Network Intrusion Classification with Feature Reduction

Md. Sirazul Munir (ID: 011142064)

Md. Shamsul Alam (ID: 011142083)

Irin Jahan (ID: 011141067)

Jannatul Ferdaous (ID: 011141059)

Department of Computer Science and Engineering

United International University

A thesis submitted for the degree of

BSc in Computer Science & Engineering

May 2019

Abstract

Nowadays, in data technology, data preservation has become a good issue. Computers and completely different security breaches are incessantly attacked by security threats. There are completely different malicious network based or host based attacks that are a massive threat to networks. For protecting computer and networks from attacks and threats, the intrusion detection system has been used that is signature based. An Intrusion detection system gathers needed data to perform analytical actions. So that, it could determine threats that would be generated from a system or organizations inner or outer atmosphere. A great amount of data that is worked by the intrusion detection system, takes varied inappropriate and unnecessary features that result in raised execution time and low detection rate. As a result, in intrusion detection, an undeniable role is played by feature selection. To cover up such aspect various literature were revealed by completely different profound authors. During this analysis, we've approached to select some important features based on some feature selection algorithm so that the computational cost, space complexity and intrusion detection time can be reduced. Our analysis over NSL-KDD data set shows that once feature reduction, except Naive Bayes classifier the accuracy of the foremost classifiers is sort of as same because the performance with none feature reduction.

Acknowledgements

We would like to thank our supervisor Dr. Dewan Md. Farid (Associate Professor, Dept. of Computer Science and Engineering) for his valuable direction, discussion, and support. We would like to thank our families, especially our parents.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

After the invention of computer systems, the escalate for diverse needs of networking and with networking came the idea of data sharing. The internet is now an undeniable medium of information exchange and sharing. Enormous information capacity, high-speed transmission, worldwide coverage, and interactivity are the features of the internet.

1.1 Motivation

In the era of global proliferation, information technology has evolved not to mention easy to access and evolution of hacking tools, important data needs to secure. Impose limits on the development of the network is a foremost factor[1]. Though firewalls may render alert administrator never gets any of it. That's why a detection system is needed. The anomaly occurs in a system or a network are automatically monitored by an intrusion detection system (IDS) which analyzes for the sign of security problem. Nowadays networking attacks have severely increased over the security. For that reason IDS have become a required part of the security framework of the most organization[2]. There are three main functions in IDS which are the monitor, detect and respond to unauthorized activity by any organization's insider and outsider intrusion.

1.1.1 Networking Threats

Privacy is the most important thing in this modern era. Every organization wants to keep their data secure. But it is not possible to develop an absolutely secure system. Because most of the existing system has security flaws and there are many options for insiders to

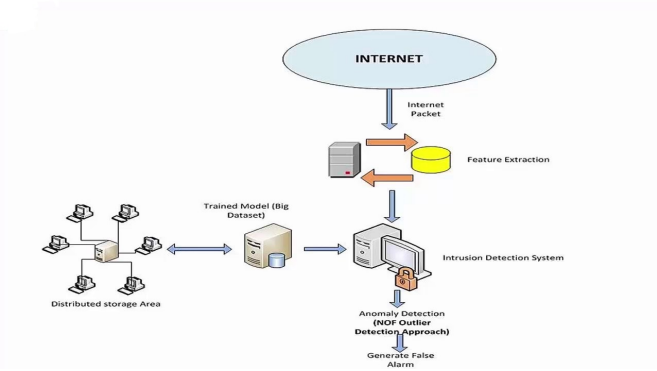

Figure 1.1: Intrusion Detection System.

abuse the system. There is also a big problem for big data. As big data requires huge memory space and huge time to execute, quick detection of intruders is most important to keep the damage in the limit. At present, there are many threats to networks like,

- Most of the existing system has security flaws.
- Many options for insiders to abuse the system.
- Unauthorized access to the network or computer information resources.
- Vulnerability the integrity of information.
- Information leakage from the storage medium, stolen from transit and so on.

