National College of Ireland

Application of Machine Learning Models for Network Intrusion Detection Systems Based on Feature Selection Approach

MSc Research Project
Data Analytics

Shibi Murugesan
X18170871

School of Computing
National College of Ireland

Supervisor: Dr.Pierpaolo Dondio

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Shibi Murugesan
Student ID:	x18170871
Programme:	Data Analytics
Year:	2019
Module:	MSc Research Project
Supervisor:	Dr.Pierpaolo Dondio
Submission Due Date:	12/12/2019
Project Title:	Application of Machine Learning Models for Network Intrusion Detection Systems Based on Feature Selection Approach
Word Count:	4922
Page Count:	21

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 required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	
Date:	12th December 2019

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

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):	

Application of Machine Learning Models for Network Intrusion Detection Systems Based on Feature Selection Approach

Shibi Murugesan
X18170871

12 December 2019

Abstract

The amount of internet usage among the industry has grown rapidly in day to day life. Network intrusion has become the major thread in terms of security and various attacks are affecting the system. Intrusion Detection Systems is one such key technique which helps in protecting the system information and detect the various attack more accurately. Proposing machine learning schemes has been increased rapidly to detect the intrusion detection.In this research study, NSL-KDD dataset is been experiments with various machine learning algorithms to classify the attack type. However, among implementing the classification models a little consideration is given to Feature Selection. In order to improve the accuracy performance two feature selection methods (Embedded Method and Filter Method) is proposed in this study. This study Results are analysed on one vs Rest class classification based of the proposed model with metrics such as Accuracy, Precision, f1- Score.

.

1 Introduction

The importance of detection of breaches in the security of networks has become a necessity with the growing trends of threats. Network Intrusion Detection Systems *NIDS* are certain applications which help administrators to detect such anomalies and breaches in networks. The development of flexible yet efficient *NIDS* is a challenging task and the requirements keep changing with unforeseen attacks that are continuously evolving. Various algorithms in Machine learning and deep learning have been adopted and developed over the years for creating a benchmark dataset for detection of intrusions in the networks. Analysis of previous literature shows different approaches by various researchers for creating efficient IDS application which have variable outcomes in terms of accuracy, precision and recall timing. The feature-based selection of tools for detection of network intrusion is essential for minimisation of computational timing and reducing model complexity levels. The present study proposes a *classification-based intrusion detection system* through an analysis of previous research and therefore supports feature extraction and dimensionality reduction in network administration processes.
The process of feature-based intrusion detection systems in networks include both utilisation and transformation requirements which indicates that customised application are

necessary for effectively handling network intrusions. The different machine learning detection approaches has been considered on *NSL-KDD* dataset.

1.1 Motivation and Background

The process of Network Intrusion Detection plays an important role in ensuring the security of information and technology performances in the modern world. The detection systems for network intrusions are widely spread, but often a choice of perfect detection processes becomes a challenge for the network administrators (Dong and Wang; 2016). Flexible detection system structure matching the variable needs on different networks is often not identified which lead to ineffective detection systems and security breaches in many cases. Large amounts of data related to network transactions and details of recent network activities are to be handled for detecting malicious activity and threats of security. Therefore, it is essential to make assessments for delivering intrusion information and detect intrusion possibilities. This study, therefore, attempts to compare and understand the different classical algorithms in Machine learning for detecting network intrusions, especially focusing on the use of the *NSL-KDD* dataset and recent updates on alternative approaches. Intrusion scope and definition of intrusion threshold degrees

can also be effectively handled through the use of effective machine learning and deep learning algorithms which are also discussed in (Mulak and Talhar; 2015). The techniques in machine learning for the detection of intrusion can also be used for the development of *NIDS* over new and changing threats in external environments. Gaining a better understanding of the concepts and determination of different data patterns through machine learning can, therefore, help in the formation of an efficient intrusion-detection model in this study.

1.2 Research Question

"How efficient is the feature selection methods (Embedded Feature Selection and Filter Feature Selection Method) in classifying the attacks on Network Intrusion Detection Systems using machine learning Models?"

The overall objective of this research is to experiment the performance of theNSL-KDD dataseton Embedded Method (Lasso and Elastic-net) and Filter method (Correl-ation based Feature Selection) by implementing them on the classification algorithms.

2 Related Work

2.1 Introduction

Different alternative machine learning approaches have been identified by various authors which are used for detection of anomalies and threats to networks. Detection of continuous breaches to secured networks is also an important function of network detection systems which have been highlighted in previous studies. The present section focuses on explaining the opinions and ideas presented in previous studies about the network intrusion detection approaches using machine learning algorithms and NSL-KDD dataset. It

also identifies the contrasting views of previous authors about the most efficient intrusion detection system for all networks.

2.2 A Study on Network Intrusion Detection

The most commonly identified challenges while trying to detect anomalies in networks involves differentiation among the large amounts of data for identifying normal and abnormal data in the data sets. Repetitive data create greater ambiguity in the data analysis processes and therefore the intrusion detection processes need to eliminate such transactional records in the data sets (Dong and Wang; 2016). The primary concept for detection of the network intrusion, therefore, involves the definition of the characteristics of malicious data on networks. The author (Javaid et al.; 2016) have mentioned that the classification systems are also required to be able to differentiate among the data on the basis of such characteristics for effective detection of malicious data. Furthermore, the modern advanced techniques in the detection processes of intrusion in networks include automatic encoding approaches which automatically calculate the location of nodes in data along with the assumption of normality in data sets (Dong and Wang; 2016). Contrarily,(Hsu et al.; 2018) have mentioned the use of short term memory-based neural networks for the detection of network intrusion which also uses less complicated memory-based definitions and detection processes. The work of (Dong and Wang; 2016) however indicates that in network security, the calculations of distances between nodes in a data can indicate that there are higher chances of presence of malicious data if the distances are longer. Other factors influencing the machine learning and deep learning implementation in the detection of network intrusions include the sanctity of data which are used for the detection of anomalies. According to (Ingre and Yadav; 2015), techniques such as normality poisoning are also used in many cases for detection of such anomalies in networks. Such techniques are, however, most commonly used for ensuring that anomalous data of not hidden with normal information masks. The authors (Pajouh et al.; 2016) also mentioned in this context that networks are often required to be manipulated by increasing traffic and information for detection of data sanctity and anomalies. The main evaluation-based indications in the Network instructions detection systems include two factors which are calculated namely Precision and Recall (Tang et al.; 2016).

2.3 Machine learning and Traditional Methods of Intrusion Detection

Detecting anomalies and intrusions in networks can involve various approaches using machine learning and deep learning algorithms. The study of (Jabez and Muthukumar; 2015) outlined previously that traditional methods of intrusion detection in the *NIDS* mainly included statistical stability detection and precision test testing which did not produce accurate results. The study of (Dong and Wang; 2016) mentioned the number of machine learning and deep learning techniques which are commonly used for detection of network intrusions such as Auto-encoder Dimensionality Reduction system and *Deep Belief Network (DBN)*. These techniques can also be combined to form hybrid detection systems which successfully improve the accuracy of anomaly detection and reduce time-complexities at the same time (Yin et al.; 2017). Analysis of data patterns are also essential for the detection of network intrusion threats and machine learning approaches

often include multidimensional techniques (speech recognition algorithms, image recognition and encryption) for the detection of unauthorized entries (Parsaei et al.; 2016).

