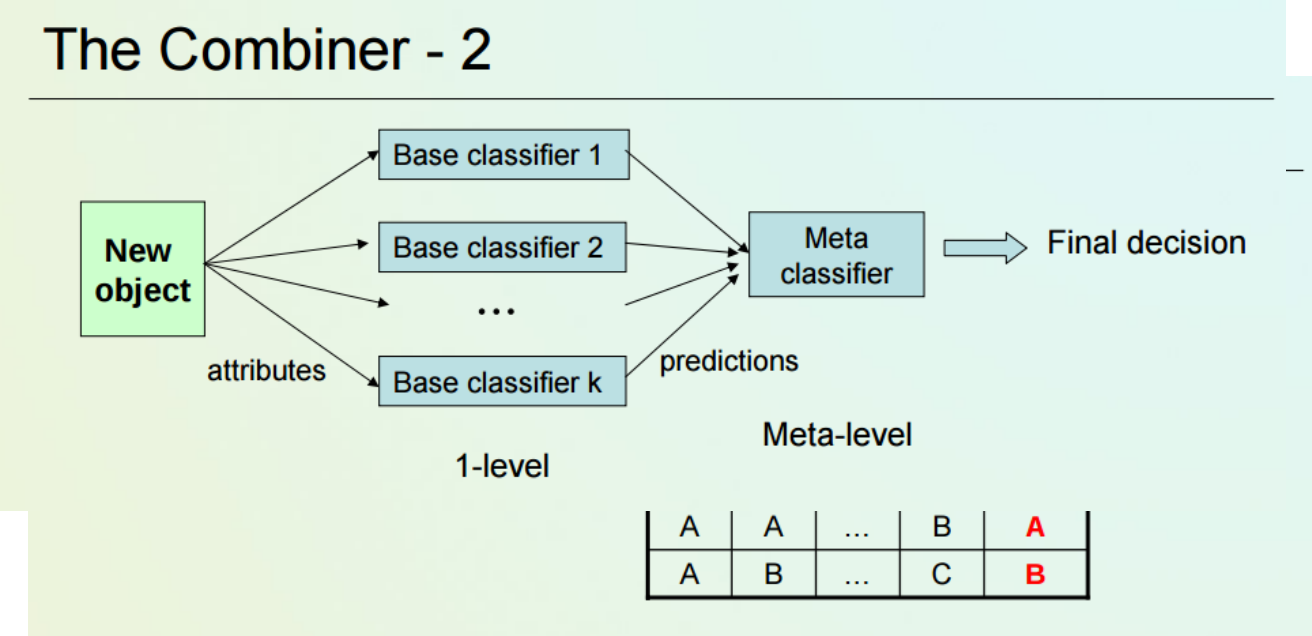
- Use meta learner instead of averaging to combine predictions of base classifiers
- Predictions of base learners (level-0 models) are used as input for meta learner (level-1 model)
- Method for generating base classifiers usually apply different learning schemes.
- Hard to analyze theoretically



Two-layered architecture:

- 1-level – base classifiers.
- 2-level – meta-classifier.
- Base classifiers created by applying the different learning algorithms to the same data.

- Predictions of base classifiers on an extra validation set (not directly training set – apply internal cross validation) with correct class decisions - a meta-level training set.
- An extra learning algorithm is used to construct a meta-classifiers.
- The idea - a meta-classifier attempts to learn relationships between predictions and the final decision; It may correct some mistakes of the base classifiers



• Other 1-level solutions: use additional attribute descriptions, introduce an arbiter instead of simple metacombiner.
• If base learners can output probabilities it's better to use those as input to meta learner
• Which algorithm to use to generate meta learner?
• In principle, any learning scheme can be applied
• Base learners do most of the work
• Reduces risk of overfitting

Neural Network :
In the above explaination, I have used J48 as a meta classifier but now I am using Neural Network as meta classifier and comapre both resutls.