Intrusion forbidding completely rely on the identification proficiency of IDS. As network speed becoming faster day by day, IDS needs to be lightweight with high identification rates. As a result, numerous feature selection methods are proposed. Filter, wrapper, and hybrid approach are three broad categories of selecting feature.[3]

1.2 Objectives of the Thesis

- In this research, we have searched several feature selection methods and intrusion detection systems which have been proposed for last 6 years. We have reviewed almost 40 research paper and journal.
- This thesis described many feature selection algorithms and search methods for how feature reduction occurred.

- Finally, this thesis showed the performance analysis comparison between feature reduction and without feature reduction in traditional classifiers.

1.3 Organization of the Thesis

Chapter 2 provides background and literature review of intrusion detection system and tradition classifiers.

Chapter 3 presents the material and methodology of feature reduction process

Chapter 4 discusses the experimental analysis using tables and line charts.

Chapter 5 presents the conclusions and discusses the future works.

Chapter 2

Background

This chapter provides a brief description of previous work about intrusion detection system, feature selection and a comprehensive review of numerous most recent methods for feature selection. Also, analyze numerous classification algorithm.

2.1 Intrusion Detection System

Intrusion means an action of intruding the act of wrongfully entering upon, seizing or taking possession of others property[4]. In Intrusion Detection, we detect those intruding actions through some detection systems. To protect a device or software application from suspicious activity or issues alerts when such activity happened which monitoring network traffic or systems called an intrusion detection system.

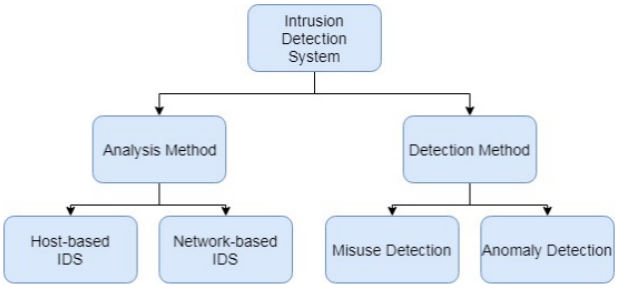

Figure 2.1: Classification of Intrusion Detection System.

2.1.1 Analysis Method

- **Host-Based IDS**: Host intrusion detection systems (HIDS) revolve around explicit owner or contrivance on the network. Individual hosting is it will bog down the computer. Just like actively scanning anti-virus computer code slows down alternative operations, IDS computer code will persist with individual machines. The incoming and outgoing packets from the contrivance are audited by a HIDS only and frequently the results is a comparison with a pre-generated image of the owner and therefore the owner's expected packet flow. If suspicious activity is detected, the user or administrator are alerted. From existing system files, it takes a snapshot and matches it to the previous one. each within the logical details of the owner likewise because the owner's activity, looking out for malevolent changes are completed by this approach. It often relies on a local applicant or operator of the IDS system so it might be entrenched on the host. Host-based IDSs have advantages that they will work with high individual knowledge that's systematically terribly descriptive. However, host-based IDSs will considerably have an effect on the machine's performance they're running on, depending upon the process dead.

- **Network-Based IDS**: With the spontaneous developing fame of the internet, there are an increasing variety of attacks supposed to the network itself that can't be expeditiously exposed by analyzing the owner audit route alone. Evolution of explicit tools has prompt that trying to find network attacks. Network intrusion detection systems (NIDS) checks and evaluate network traffic to secure a system from network-based intimidation. A NIDS learns all incoming packets and inquiries for any unsure patterns. once risks are detected, supported its harshness, actions will be taken by the system through giving barricade the IP address from attaining the network or notifying directors or administrators. completely different malicious activities like denial of service attacks and port scans and attacks by observance the network traffic will be detected by NIDS.

2.1.2 Detection Method

- **Misuse Detection**: In misuse detection, human experts offer system susceptibility and wide wisdom of common attacks. The procedure of misuse detection is employed

to detect PC and different system attacks. There are variations between misuse detection and anomaly detection . In misuse detection,at first abnormal system behavior is outlined then normal is going to be defined all different behavior . On the opposite hand, an anomaly detection, normal system behavior is outlined initial then all different behavior is defined as abnormal behavior. With misuse detection, not knowing something is normal . Attempting to find intruders who want to utilize famous vulnerabilities and for that reason they struggle to act these attacks. Though known attacks is acceptable for misuse detection, it cannot observe anonymous and outgoing processed risks. In the trendy creation of economic intrusion detection systems (IDSs) misuse detection is that the most usual access and that we will classify these approaches into four elements - (i) signature-based ways, (ii) rule-based techniques, (iii) ways supported state-transition analysis, and (iv) data processing primarily based techniques.