2.4 Study on Deterministic and Probabilistic Analysis

The deterministic machine learning algorithms use smaller and simpler data sets for analysing the deviations and irregularities from normal patterns. The irregularities in the information data and network transactional data are analysed by IT professionals through comparisons with the baseline data (Bhuyan et al.; 2015). It is a simple process which detects all data that is beyond the normal thresholds to be intrusive actions. The systems which operate on knowledge principles of machine learning such as deterministic and probabilistic machine learning use specific models formulated for detection of anomalies within the network data. In the opinion of (Aljawarneh et al.; 2018), the probabilistic machine learning algorithms evaluate the underlying patterns in an assessment process which may be overlooked in simpler processes such as deterministic analysis. On the other hand, (Ahmad and Aftab; 2017) also mentioned that such systems are more relying on clustering for the detection of unsupervised activities and unauthorised access.

2.5 Study on Feature Engineering over *NSL-KDD*

The classification of the processes of *NSL-KDD* are optimised since the types of attacks are classified according to the types of malicious activities. The studies of (Aggarwal and Sharma; 2015) showed that the key issue in the network intrusion detection is safe communication during the detection process. Alternatively, (Chaudhari and Patil; 2017) also pointed out that the classification techniques in the KDD dataset pre-treatment effectively succeed in just classification of the traffic data in networks. The traditional intrusion detection methods are inefficient in many ways since many legitimate users can be detected as intruders with baseline comparisons. As per the study of (Farnaaz and

Jabbar; 2016), the *KDD-99* dataset has been a widely used data set for comparing the attacks. Many of the recent studies have also identified some issues in the *KDD-99* data set used for the detection of intrusion since the categorical classifications in this process are outdated according to most authors (Sharma and Gaur; 2016). The indications in the study of (Dhanabal and Shantharajah; 2015) also point towards the fact that there is some imbalance in the methods of classification of traffic in the *KDD-99*. The authors (Dhanabal and Shantharajah; 2015) suggest alternative updated versions such as *NSL-KDD* for detection whereas, (Dong and Wang; 2016) suggested **Synthetic Minority Over-Sampling Technique (SMOTE)** for mitigating such issues.

2.6 Use of *NSL-KDD* Dataset for Feature Selection and Intrusion Detection

The use of KDD datasets for comparison and classification of the threats on networks has been mentioned by many authors. Recently the *KDD-99* dataset has been replaced in most IT environments with the *NSL-KDD* dataset testing which is an improved version of the classifier model based on tree-based algorithms (Kevric et al.; 2017). The findings in the study of (Kevric et al.; 2017) also specifically mentioned that the degree of detection accuracy in case of the *NSL-KDD* dataset is about 89.24% which is the highest degree

achieved till date. Contrarily, the studies of (Hashem; 2017) focus on the utility of the KDD datasets in the detection of Denial of service attacks in cloud environments. The findings of the study of (Mulak and Talhar; 2015) also indicated that the efficiency of the *KDD-99* dataset in terms of accuracy for detecting *DoS* attacks was much higher as compared to the *NSL-KDD* dataset.

The works of (Ashfaq et al.; 2017) also present the fact that combines approaches in machine learning can lead to more effective detection of intrusion on networks. In modern technological environments, the most prominent threats arise from the DoS attacks on networks and therefore the *NIDS* are required to include all steps such as normalisation, discretion, feature selection and training and test testing (Hussain and Lalmuanawma; 2016)

2.7 Problems in Intrusion Detection

One of the core issues in the formulation of the intrusion detection systems is the accuracy and the capabilities of detecting different types of attacks on networks (Sultana et al.; 2019). Alternative machine learning algorithms capable of developing NIDS include **Artificial Neural Networks (ANN), Naive-Bayesian (NB), Random Forests (RF) and Support Vector Machines (SVM)** (Shone et al.; 2018). The testing of the class features in networks also turn out to be great challenges and the authors (Homoliak et al.; 2017) mentioned that machine learning algorithms such as SVM and ANN help in the calculation of neural nodes in networks which allows the architecture to learn about the class features.

2.8 Conclusion

The review of the literature shows that there are multiple machine learning algorithms which are used for detection of intrusions such as DoS attacks and Probe attacks. Some other intrusions types that can be identified by the KDD datasets also include the U2R and R2L attacks. Most of the authors have agreed that the new improved version of the *KDD-99* dataset named *NSL-KDD* dataset is more efficient in detecting various types of anomalies in networks. Contrasting views about the deficiencies of the algorithms, however, are evident in the study of previous literature.

3 Methodology

In all the research studies related to the data analytics field either **CRISM-DM (Cross Industry Standard Process for Data Mining) or KDD (Knowledge Discovery Database)** is implemented. In our research study we choose to implement KDD methodology which involves structured systematic process of implementing in order to gain insights for business which will be helpful for stakeholders in decision making and to reach the desired goal. In this research two tier approach has been implemented which are client tier and business logic tier. Methodology involves several step by step process:

1. **Dataset Selection**
   Dataset selection is the first and initial most step in the KDD methodology. For any research study the data set selection play a key role as the desired dataset should be selected reliable and to the need of study. The background of this dataset id explained in details in the following section. Below is the two different dataset used in this research study.

   - NSL-KDD
   - UNSW-NB15

2. **Exploratory Data Analysis**
   After selecting the data, the main step is to explore the features in the dataset to check the trends contributing for the class labels.

3. **Dataset Preprocessing**
   After exploring the data the features in the dataset is changed according to the desired input. Below are the major steps which are been done to make the dataset preprocessed.

   - Normalization
   - Re-sampling to make it Balanced dataset
   - One hot Encoder

4. **Feature Selection Techniques**
   This is major step in the KDD methodology where the more suitable feature have been selected which are all required to build the machine learning models. Here in this research study two major feature selection technique is been implemented before implementing into the model. This feature selection helps in selecting the best top rated features which need to considered. Below are the proposed methods used in this research study.

   - Embedded Feature Selection
   - Filter-based Feature Selection

5. **Implementation of Model on selected Features with NSL-KDD dataset**
   After sorting out the best feature through different feature selection methods the next process is to implement the classification models to test the performance of the build model.

   - With Classical Models
   - With Boosted Model and Feature Transformation
     After testing the build model with the test set the performance and their individual accuracy score is been evaluated and compared with the evaluation metrics.