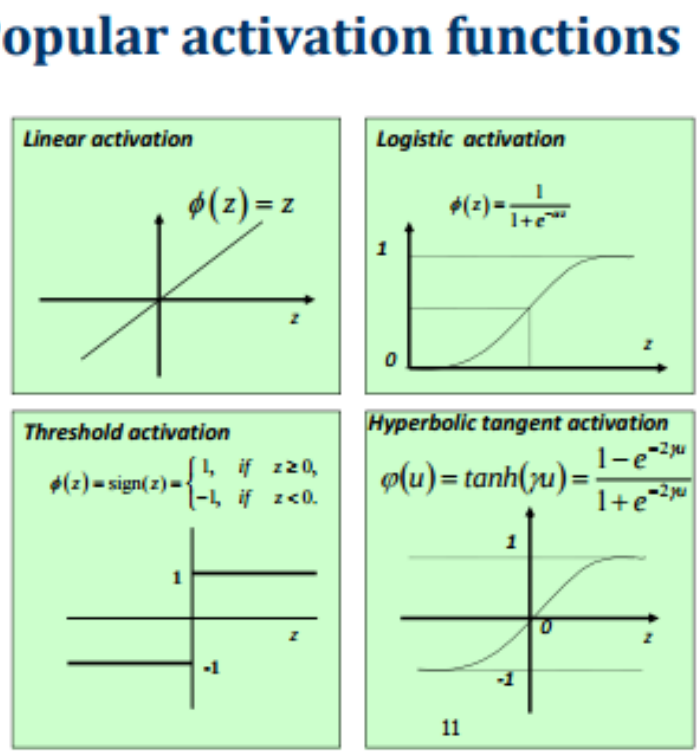
What is neural network(NN)?

- NN is a set of connected input output units(neurons) where each connection has weight.
- During the learning phase, network learns by adjusting the weight so that it can predict correct class label of the input sample.
- Weights are used to store the acuired information.

How Neural Networks Trained?

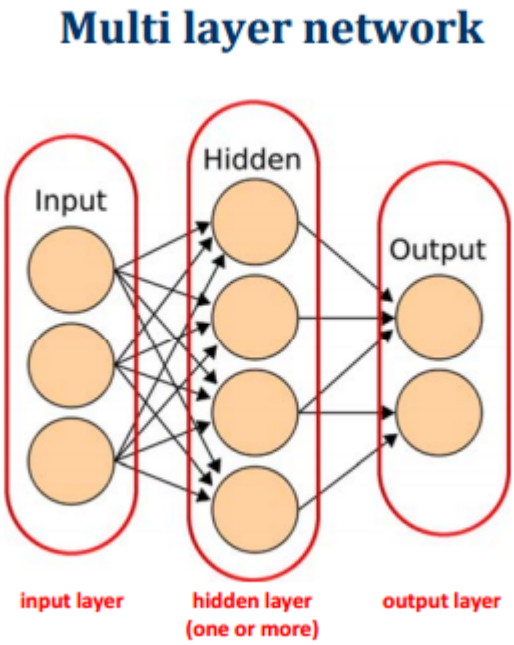
● Initially
» Choose small random weight
» Set threshold = 1
» Choose small learning rate

● Apply each member of the training set to the neural net model using a training rule to adjust the weight.
» For each unit
→ Compute the net input to the unit as a linear combination of all input to the unit.
→ Compute the output value using activation function.
→ Update the weights and threshold.



MultilayerPercepton(NN) : In the neural network basically there are three layers:

- Input Layer
- Hidden Layer(one or more)
- Output Layer



In my Neural Network, I have used two hidden layers. One hidden layer consists of a neurons where a = (number of attributes * number of classes) / 2, and second hidden layer consists of 10 neurons.