- **Anomaly Detection**: The system that monitors activity by classifying it within the normal or abnormal category and detects intrusion from both computer intrusion and misuse is called Anomaly detection[5]. Anomaly-based IDSes ordinarily defines normal and abnormal behavior[6]. Usually it takes a type of normal traffic and actions that take place on the network. The abnormal patterns that are sometimes not found within the traffic are measured against these baselines at the current state. Once we are looking out to spot new attacks or attacks that are severally made to bypass IDSs, this approach will operate terribly powerfully. Anomaly detection based IDS will detect attacks indications while not process attack models, however this model is extremely sensitive to the false alarm[7]. Recognizing sudden attacks dynamically is that the major blessings of the anomaly detection method. Anomaly detection ways are often classified into 5 categories: (i) applied mathematics methods, (ii) rule-based ways, (iii) distance-based ways, (iv) identification ways, and (v) model-based approaches.

2.2 Single Classifier

It is a procedure for evaluating the performance of an individually trained classifier.

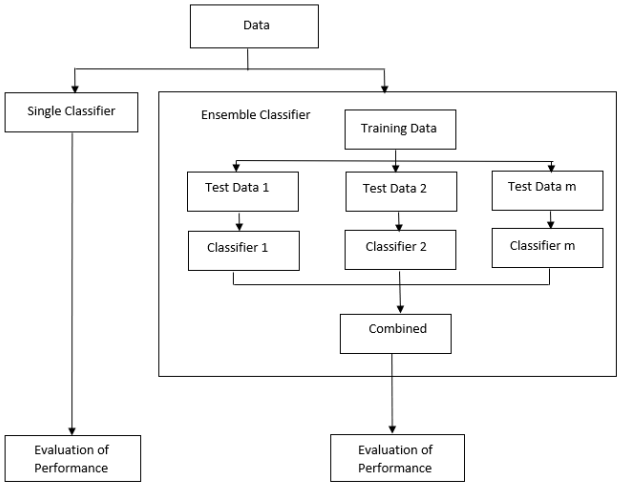

Figure 2.2: Single and Ensemble Classifier.

2.2.1 Naiive Bayes

A simple probabilistic machine learning model that commonly used for classification task which classifies data using the maximum posterior decision rule based on Bayesian setting is the Naiive Bayes classifier[8]. It is very productive and additionally can be represented using a simple Bayesian network. The assumption of the features is independent here. In this method at first, classifier finds the prior probability of each class from the training data set. Then find the class conditional probabilities of each attribute value from each attribute. Then finally find the posterior probability for each instance of a data set. It has been very popular and important for pattern classifications and also applied for solving real-life classification because of its high classification performance[9]. It also handles missing value easily.

2.2.2 J48

The Decision Tree (DT) belongs to supervised learning algorithms, used for solving both classifications and regression problems. DT is used to create a training model which can predict class or values of target attributes by learning decision rules. In a decision tree, each internal node of the tree compares to features and each leaf node corresponds to a class label[10]. In Decision Tree algorithms, at first, putting the best feature of the data set at the root of the tree. Secondly, dividing the training data set into subsets. Subsets should be in

that way where each subset have data with the same value for a feature. Until finding leaf nodes in all the branches of the tree, repeat step 1 and step 2 on each subset. In decision tree algorithms, We start from the root of the tree for predicting a class label for a data-set. We analyze the values of the root attributes with data's attribute. On the basis of analogy, we follow the branch comparable to that value and jump to the next node. Until we reach a leaf node with predicted class value, we continue comparing data's attribute values with other internal nodes of the tree. The J48 algorithm creates a decision tree using the highest information gain which can be chosen splitting features from the data-sets. The amount of information combined with a feature value is associated with the probability of incidence. To measure the amount of randomness from a data-set, entropy is used. When all data in a set exist to a single class, entropy is zero, no ambiguity is there. The objective of DT classification is to iterative division the given data-set into subsets where all components in every concluding subset exist to the same class[11].