6. **Result and Evaluation**

7. **Conclusion**

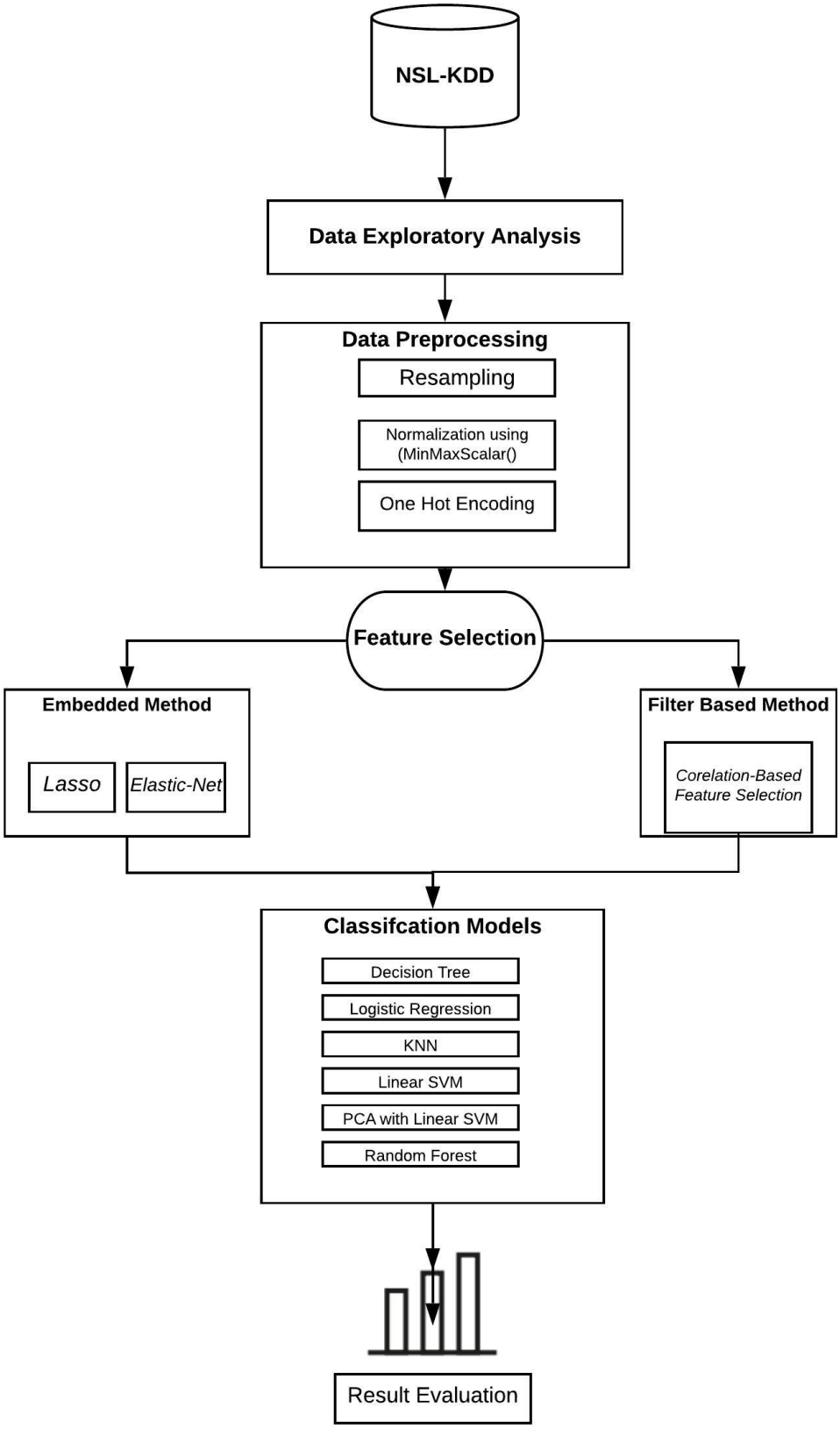


Figure 1: Complete Architecture of Implementation with NSL-KDD

3.1 Dataset Selection

For our research study, the dataset is taken form public repository [1].In which NSL-KDD intrusion dataset has been used and it has been widely recognized by many researchers in analysing and experimenting the intrusion detection anomalies. The reason why we choose NSL-KDD dataset is that it is very huge and contains large variation in its collection. Among those *KDDTrain+.ARFF* is chosen for training and the *KDDTest+.ARFF* is used as the testing set for our experiment. Further KDDTrain+.ARFF has been separated as validation set (80 %) and test set (20 %). the The dataset contains 41 features which has been categorized as **basic**, **content**, and **traffic** features and 1 target feature which contains either the normal or different attack types (among 4 possible attacks). Table 1 shows the number of records falls per class in each Training and Testing Class.

Table 1: Samples in Training and Test Set

Set	Total	Normal	Probe	DoS	R2L	U2R
KDDTrain+	125973	67343	45927	11656	995	52
KDDTest+	22544	9711	7458	2421	2754	200

3.2 Types of Attacks in *NSL-KDD*

1. **DoS:** *Denial of Service* denies the service or making the network unavailable to the users.

2. **U2R:** *User to Root* is an attack in which the attacker tries to access the local machine with the root privileges and trying to obtain the administrative accesses.

3. **R2L:** *Remote to Local* attack type exploits when as *unauthorized* user who tries to send the data packets to the local machine and trying to access the data.

4. **Probe**: This attack mainly falls under the category when the attackers tries to gather information about the local or target systems for launching the attack in those system later.

Table 2: Attacks Categorization

DoS	Probe	U2R	R2L
teardrop	ipsweep	perl	ftp_write
pod	satan	loadmodule	phf
land	nmap	rootkit	guess_passwd
back	portsweep	buffer_overflow	warezmaster
neptune	saint	xterm	warezclient
smurf	mscan	ps	imap
mailbomb			spy
			multihop
			named
			snmpguess
			worm
			snmpgetattack
			xsnoop
			xlock
			sendmail

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

3.3 Exploratory Data Analysis

Once the data has been selected, is it essential to explore the data in any research study to gain insights before applying it to any machine learning algorithms. The basic features have been plotted against the class labels to explore and understand the trend in the data.

In the beside graph the very first basic feature *duration* has been plotted against the different class labels. It clearly implies that all the attack types happened at the very first *0th* second. As only in the 'normal' and 'U2R' attack types, the duration features have showed lot many variations it is not good practice to include this feature in training the model.

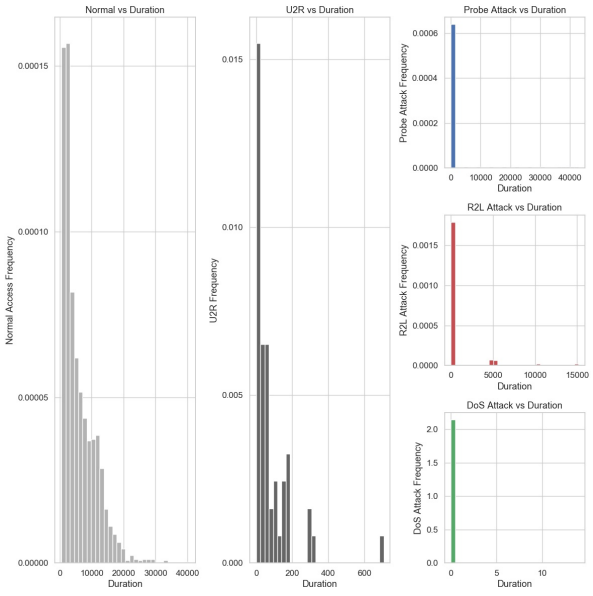
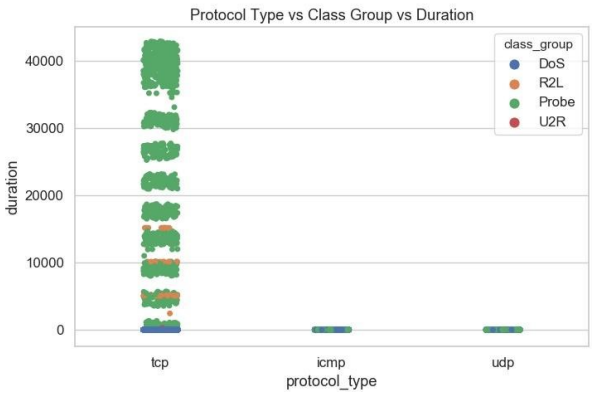
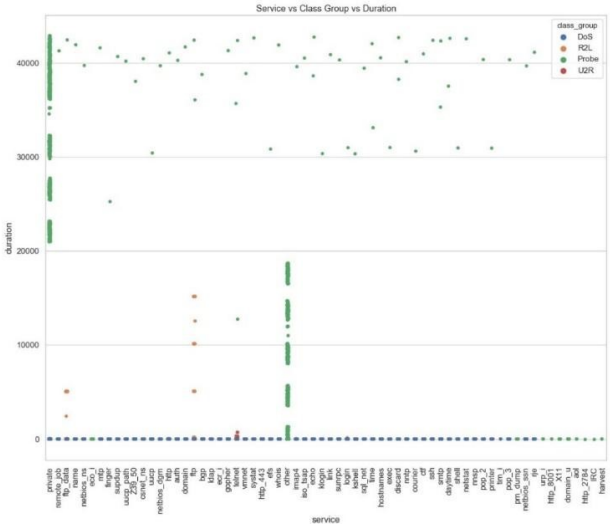