B. PREPROCEESSING FILTERS :-

Real-world data is often incomplete and lacks in certain behaviours or trends, may also contain many errors. Preprocessing of the data converts the raw data into a format that is useful for analysis and is easily to understand[12]. The weka.filters package includes classes that transform datasets by removing instances, resampling the dataset, removing or adding attributes, and so on. This package is divided into supervised and unsupervised filtering, which are further subdivided into instance and attribute filter [13]. I have used unsupervised instance filter called RemoveDuplicates. It removes all the duplicate instances from the first batch of data it receives.

C. DATASET :-

KDD‟99 Dataset was prepared by Stolfo et al, based on the data captured in DARPA‟98 IDS evaluation program. KDD'99 dataset is about 4GB in size prepared by monitoring 7 weeks of network traffic, which contains about 5 million connection records with each record taking about 100 bytes of memory.

It consists of 41 features and 4,940,200 instances. The 42nd feature is the class label which shows the attack category the instance belongs to and is determined by these 41 features. The following table represents all features and descriptions of each feature presents in dataset :

S.No.	Feature	Description
1.	duration	Duration of the connection
2.	protocol type	Connection protocol (e.g. TCP, UDP, ICMP)
3.	service	Destination service
4.	flag	Status flag of the connection
5.	source bytes	Bytes sent from source to destination
6.	destination bytes	Bytes sent from destination to source
7.	land	1 if connection is from/to the same host/port; 0 otherwise
8.	wrong fragment	Number of wrong fragments
9.	urgent	Number of urgent packets
10.	hot	Number of “hot” indicators
11.	failed logins	Number of failed logins
12.	logged in	1 if successfully logged in; 0 otherwise
13.	#compromised	Number of “compromised” condition
14.	root shell	1 if root shell is obtained; 0 otherwise
15.	su attempted	1 if “su root” command attempted; 0 otherwise
16.	#root	Number of “root” accesses
17.	#file creations	Number of file creation operations
18.	#shells	Number of shell prompts
19.	#access files	Number of operations on access control files
20.	#outbound cmds	Number of outbound commands in a ftp session
21.	is host login	1 if login belongs to the “host” list; 0 otherwise
22.	is guest login	1 if login belongs to the “guest” list; 0 otherwise
23.	count	Number of connections to the same host as the current connection in the past 2 seconds
24.	srv count	Number of connections to the same service as the current connection in the past two seconds
25.	serror rate	% of connections that have “SYN” errors
26.	srv serror rate	% of connections that have “SYN” errors
27.	rerror rate	% of connections that have REJ errors
28.	srv rerror rate	% of connections that have REJ errors

29.	same srv rate	% of connections to the same service
30.	diff srv rate	% of connections to different services
31.	srv diff host rate	% of connections to different hosts
32.	dst host count	Count of connections having the same destination host
33.	dst host srv count	Count of connections having the same destination host and using the same service
34.	dst host same srv rate	% of connections having the same destination host and using the same service
35.	dst host diff srv rate	% of different services on the current host
36.	dst host same src port rate	% of connections to the current host having the same src port
37.	dst host srv diff host rate	% of connections to the same service coming from different hosts
38.	dst host serror rate	% of connections to the current host that have an S0 error
39.	dst host srv serror rate	% of connections to the current host and specified service that have an S0 error
40.	dst host rerror rate	% of connections to the current host that have an RST error
41.	dst host srv rerror rate	% of connections to the current host and specified service that have an RST error

All features are divided into 3 groups :

- Basic features : It includes all features that can be linked to TCP/IP connection.
- Traffic features : These are relaled to time interval a connection is examined. They includes same host features and same service features.
- Contect features : These feature search for the suspecious behaviour in dataset.

This KDD’99 dataset contains four types of intrusions that I have explained earlier. These are :

- DoS(Denial of Service)
- U2R(User to Root)
- R2L(Remote to Local)
- Probe

The following tables compares the number of intrusion instances present for specific attack category in the original 10% KDD`99 dataset and the filtered 10% KDD`99 dataset.