2.2.3 CART

An alternative decision tree building algorithm is called CART. Classification and regression both tasks are easily handle by it. This algorithm applies a new metric named Gini index to generate decision points for classification assignments[12]. It stores the sum of squared probabilities of each class. It can be formulated as illustrated below.

$$Gini = 1 - sum(Pi)^2 \tag{2.1}$$

for i=1 to number of classes

The CART algorithm works to find the self-governing variable that produces the best homogeneous group when breaking the data. For a classification problem where the response variable is categorical, this is measured by calculating the information gained based upon the entropy resulting from the split. For the numeric response, homogeneity is decided by statistics such as standard deviation. Two relevant parameters of the CART procedure are the minimum split criterion and the complexity parameter (Cp). The minimum split criterion is the least number of records that must be present in a node before a split can be ventured. This has to be described at the outset. Cp is a complexity parameter which avoids

splitting those nodes that are obviously not helpful. Another way to examine these parameters is that the Cp value is defined after "growing the tree" and the optimal value is applied to "prune the tree."

2.2.4 Support Vector Machine

For regression and classification problems support vector machine is a linear model. It is used for both linear and non-linear problems. It looks at the extremes of the data sets and draws a decision boundary which is known as a hyper plane near the extreme points in the data set. So support vector machine algorithm is a frontier which best segregates the two classes. This algorithm implies that only support vector is important whereas other training examples are disregard able. Suppose we have a data set where we need to classify the red rectangles from blue ellipses. There are two lines (green line and yellow line) for separating the data.

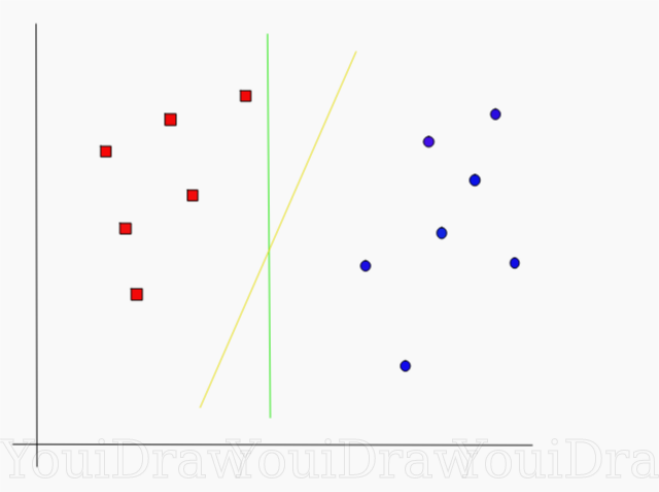

Figure 2.3: Finding the Best Lines Separating Data.

Here, the green line separates the two class but it is very close to the red class. That's why the green line is not a hypothesize line because our target is to find more hypothesize divider. On the other hand, it is quite spontaneous that yellow lines divide the two classes better. But to adjust the line, we have to do something specific. As stated in the SVM algorithm, support vectors are the points from both classes which are closest to the lines[13]. We have to find the gap between the support vectors and the lines. This gap is called margin

and our objective is to maximize the margin. Any hyper-plane will be optimal when it has a maximum margin.

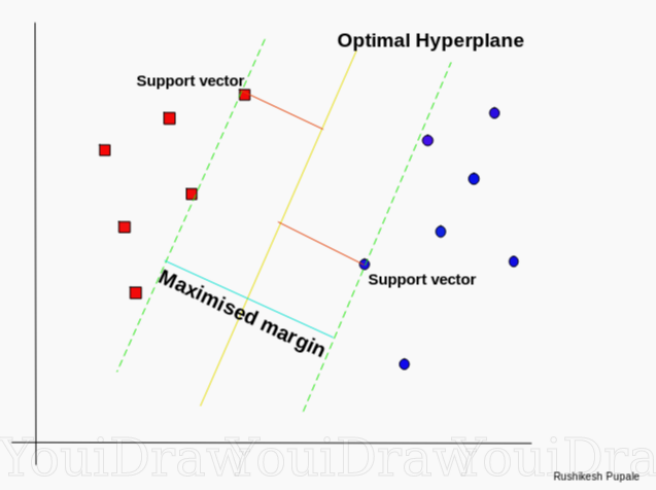

Figure 2.4: Lines Separating Data.