Figure 2: Duration vs Class

From the below figure 3a it is clearly understood that attackers had launched their attacks by using the TCP protype suite. Attackers found TCP suite as the most suitable and easy way to interface and launch their attacks with the target systems. And it is evident that only the probe type attack is happening during the various duration. The R2L attack types used the tcp protocol it also happens in the different seconds. The other attack types used different types of protocol type as their interface. Hence is this plot the protocol shows lot many variations we need to include this basic feature in our experiment study.



(a) Protocol Type vs Duration



(b) Service vs Duration

Figure 3: Basic Features

The service feature is been plot along with the duration feature against the class group

feature. From the figure 3b it is seen that the probe attack type is widely distributed against the all the service type. The probe attack type seems to mainly happen at private and other type of services. The R2R attack type is also distributed with the various services and it happened at the different time intervals. As considering service as one of the basic feature it contributes a lot in defining the class group and hence it can be considered as the strongest attribute to detect correct accuracy.

3.4 Dataset Pre-processing

1. It seems that NSL-KDD is highly imbalanced if we want to classify between 5 categories. In the given figure 4, count of U2R and R2L samples are very less.

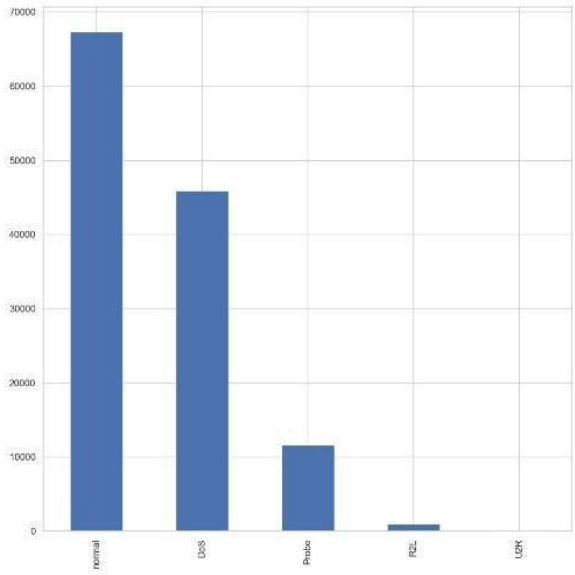


Figure 4: Distribution of Classes

2. Hence there are various techniques to balance the structured dataset like Synthetic Minority Oversampling Technique (Dong and Wang; 2016) for oversampling. But we will use simple resampling technique from sklearn.utils which will increase the count of U2R and R2L samples to match with Probe Samples count. Also, the count of normal samples is higher in the original NSL-KDD dataset, we can under sample the normal samples count.

3. There are some categorical attributes in the NSL-KDD dataset like [protocol_type, service, flag]. As these attributes are stored in text values it needs to be converted into numerical variables. The most common method to convert the string variable into categorical variables is Label Encoder. By converting the variable, it may contribute in giving the high accuracy of the machine learning models. Further, the five attacks under class_group attribute are processed and converted into integer form as 0 if it is DoS, 1 if it is Probe, 2 if it is R2L, 3 if it is U2R and 4 if it is normal.

4. The final step of data processing is to normalize/scale all continuous values under different columns between 0 and 1 by using *MinMaxScaler()*. Since we do not have validation set, we will consider 5% of data from training set into validation set and will apply our model on it. In this way the test set from KDDTest+.arff will remain untouched till the end. Table 3 shows the summary of data after resampling. The *MinMaxScaler()* can be given as in equation form as $X_{sc} = \frac{X - X_{min}}{X_{max} - X_{min}}$

Table 3: Imbalanced vs Balanced Dataset

Type	Normal	Probe	DoS	U2R	R2L
Imbalanced	67343	11656	45927	52	995
Balanced	11656	11656	45927	11656	11656

3.5 Feature Selection Technique

3.5.1 Embedded Methods

Feature selection is the process of selecting the most relevant and appropriate features from the original data set and using it in building the model. Embedded is one such method which are used to select the best accurate features by ranking the features based on regularization methods which bias the feature coefficients to zero. *Lasso* and *Elastic-Net* method have been used in our study to select the best features.

LASSO: A Stability Selection Technique which filter the top relevant features while making the other features nearing it to zero. Lasso performs a so called *L1 regularization* (a process of introducing additional information in order to prevent overfitting), i.e. adds penalty equivalent to absolute value of the magnitude of coefficients.

The residual sum of squares (RSS) is calculated as follows:

$$RSS = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

In particular, the minimization objective does not only include the residual sum of squares (RSS) - like in the OLS regression setting - but also the sum of the absolute value of coefficients.

The residual sum of squares (RSS) is calculated as follows:

$$RSS = \sum_{i=1}^{n} \left(y_i - \left(\beta_0 + \sum_{j=1}^{p} \beta_j x_{ij} \right) \right)^2$$

- n represents the number of distinct data points, or observations, in our sample.
- p denotes the number of variables that are available in the dataset.
- X_{ij} represents the value of the jth variable for the ith observation, where $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, p$.

In the lasso regression, the minimization objective becomes:

$$\sum_{i=1}^{n}\left(y_i - \left(\beta_0 + \sum_{j=1}^{p}\beta_j x_{ij}\right)\right)^2 + \lambda\sum_{j=1}^{p}|\beta_j|$$

which equals:

$$RSS + \lambda\sum_{j=1}^{p}|\beta_j|$$

λ(lambda) provides a trade-off between balancing RSS and magnitude of coefficients. λ can take various values:

- $\lambda = 0$: Same coefficients as simple linear regression.
- $\lambda = \infty$: All coefficients zero (same logic as before).
- $0 < \lambda < \infty$: coefficients between 0 and that of simple linear regression.

Elastic-Net: A mixed combination of LASSO and Ridge (*L2 Regularization*) which solves the limitations of those methods. It is been found that it outperforms the feature selection better than LASSO. It also makes the coefficients of the features to zero because of involvement of Ridge.

Table 4: Top 20 Ranked Features using LASSO and Elastic-Net

index	LASSO	Elastic-Net	Mean	Rank
protocol_type	1	1	1	1
wrong_fragment	0.94	0.94	0.94	2
same_srv_rate	0.8	0.82	0.81	3
dst_host_same_src_port_rate	0.73	0.77	0.75	4
logged_in	0.66	0.63	0.65	5
srv_serror_rate	0.68	0.46	0.57	6
diff_srv_rate	0.53	0.59	0.56	7
dst_host_serror_rate	0.26	0.38	0.32	8
root_shell	0.11	0.23	0.17	9
flag	0.15	0.14	0.15	10
num_file_creations	0.1	0.11	0.11	11
dst_host_srv_serror_rate	0	0.19	0.1	12
dst_host_rerror_rate	0	0.15	0.07	13
hot	0.03	0.03	0.03	14
num_compromised	0.01	0.03	0.02	15
num_failed_logins	0	0.04	0.02	16
num_root	0.01	0.03	0.02	17
dst_host_srv_count	0.01	0	0.01	18
service	0.01	0	0.01	19
dst_host_srv_diff_host_rate	0	0	0	20
dst_host_srv_rerror_rate	0	0	0	21
dst_host_diff_srv_rate	0	0	0	22
dst_host_same_srv_rate	0	0	0	23
dst_host_count	0	0	0	24

From the Table 4 that the first 20 features have some feature importance value for predicting the classification labels. Hence, we will select all first 20 features which has some score value likely *>= 0.01* and, hence, remaining features would not be considered.