The Filtered KDD`99 dataset was obtained by applying the pre processing unsupervised instance filter on 10% KDD`99 dataset.

The number of instances present in the 10% KDD`99 dataset= 4, 94,021

The number of instances present in the 10% filtered KDD`99 dataset= 1,45,586

- NORMAL (not an attack) :

Total instances in original 10% kdd data – 97277
Total instances in filtered 10% kdd data – 87831

- DoS :

Attack Type	Total instances in original 10% KDD data	Total instances in filtered 10% KDD data
TEARDROP	979	918
NEPTUNE	107201	51820
BACK	2203	968
LAND	21	19
POD	264	206
SMURF	280790	641

- U2R :

Attack Type	Total instances in original 10% KDD data	Total instances in filtered 10% KDD data
LOADMODULE	9	9
PERL	3	3
BUFFER OVERFLOW	30	30
ROOTKIT	10	10

- R2L :

Attack Type	Total instances in original 10% KDD data	Total instances in filtered 10% KDD data
FTP_WRITE	8	8
IMAP	12	12
SPY	2	2
PHF	4	4
MULTIHOP	7	7
GUESS_PASSWORD	53	53

WAREZMASTER	20	20
WAREZCLIENT	1020	893

- PROBE :

Attack Type	Total instances in original 10% KDD data	Total instances in filtered 10% KDD data
NMAP	231	158
SATAN	1589	906
IPSWEEP	1247	651
PORTSWEEP	1040	416

D. PARAMETERS USED :-

K-fold Cross Validation : The whole data is divided into k-sets and k-times the method is repeated. Each time a different set is used for testing and rest of k-1 is used for training the data. After k iterations average error across k-trials is measured.

I have used k=2.

True Positive rate : Ratio of instances that are correctly classified for a class to the total number of instances of that class.

$$\text{True positive rate} = \frac{\text{True positive}}{\text{True positive+False negative}}$$

Precision : ratio of instances that are actually belong to that class to the total number of instances that are classified as that class.

$$\text{Precision} = \frac{\text{True positive}}{\text{True positive+False positive}}$$

Recall : Ratio of instances that are classified as a given class to the actual number of instances belong to that class.

$$\text{Recall} = \frac{\text{True positive}}{\text{True positive+False negative}}$$

F-measure : It depends upon precision and recall.