Thus SVM makes a decision boundary in such a way that the separation between the two classes is as wide as possible[14].

2.3 Ensemble Classifier

A procedure of merging non-identical trained classifiers to build a powerful made up model whose predictions are evaluated to provide a single output.

2.3.1 Random Forest

Random Forest could be a technique that operates by constructing multiple trees during training phases. The decision of the majority of the trees is chosen by the random forest as the final decision. For each classification and regression issues, that create the most votes of current machine learning systems, it will be used. Two steps happen within the Random Forest rule, First, making the random forest, second is to make a prediction from the random forest classifier creating in the 1st step. At first, choosing "p" features from total "m" features randomly where p ł m. Secondly, using the best split point, calculate the node "d" among the "p" features. Then, child nodes are generated by splitting the node using the best split. Until "l" number of nodes reach, the 1st to 3rd steps have been repeated. To create the

"n" number of trees, repeat steps 1st to 4th for "n" number times and thus build a forest. Within the next stage, the forecast will be marked using the random forest classifier created. Firstly, the test features have taken and to predict the outcome, the rules of each randomly created decision have used and the predicted outcome has stored. Then, the votes for every predicted object have been calculated. Finally, from the random forest algorithm, the final prediction has been considered from the high voted predicted object.

2.3.2 Bagging

Bagging is an ensemble sampling technique. It stands for bootstrap aggregation. This algorithm invented by Bremen. In bagging, at first, we create a number of subsets of the data. Suppose there is m number of bags which represent data. Each one of this bag is a subset of original data. For each subset, we collect data randomly with replacement[15]. So any data can be repeated in any bags multiple times. Suppose a standard training set D of size n, Multiple Bootstrap sets, each sets contains n numbers instances which are derived out from original training set using randomly with replacement. Now we use each of these sets of data to train the different model. So we get m different models for m numbers of bags where each one trained on a little bit of different data. Now we query each model with the same test data and collect all models outputs[16]. Let we take the output of each model and take mean of them for determining the final result.

2.3.3 Boosting

Boosting is associate ensemble technique to convert a weak classifier to a powerful classifier. the most plan is that it learns from its previous mistakes and improves its performance, therefore, it uses multiple classifiers and every classifier provides a lot of priority to the previous one's flaws. In boosting, initially giving the same weight to all or any instances of the data set. Then choose "k" instances from the data set randomly to form the subset. Using the subset, build a model and test all the instances of the data set. Then increase the weight of these instances that are unclassified and reduces the weight of those instances that are properly classified. Again choose "m" instances randomly however now ensure that high weighted instances get a lot of priority. Then again build set of the data set and build a model using that subset and test all the instances. Thus the quantity of unclassified instances reduces and will increase properly classified instances.

2.3.4 Adaptive-Boosting

Adaptive Boosting is a supervised ensemble learning algorithm which is classify the models based on a boosting mechanism[17]. Initially, weights are given to all the points in training data set and weights of all the points are the same. Then randomly select The training set points and given priority of those which are the highest weighted.Those training set points are dropped which error is greater than 50% and the training set is again selected. On contrary, accepted model is assigned with a weight based on the accuracy of the model. After getting the final model set when a new instance appears, all the models vote for the new instance. The class or label with the maximum votes is assigned to the new instance.Thus, the Adaptive boosting mechanism works.