3.5.2 Filter-based Correlation Method

Correlation based feature selection or Brute Force Method is based on below concept:

"A good feature subset is one that contains features highly correlated with (predictive of) the class, yet uncorrelated with (not predictive of) each other."

This method is used to find and grade the new feature subsets alternatively. The highly correlated features with a correlation coefficient value > 0.8 should not be considered for classification and such features can be discarded during classification. Correlation based feature selection help us to find the best subset of features from the training set which can help in increasing the performance of the machine learning models. Hence by implementing the filter-based method highly correlated features which can be dropped are as follows:

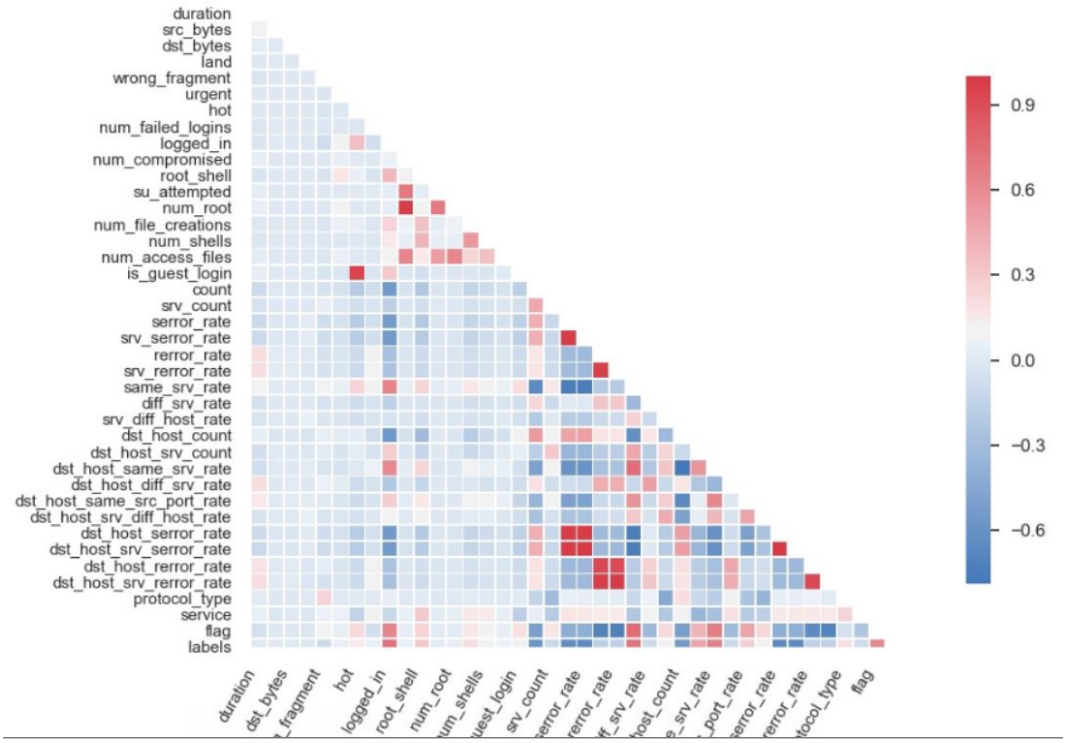


Figure 5: Pearson Correlation across different features from *NSL-KDD*

From Figure 5, few features like *dst_host_srv_serror_rate, dst_host_srv_rerror_rate* seems to be highly correlated and has coefficient value more than 0.9. Hence it is suggested to remove such highly correlated features. Likely, all highly correlated features can be dropped which has correlation coefficient value greater than 0.8. From the table 5 we can see the features which are dropped after applying filter method.

Table 5: Features which were dropped after applying Filter-based correlation

No	Dropped Features
1	dst_host_rerror_rate
2	dst_host_srv_serror_rate
3	dst_host_srv_rerror_rate
4	is_guest_login
5	dst_host_serror_rate
6	srv_serror_rate
7	num_root
8	srv_rerror_rate

4 Implementation of Different Models with 20 Selected Features

A 5 different classical models are applied on same dataset but only 20 features are considered. Rest of the features are truncated.

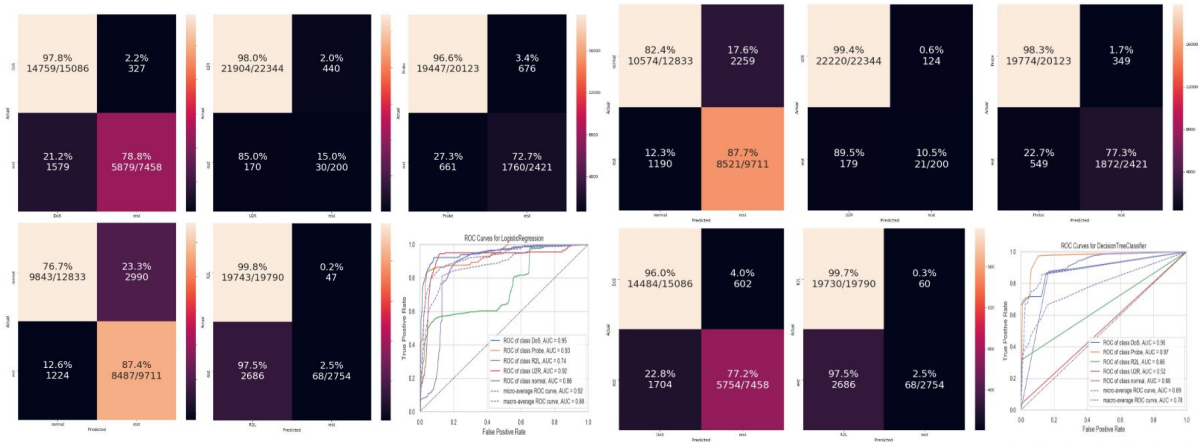
4.1 With Classical Models

Since the NSL-KDD dataset is highly imbalanced, each types of attack (DoS, Probe, Normal, U2R and R2L) is classified against the rest. So instead of multi-classification between 5 classes, we chose for binary classification like *One vs Rest*, and this type of classification is repeated further 4 times. In short, 5 classification with same model would be

- ***Normal vs Rest***,
- ***Probe vs Rest***,
- ***DoS vs Rest***,
- ***U2R vs Rest*** and
- ***R2L vs Rest***

4.1.1 Decision Tree and Logistic Regression

- ***With Decision Tree:*** Classification for Normal, Probe and DoS seems to be performing well. But the same model seems to be performing inaccurately for R2L and U2R type of attacks. Even ROC curve from figure 6a for U2R and R2L confirms the same.
- ***With Logistic Regression:*** From the figure 6b since the NSL-KDD is highly sparsed, hence we opted for simple Logistic Regression model, though the accuracy in each confusion matrix after applying this model shows improvement in performance but still it did not gave any breakthrough state-of-arts results.