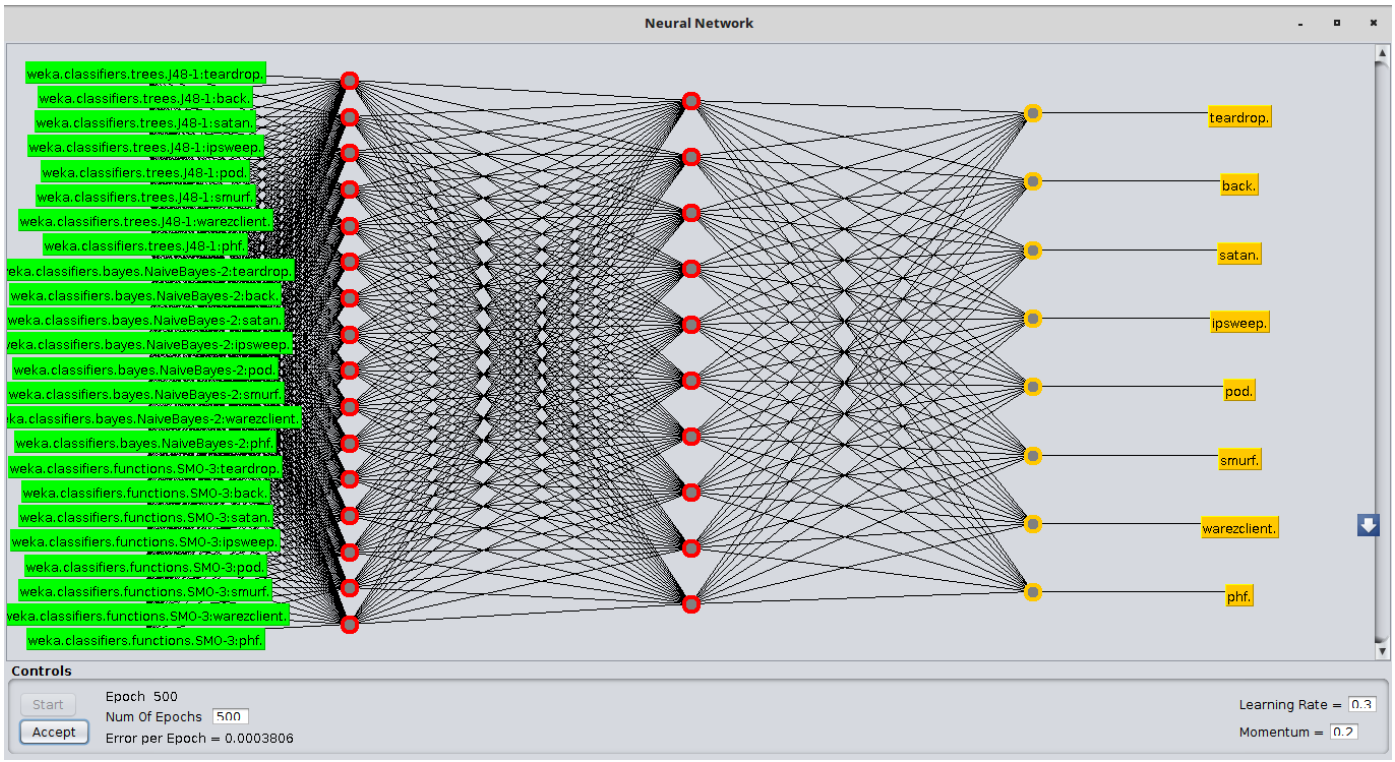
F-measure =(2*precision*recall) / (precision*recall)

RESULT

- All the experiment is done using WEKA 3.8.1 and Rstudio.

- With original 10% KDD'99 dataset, I have performed NBC, J48 and SVM.

- With Filtered 10% KDD'99 dataset, I have performed NBC, J48, SVM and all Ensembling technique, Since My system was taking so much time to perform ensembling with the original 10% KDD'99 dataset so I did perform various ensembling with filtered data only and compare the results, if your system is much faster than u can perform with original data also.
- Since when I used Neural Network as a meta classifier then my system is getting very slow and it took so much of time so I have performed the task only for 7 classes of attack and take the results. In this total number of intances are around 7000. My neural network looks this type :



Time taken to build the model for both 10 % original and filtered KDD dataset :

- Time taken to build the model using NBC :
  With original data : 11.59 sec
  With Filtered data : 3.53 sec

- Time taken to build the model using J48 :
  With original data : 107.14 sec
  With Filtered data : 49.28 sec

- Time taken to build the model using SVM :
  With original data : 859.17sec
  With Filtered data : 169.76 sec

Comparision of Naivebase , J48 and SVM :

Using 10% original dataset -

True Positve rate :
- NBC : 0.930
- J48 : 1.000
- SVM : 0.999

Precison :
- NBC : 0.990
- J48 : 0.999
- SVM : 0.999

F-measure :
- NBC : 0.951
- J48 : 0.999
- SVM : 0.999

Recall :
- NBC : 0.930
- J48 : 1.000
- SVM : 0.999

Using 10% filtered dataset -

True Positve rate :
- NBC : 0.781
- J48 :0. 998
- SVM : 0.997

Precison :

- NBC : 0.972
- J48 : 0.998
- SVM : 0.997

F-measure :

- NBC : 0.847
- J48 : 0.998
- SVM : 0.997

Recall :