Chapter 3

Methodology

3.1 Feature Selection

In machine learning, feature selection is a vital issue which call for classify an appropriate set of features from which can build a classification model perfectly. Feature selection is a procedure for eliminating unrelated and superfluous features which will help to increase the predictive accuracy of classifiers[18]. The Intrusion detection system handles a big amount of data which enclose enormous unrelated and superfluous features resulting in expanded processing time and low detection rate[19]. Therefore feature selection plays a vital role in intrusion detection. Here we used some attribute evaluator for feature selection. Basically, Attribute evaluator is used for ranking all the features based on some metric[20]. Used three feature selection algorithms:

a) Correlation based feature subset selection evaluation (CfsSubSetEval)
b) Consistency based subset evaluation (consistancySubSetEval)
c) Classifier Attribute evaluation (ClassAttributeEval)

3.1.1 CfsSubSetEval

CfsSubSetEval is a simple filter algorithm[3]. It assesses the worth of a subset of attributes by appraising the respective ability of each feature along with the degree of redundancy between them. It ranks the feature subsets conforming to a correlation based heuristic evaluation function. The evaluation function is biased to those subsets that encompass features

that are strongly correlated with the class and uncorrelated with each other. Irrelevant features must be disregard because of the low correlation with the class. Redundant features are strongly correlated with one or more of the resting features, so they should be removed. The approval of a feature will rely on the latitude to which it predicts only those classes in areas of the instance which are not already predicted by other features[21].

$$M_s = \frac{k\bar{r_{ef}}}{\sqrt{k + k(k-1)\bar{r_{ff}}}} \tag{3.1}$$

where MS is the heuristic "merit" of a feature subset S encompassing k features, rcf is the mean feature-class correlation (f S), and rff is the average feature-feature inter-correlation. The numerator of Equation can be a concern of as arranging a hint of how predictive of the class a set of features are; the denominator of how much redundancy there is among the features. This Equation forms the core of CFS and put in a ranking on feature subsets in the search space of all possible feature subsets. There are some valid option of CfsSubSetEval:

- Treat missing values as a separate value.
- Do not add locally predictive attributes.
- Pre-calculate the full correlation matrix at the outset, rather than calculate correlations slowly during the search. Use this in conjunction with parallel processing according to speed up a backward search.

3.1.2 ConsistencySubSetEval

ConsistencySubSetEval appraises the consistency of a subset attributes in the class values when the training instances start to build the subset of attributes. The consistency of any subset must be higher the full set of attributes. So that the typical implementation is to use this subset evaluator in conjunction with a Random search which queries for the smallest subset with consistency equal to that of the full set of attributes[22]. Liu setiono (1996)-[23] has proposed a probabilistic approach for the selection of feature which evaluates the worth of a subset of attributes by the level of consistency in the class value. For these evaluation methods, consistancy metrics given by [24]

$$Consistency_s = 1 - \frac{\sum_{i=0}^{j} |D_i| - |M_i|}{N} \tag{3.2}$$

where s denotes an attribute subset, j indicates the number of distinct combinations of attribute values for s. D indicates the number of appearances of ith attribute value and combination and M indicates the cardinally of the majority class for jth attribute value combination and N denotes a total number of instances in the data set(Hail Holmes, 2003). The consistency-based subset evaluation method produces a random subset, S from the feature subset space (N) in every round of the process. If the number of feature (C) included by S is less than the current best subset, the inconsistency rate of data formed in S is checked against the inconsistency rate of the current best subset. If S has more or equal consistency with the best subset, then the best subset replaced by S.

Algorithm 1 ConsistencySubSetEval

Input: MAX-ITER, maximum number of iteration, L: data set, Q: number of attributes, x: allowed instability rate.
Output: Sets of O features satisfying the selection benchmark.

```
A = Q;
for i = 1 to MAX-ITER
F = random Set(seed);
B = number_of_features(F);
if(B < A)
If(Instability Check(F, L)<x)
G = F; A = B
print G
else if((B=A) and (Instability Check(F, L)<x))
print G
end for
```

3.1.3 ClassifierAttibuteEval

It is a abstract of ClassAttributeEval. Employing a user fixed classifier, the worth of an attribute is evaluated by this rule. The best way to evaluate individual attributes is to use ClassifierAttributeEval with a suitably chosen classifier. There are some valid option for classattibuteEval:

- By measuring the impact of leaving from the total set rather than considering its significance in isolation, the value of a subset of attributes is evaluated by it.