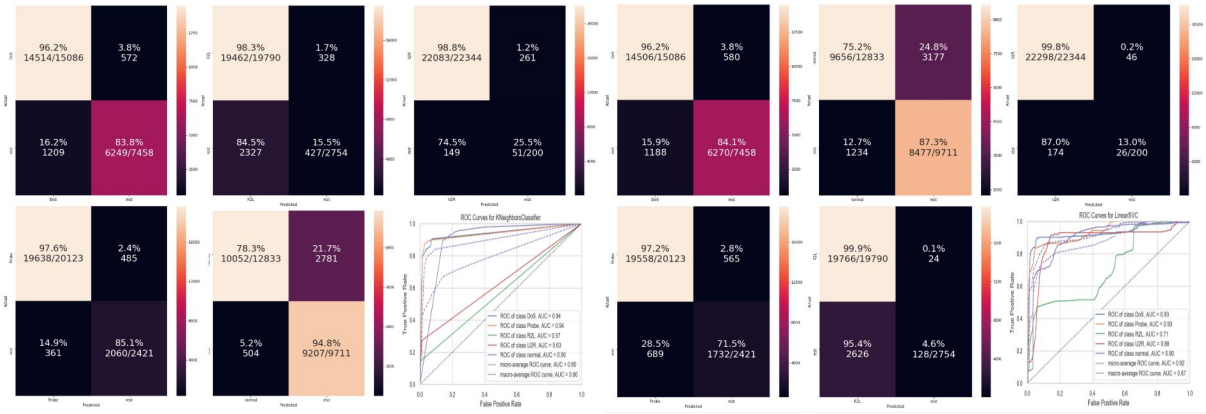
(a) Logistic Regression

(b) Decision Tree

Figure 6: Confusion matrix and ROC curve

4.1.2 K-Nearest Neighbours and Linear-SVM

- ***With K-Nearest Neighbours:*** This model out performs the previous classification models to classify between each type of attack vs rest. When we went for multi-class(5 types of attacks) classification with KNN, the overall performance achieved was *80.4%*. The respective ROC curves (from Figure 7a) for each type of attacks seems to be stable but still the model shows misbehaviour with U2R and R2L types of attack.



(a) KNN

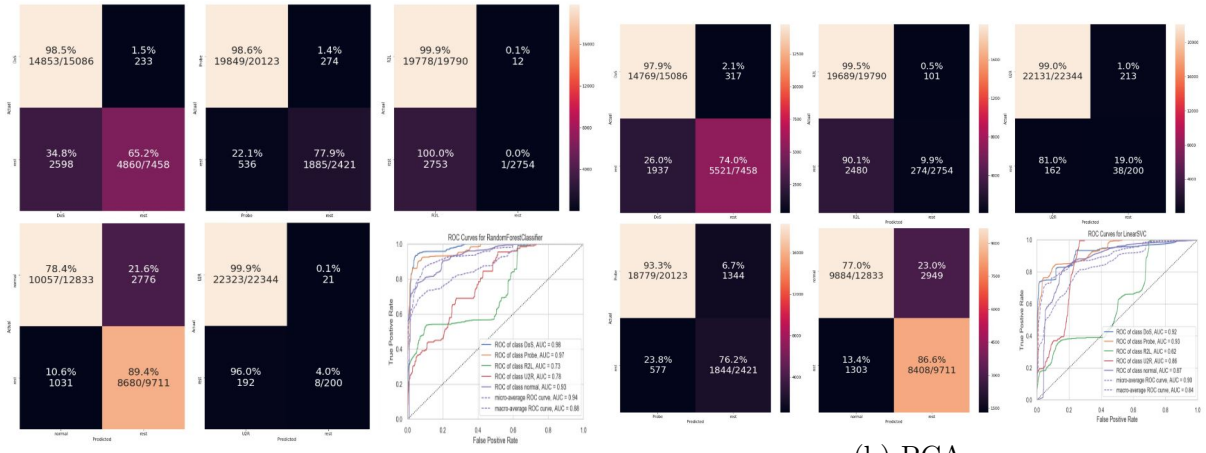
(b) Linear SVM

Figure 7: Confusion matrix and ROC curve

- ***With Linear-SVM:*** This model was suggested to represent non-linear data like *NSL-KDD* in higher dimensional space with strong support vectors. Though the initial approach was using Support Vector Classifier with certain RBF or Polynomial kernel approaches, but the model seems to be strongly under-performing with different parameter settings of *C* and *Gamma* values. Hence linear-SVM was chosen which almost gives similar results as previous Logistic Regression performance. But this performance was slightly improved with the application of **PCA**.

4.2 With Bagging Model and PCA Feature Transformation

- ***With Random Forest:*** An ensemble model was approached to classify the *One vs Rest* types of attacks. The number of estimators/trees considered was *n_ estimators=900*. The performance of this model was not that perfect for classification of attacks.



(a) Random Forest (b) PCA

Figure 8: Confusion matrix and ROC curve

- ***With Linear-SVM with PCA:*** Principal Component Analysis was approached to reduce the number of highly correlated features. The number of components considered was 10. Below figure 9 shows heat map of the 10 principal components on the *NSL-KDD* dataset.