- NBC : 0.781
- J48 : 0.998
- SVM : 0.997

Number of intrusion instances correctly as well as incorrectly classified by Naivebase , J48 and SVM :

Intrusion	Attack	Correctly classified instances in 10% original data (wrongly classified)			Correctly classified instances in 10% filtered data (wrongly classified)		
		By Naivebase	By J48	By SVM	By Naivebase	By J48	By SVM
DOS	TEARDROP	974 (5)	978 (1)	9749 (5)	914 (4)	917 (1)	911 (7)
	NEPTUNE	106821 (380)	107200 (1)	107200 (1)	51577 (243)	51800 (20)	51817 (3)
	BACK	2143 (60)	2191 (12)	2193 (10)	933 (35)	962 (6)	953 (15)
	LAND	19 (2)	1 17(4)	21 (0)	18 (1)	16 (3)	19 (0)
	POD	259 (5)	260 (4)	260 (4)	203 (3)	204 (2)	203 (3)
	SMURF	280383 (407)	280784 (6)	280786 (4)	639 (2)	634 (7)	637 (4)
U2R	LOADMODULE	6 (3)	1 1 (8)	0 (9)	0 (9)	0 (9)	0 (9)
	PERL	0 (3)	0 (3)	0 (3)	1 (2)	1 (2)	1 (2)
	BUFFER OVERFLOW	15 (15)	22 (8)	17 (13)	12 (18)	19 (11)	16 (14)
	ROOTKIT	5 (5)	0 (10)	0 (10)	3 (7)	0 (10)	0 (10)
R2L	FTP_WRITE	6 (2)	0 (8)	4 (4)	6 (2)	0 (8)	4 (4)
	IMAP	11 (1)	2 (10)	10 (2)	11 (1)	3 (9)	10 (2)
	SPY	2 (0)	0 (2)	0 (2)	2 (0)	0 (2)	0 (2)
	PHF	3 (1)	4 (0)	0 (4)	3 (1)	4 (0)	0 (4)
	MULTIHOP	2 (5)	2 (5)	0(7)	2 (5)	0 (7)	1 (6)

	GUESS_PASSWORD	50 (3)	50 (3)	50 (3)	50 (3)	50 (3)	51 (2)
	WAREZMASTER	16 (4)	16 (4)	15 (5)	17 (3)	15 (5)	15 (5)
	WAREZCLIENT	458 (562)	987 (33)	938 (82)	438 (455)	829 (64)	814 (79)
PROBE	NMAP	111 (120)	223 (8)	223 (8)	29 (135)	133 (25)	149 (9)
	SATAN	1513 (76)	1568 (21)	1546 (43)	852 (54)	881 (25)	861 (45)
	IPSWEEP	1163 (84)	1240 (7)	1219 (28)	561 (90)	639 (12)	630 (21)
	PORTSWEEP	966 (74)	1025 (15)	1034 (6)	324 (92)	400 (16)	410 (6)
NORMAL	Normal(not an attack)	64319 (32958)	97208 (69)	97131 (146)	57036 (30795)	87771 (60)	87696 (135)

Correctly classified instances of each intrusion in percenatge :

Intrusion	Using original data			Using filtered data		
	Naivebase	J48	SVM	Naivebase	J48	SVM
DOS	99.78	99.99	99.99	99.47	99.93	99.94
U2R	50	44.23	32.69	30.76	38.46	32.69
R2L	48.66	94.22	89.52	52.95	90.19	88.70
PROBE	91.38	98.75	97.93	82.87	96.33	96.19
NORMAL	66.11	99.92	99.84	64.93	99.93	99.84

Now I have performed some ensemble techniques to get better accuracy than that of individual classifiers.
The experiment is performed only with filtered data because of my system.

- The following table describes the comparision among Majority voting, Stacking using J48 as a meta classifer, Stacking usind Neural Neatwrok as a meta classifiers.