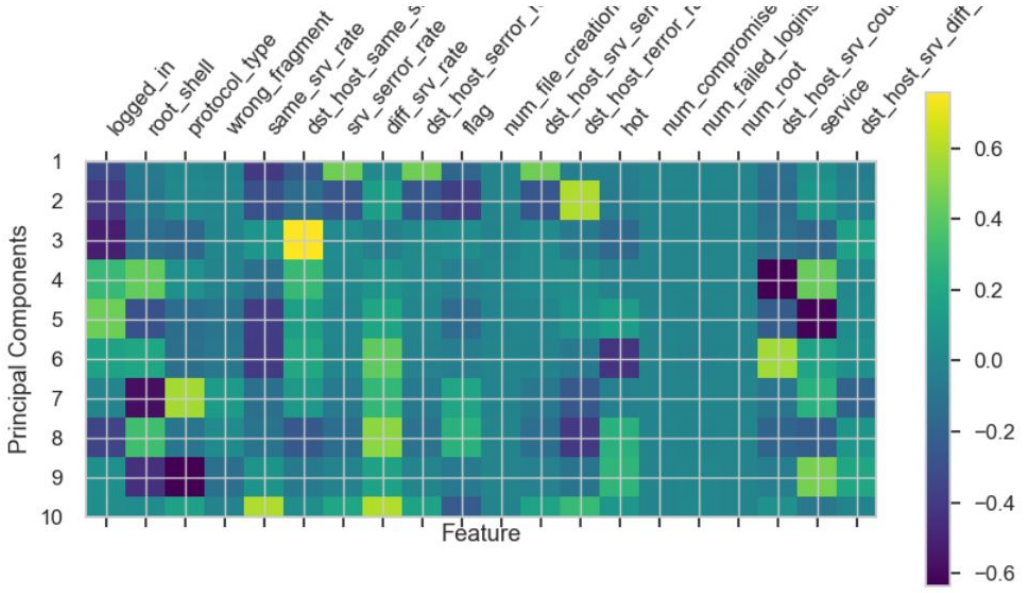


Figure 9: Pearson Correlation across different features from *NSL-KDD*

5 Results

5.1 Evaluation Performance with Embedded Method

Table 6: Evaluation results with Features selected from Embedded Techniques

Classification Algorithm	**Class Names**	**Test Accuracy with 20 features**	**Precision**	**f1-score**	***Mean Accuracy Score on Validation Set (One vs Rest)***	**Overall Stacked Accuracy/Ensemble Model Result**
Logistic Regression	DoS	92.18 %	82.94 %	92.0 %	96.47%	**90.67%**
	Probe	93.79 %	54.76 %	93.89 %		
	R2L	87.9 %	13.54 %	82.75 %		
	U2R	98.05 %	2.53 %	98.25 %		
	Normal	82.26 %	71.2 %	82.34 %		
	Average	**90.84 %**	**44.99 %**	**89.95 %**		
K-Nearest Neighbours	DoS	93.35 %	84.99 %	93.27 %	99.42 %	
	Probe	96.29 %	70.69 %	96.32 %		
	R2L	89.26 %	27.31 %	87.11 %		
	U2R	98.03 %	3.28 %	98.26 %		
	Normal	87.61 %	74.0 %	84.67 %		
	Average	**92.43 %**	**52.05 %**	**91.93 %**		
Decision Tree	DoS	89.89 %	77.62 %	89.67 %	99.24 %	
	Probe	95.0 %	59.27 %	94.76 %		
	R2L	87.63 %	12.43 %	82.24 %		
	U2R	99.0 %	3.46 %	98.74 %		
	Normal	84.33 %	74.12 %	84.4 %		
	Average	**91.17 %**	**45.38 %**	**89.96 %**		
Random Forest	DoS	87.3 %	73.45 %	86.55 %	99.19 %	
	Probe	96.04 %	67.31 %	95.9 %		
	R2L	87.79 %	12.41 %	82.19 %		
	U2R	99.04 %	1.82 %	98.69 %		
	Normal	83.52 %	72.8 %	83.59 %		
	Average	**90.74 %**	**45.56 %**	**89.38 %**		
Linear-SVM	DoS	92.69 %	83.78 %	92.57 %	96.82 %	
	Probe	94.07 %	57.62 %	94.25 %		
	R2L	88.9 %	20.4 %	85.07 %		
	U2R	96.39 %	1.67 %	97.36 %		
	Normal	80.73 %	69.24 %	80.8 %		
	Average	**90.56 %**	**46.54 %**	**90.01 %**		
Linear-SVM with PCA	DoS	91.47 %	80.67 %	91.38 %	95.58%	
	Probe	92.84 %	45.67 %	92.59 %		
	R2L	88.56 %	18.9 %	84.87 %		
	U2R	98.14 %	2.75 %	98.31 %		
	Normal	82.1 %	71.07 %	82.19 %		
	Average	**90.62 %**	**43.81 %**	**89.87 %**		

After selecting top 20 features using embedded techniques (*LASSO* and *Elastic-Net*), the best overall performance seems to be from **K-Nearest Neighbours** with $K = 30$. *k-NN* almost gives test accuracy of *92.43%* and validation accuracy of *99.42 %*. Though the average precision scores seems to be always lowest due to presence of anomalies like *R2L* and *U2R* samples in the *NSL-KDD* dataset with respect to all implemented classification algorithms, but samples under remaining classes tries to promote the average accuracy to more than *90 %*. The features reduction technique of *Principal Component Analysis with*

Linear-SVM almost gives similar performance results as that of standalone *Linear-SVM*. This makes sense since some of the features were highly correlated and PCA rotates the features in such a way that all correlated features becomes uncorrelated, and hence giving similar performance as that of *Linear-SVM* with all features.

At the end an ensemble model is applied which follows voting mechanism to choose the majority voted classes for the samples from test set. This ensemble model is the combination of all classification algorithms listed in first column of Table 6. The last

column from Table 6 *"Overall Stacked Accuracy/Ensemble Model Result"* is the stacked accuracy or ensemble accuracy obtained on test samples.

5.2 Evaluation Performance with Filter Based Method

After Implementing the Filter technique *(Correlation based feature selection)*, from the Table 7 it is seen that, again *K-Nearest Neighbours* outperforms well with this method with the highest accuracy of *91.76 %*. As there is a huge imbalance of data with respect to *R2L* and *U2R*.It is seen that the text accuracy seems to have the lowest among all the other class types.

The *Precision* score seems to be low for the individual class *R2L & U2R* across all the classification model as there is a huge imbalance of class records. As seen from the table due to the low precison score of the individual classes it affect the overall score of the classification model. The highest precision score is obtained with the *K-NN* classification with *48.75 %*.When comparing the *f1-score* the highest score obtained from K-NN model with *91.06 %*. It is seen that as *Decision Tree* gives the better accuracy in the filter technique when compared the same model with *Embedded method* with second accuracy of *90.88 %* The same ensemble model is applied with the filter technique with the test set where the overall accuracy obtained by the ensemble model is *90.45 %* . Hence overall when compared it can conclude that Embedded feature selection method outperforms well than filter method.

Table 7: Evaluation results with Features selected from Filter Based Technique

Classification Algorithms	**Class Names**	***Test Accuracy with Filter Based Feature Selection***	***Precision***	***f1 -Score***	***Mean Accuracy Score on Validation Set (One vs Rest)***	***Overall Stacked Accuracy/Ensemble Model Result***
Logistic Regression	Dos	92.05 %	82.46 %	91.89 %	**96.95 %**	**90.45 %**
	Probe	96.1 %	68. 8 %	96.09 %		
	R2L	87.72 %	12.52 %	82. 29 %		
	U2R	98.03 %	2.41 %	98. 24 %		
	Normal	78.87 %	67.13 %	78. 96 %		
	Average	**90. 55 %**	**46.66 %**	**89.49 %**		
K- Nearest Neighbours	Dos	93.67 %	85.71 %	93.59 %	**99.49 %**	
	Probe	85.14 %	60.57 %	94. 95 %		
	R2L	89.25 %	22.83 %	85.74 %		
	U2R	98.05 %	3.11 %	98.27 %		
	Normal	82.71 %	71.52 %	82.74 %		
	Average	**91. 76 %**	**48.75 %**	**91.06 %**		
Decision Tree	Dos	93.26 %	85.45 %	93.11 %	**99.21 %**	
	Probe	93.89 %	49.69 %	93.26 %		
	R2L	87.92 %	13.22 %	82.47 %		
	U2R	99.01 %	5.94 %	98.82 %		
	Normal	80.25 %	68. 72 %	80.33 %		
	Average	**90.88 %**	**44.6 %**	**89.6 %**		
RandomForest	Dos	91.55 %	81.91 %	91.3 %	**99.55 %**	
	Probe	94.5 %	54.61 %	94.03 %		
	R2L	87.78 %	12.24 %	82.09 %		
	U2R	99.06 %	1.64 %	98.69 %		
	Normal	84.29 %	73.68 %	84.26 %		
	Average	**91.44 %**	**44.82 %**	**90.09 %**		
Liner-SVM	Dos	90.12 %	78.7 %	89.8 &	**96.75 %**	
	Probe	93.11 %	56.42 %	93.59 %		
	R2L	88.22 %	15.47 %	83.36 %		
	U2R	97.37 %	1.81 %	97.88 %		
	Normal	84.71 %	75.58 %	84.66 %		
	Average	**90.71 %**	**45.6 %**	**89.96 %**		
Liner-SVM with PCA	Dos	87.78 %	73.1 %	87.48 %	**96.06 %**	
	Probe	96 %	67.86 %	95.96 %		
	R2L	88.23 %	16.48 %	84.01 %		
	U2R	98.93 %	6.25 %	98.78 %		
	Normal	78.76 %	67.07 %	78.86 %		
	Average	**89.94 %**	**46.15 %**	**89.02 %**		

5.3 Discussion

A two different feature selection approaches were used ie. Embedded Technique and Filter-based Selection Technique and it is found out that performance is better after applying learning models on feature selected using Embedded Technique. Also, there was a main problem of Imbalanced dataset where the count of *U2R* and *R2L* samples were lowest almost below 200. Hence tried to increase the samples count of *U2R* and *R2L* to match with total samples of *Probe*. A slight improvement in performance was observed with increased accuracy of almost 4% to 5% after increasing the samples count of *U2R* and *R2L*. The highest accuracy achieved was *80.4 %* with k-Nearest Neighbour on untouched *KDDTest+.arff*.

Also, the same sets of model without any feature selection techniques were applied on newest dataset **UNSW-NB15** (a comprehensive data set for network intrusion detection systems) `https://ieeexplore.ieee.org/abstract/document/7348942` and the performance was observed as below:

Table 8: Performance with UNSW-NB15 after applying similar classical models

Classification Algorithm	**Test Accuracy**	***Validation Accuracy***
Logistic Regression	83.11 %	94.63 %
K-Nearest Neighbours	98.92 %	99.58 %
Decision Tree	99.00 %	99.24 %
Random Forest	98.99 %	99.82 %
Linear-SVM	84.21 %	94.61 %
Linear-SVM with PCA	83.02 %	92.34%

6 Conclusion and Future Work

Feature selection is applied on the *NSL-KDD* dataset to reduce dimensionality and to remove redundant and irrelevant features. Because of Embedded feature selection the total count of features were brought down from 43 to 20. Also we applied samples oversampling technique to convert highly imbalanced dataset to a balanced set. The count of *U2R* and *R2L* were almost 500 to None, hence oversampling technique was required. We compared the performances using different classification algorithms and found *kNearest Neighbour* gives the most perfect accuracy on untouched test set. A similar implementation was then repeated but without feature selection technique on UNSW-NB15 dataset and the result from most of the classification algorithms seems to be more than *98 %* on test set. in the future research the classification model can be enhanced with consensus approach of the feature selection can be experimented to get the better improvement of accuracy.

Acknowledgement

I would like to convey my sincere gratitude to my professor Dr.Pierpaolo Dondio in completing this thesis project successfully. I would also like to thank my family and friends for their support.

References

Aggarwal, P. and Sharma, S. K. (2015). Analysis of kdd dataset attributes-class wise for intrusion detection, *Procedia Computer Science* **57**: 842–851.

Ahmad, M. and Aftab, S. (2017). Analyzing the performance of svm for polarity detection with different datasets, *International Journal of Modern Education and Computer Science* **9**(10): 29.

Aljawarneh, S., Aldwairi, M. and Yassein, M. B. (2018). Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model, *Journal of Computational Science* **25**: 152–160.

Ashfaq, R. A. R., Wang, X.-Z., Huang, J. Z., Abbas, H. and He, Y.-L. (2017). Fuzziness based semi-supervised learning approach for intrusion detection system, *Information Sciences* **378**: 484–497.

Bhuyan, M. H., Bhattacharyya, D. K. and Kalita, J. K. (2015). Towards generating real-life datasets for network intrusion detection., *IJ Network Security* **17**(6): 683–701.

Chaudhari, R. R. and Patil, S. P. (2017). Intrusion detection system: Classification, techniques and datasets to implement.

Dhanabal, L. and Shantharajah, S. (2015). A study on nsl-kdd dataset for intrusion detection system based on classification algorithms, *International Journal of Advanced Research in Computer and Communication Engineering* **4**(6): 446–452.

Dong, B. and Wang, X. (2016). Comparison deep learning method to traditional methods using for network intrusion detection, *2016 8th IEEE International Conference on Communication Software and Networks (ICCSN)*, IEEE, pp. 581–585.

Farnaaz, N. and Jabbar, M. (2016). Random forest modeling for network intrusion detection system, *Procedia Computer Science* **89**: 213–217.

Hashem, S. H. (2017). Denial of service intrusion detection system (ids) based on naïve bayes classifier using nsl kdd and kdd cup 99 datasets, *Al-Rafidain University College For Sciences* (40): 206–231.

Homoliak, I., Breitenbacher, D. and Hanacek, P. (2017). Convergence optimization of backpropagation artificial neural network used for dichotomous classification of intrusion detection dataset., *JCP* **12**(2): 143–155.

Hsu, C.-M., Hsieh, H.-Y., Prakosa, S. W., Azhari, M. Z. and Leu, J.-S. (2018). Using long-short-term memory based convolutional neural networks for network intrusion detection, *International Wireless Internet Conference*, Springer, pp. 86–94.

Hussain, J. and Lalmuanawma, S. (2016). Feature analysis, evaluation and comparisons of classification algorithms based on noisy intrusion dataset, *Procedia Computer Science* **92**: 188–198.

Ingre, B. and Yadav, A. (2015). Performance analysis of nsl-kdd dataset using ann, *2015 International Conference on Signal Processing and Communication Engineering Systems*, IEEE, pp. 92–96.

Jabez, J. and Muthukumar, B. (2015). Intrusion detection system (ids): anomaly detection using outlier detection approach, *Procedia Computer Science* **48**: 338–346.

Javaid, A., Niyaz, Q., Sun, W. and Alam, M. (2016). A deep learning approach for network intrusion detection system, *Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS)*, ICST (Institute for Computer Sciences, Social-Informatics and . . . , pp. 21–26.

Kevric, J., Jukic, S. and Subasi, A. (2017). An effective combining classifier approach using tree algorithms for network intrusion detection, *Neural Computing and Applications* **28**(1): 1051–1058.

Mulak, P. and Talhar, N. (2015). Analysis of distance measures using k-nearest neighbor algorithm on kdd dataset, *International Journal of Science and Research* **4**(7): 2101–2104.

Pajouh, H. H., Javidan, R., Khayami, R., Ali, D. and Choo, K.-K. R. (2016). A two-layer dimension reduction and two-tier classification model for anomaly-based intrusion detection in iot backbone networks, *IEEE Transactions on Emerging Topics in Computing* .

Parsaei, M. R., Rostami, S. M. and Javidan, R. (2016). A hybrid data mining approach for intrusion detection on imbalanced nsl-kdd dataset, *International Journal of Advanced Computer Science and Applications* **7**(6): 20–25.

Sharma, N. and Gaur, B. (2016). An approach for efficient intrusion detection for kdd dataset: a survey, *International Journal of Advanced Technology and Engineering Exploration* **3**(18): 72.

Shone, N., Ngoc, T. N., Phai, V. D. and Shi, Q. (2018). A deep learning approach to network intrusion detection, *IEEE Transactions on Emerging Topics in Computational Intelligence* **2**(1): 41–50.

Sultana, N., Chilamkurti, N., Peng, W. and Alhadad, R. (2019). Survey on sdn based network intrusion detection system using machine learning approaches, *Peer-to-Peer Networking and Applications* **12**(2): 493–501.

Tang, T. A., Mhamdi, L., McLernon, D., Zaidi, S. A. R. and Ghogho, M. (2016). Deep learning approach for network intrusion detection in software defined networking, *2016 International Conference on Wireless Networks and Mobile Communications (WINCOM)*, IEEE, pp. 258–263.

Yin, C., Zhu, Y., Fei, J. and He, X. (2017). A deep learning approach for intrusion detection using recurrent neural networks, *Ieee Access* **5**: 21954–21961.