Correctly classified instances (Incorrectly classified) by all three Ensemble Technique (With filtered data):
Majority Voting : 145311 (275)
Stacking (J48 meta classifier) : 145264 (322)

- Total instances to test the Neural Network is 5187(because of my system)

Stacking (Neural network meta classifier) : 5186 (6)

Accuracy In Percentage :

Majority Voting : 99.77
Stacking (J48 meta classifier) : 99.81
Stacking (Neural network meta classifier) : 99.88

- Comparision of All three Ensemble Techniques :

Techniques	TP rate	Precision	Recall	F-measure
Majority Voting	0.998	0.998	0.998	0.998
J48 as a meta classifier	0.998	0.998	0.998	0.998
Neural network as a meta classifier	0.999	0.999	0.999	0.999

CONCLUSION

I have tested KDD'99 dataset wih Naivebase, J48 and SVM and finally you can see that J48 and SVM are giving far better result than Naivebase. But in building model NBC is taking the least time, so we can say that if you want good accuracy then your model will take more time in training.

I have perforemed some Ensemble technique to combine the result of NBC, J48 and SVM to get better classification than that of individuals. And the above table shows Stacking using J48 as a meta classifier and using Neural Network as a meta classifier is increasing the the correctly classified instances as compared to j48 and SVM alone but Majority Voting is not that effective as comoared to Stacking.

So we can see that we have enhanced the overall classification by combining all three classifiers. With filterd data NBC is correctly classifying 78.08% instances, J48 is 99.78% and SVM is 99.73% , so by combining then with stack generalization we can get 99.88% classification accuracy.

REFERNCES

[1] https://en.wikipedia.org/wiki/Intrusion_detection_system

[2] K.Lahre, T. Diwan, P. agrawal, S. K. Kashyap, "Analyze different approaches for IDS using KDD‟99 data set", International Journal on Recent and Innovation Trends in Computing and Communication, August 2013,pp. 645-651.

[3] Bilal Maqbool Beigh, "A New Classification Scheme for Intrusion Detection Systems", I.J. Computer Network and Information Security, 2014, 8, 56-70

[4] Ashutosh Gupta, Bhoopesh Singh Bhati, Vishal Jain," Artificial Intrusion Detection Techniques: A Survey", I.J. Computer Network and Information Security, 2014, 9, 51-57.

[5] https://www.ll.mit.edu/ideval/docs/attackDB.html#anypw

[6] 2013 Kamarularifin Abd Jalil, Mohamad Noorman Masrek, "Comparison of Machine Learning Algorithms Performance in Detecting Network Intrusion", International Conference on Networking and Information Technology, IEEE, 2010, pp 221-226.

[7] International Conference, IEEE, Sept. 2012, pp 1-5. T. R. Patil, S. S. Sherekar,"Performance Analysis of Naive Bayes and J48 Classification Algorithm for Data Classification", International Journal Of Computer Science And Applications Vol. 6, No.2, pp-256-261, Apr 2013

[8] N. S. Chandolikar, V. D. Nandavadekar," Efficient Algorithm for Intrusion Attack Classification by Analyzing KDD Cup 99", Wireless and Optical Communications Networks (WOCN), 2012 Ninth International Conference, IEEE, Sept. 2012, pp 1-5.

[9] W. Lee, S. Stolfo, and K. Mok, "A Data Mining Framework for Building Intrusion Detection Models", Proc, Of the 1999 IEEE Symposium on Security and Privacy, IEEE, May 1999.

[10] J. Han, and M. Kamber, "Data mining: concepts and techniques'"(2nd ed.). Morgan Kaufmann Publishers, 2006.

[11] Tom M. Mitchell, Machine Learning, McGrawHill, 2015

[12] Data Preprocessing. Available on: https://www.techopedia.com/definition/14650/data-preprocessing.

[13] Weka. Filters package. Available on: http://weka.sourceforge.net/doc.dev/weka/filters/Filter.html