- Here, Seed is used for randomly generating fold splits.
- The number of attributes to evaluate in parallel.
- There is used a base learner which class name used for estimate the accuracy and number of fold folds to use for estimating accuracy

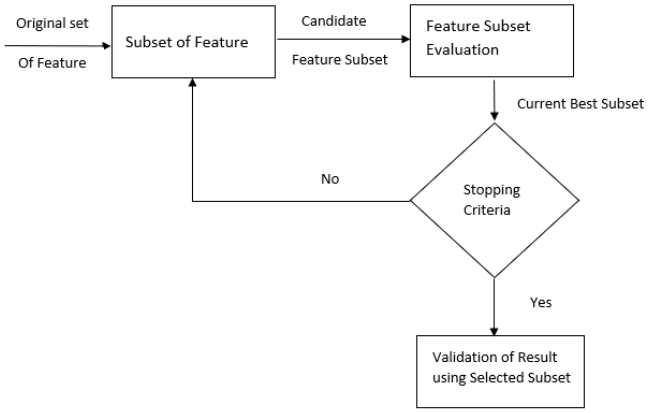

Figure 3.1: Feature Selection Process.

3.2 Feature Selection Process

In an exceedingly classic feature selection technique, four main steps are enforced by the feature selection method. First, it generates the subsequent candidate set of a feature from the total original set of the features, the second one is feature set analysis operate to seek out the candidate feature of the subset , stopping criteria is that the third one that decides once to prevent and also the last one is the accepted technique to justify in case the set is correct[25].

3.3 Search Method

For locating out the most effective set of attributes, there are some search strategies out there in WEKA. So as to seek out the most effective set of features, these strategies inquiry the set of all out their features. Best first, Greedy Step wise, and Ranker are 3 search technique that is employed during this work for correlation reason.

3.3.1 Best First Search

This searches the space of attribute subsets by greedy hill climbing augmented with a backtracking skill. There are some ensuing non-developing nodes decide to supervises the hole process of level of backtracking[25]. Best first may take the full set of attributes and start to search backward, or start with the empty set of attributes and search forward, or start at any point of the full set and search in forward or backward both directions (by acknowledging all possible single attribute additions and deletions at a given point).

3.3.2 Greedy Step Wise Search

A greedy forward or backward search through the area of feature subsets is performed within the greedy step wise search technique. It either begins with a blank set of attributes or a full set of attributes or from a random position within the area and breaks once the inclusion/cancellation of any resting features outcomes in an exceedingly reduction in the assessment[25]. it's going to give a rated list of features by passing through area from one aspect to the opposite and recount the form that features are elected.

3.3.3 Ranker search

By individual evaluations of features, this search technique ranks features and in-conjunction with attribute evaluators (Chi-square, Gain Ratio, Info Gain, etc) are used here.

3.4 Analysis Procedure

In this modern era, in every 90 second, there is a cyber attack occurred. So, detect those intrusions is the top-ranked priority for any organizations. That's why our main goal is to exclude some features with the help of the feature selection algorithm and reduce or at least remaining the initial accuracy of all traditional classifier. Here is our work procedure, First select the feature selection algorithm and search methods. Then used those algorithms to select the important features based on their own criteria. After feature selection, different classifiers applied for classification and finally, Accuracy is compared with the traditional classifiers without feature selection.

Chapter 4

Experimental Analysis

At first, we found the classification results of different classifiers without doing any feature reduction. Then we again find classification results of those classifiers we had used before. This time we did feature reduction using CfsSubSetEval, ConsistencySubSetEval, ClassAttributeEval feature Selection Algorithm respectively and compared all the results with the classification results without feature selection[26]. The comparison table is given below:

4.1 Data sets

In these analyses, we used NSL-KDD data set and the source of our data set is UCI machine learning data set. This data set is a waned version of the original KDD 99 data set. All features of NSL-KDD data set is the same features as KDD 99[27]. Some crucial problems within the KDD 99 data set that extremely affects the performance of the systems which leads to an awfully poor enumeration of anomaly detection approaches. To solve these problems, NSL-KDD data set is raised with elect records of the whole KDD data set. These are some advantages of using NSL-KDD data set:

- The classifier will not give biased result because of having no redundant data in the train set.
- There are no identical data with better reduction rates in the test set.
- The ratio of selected data from different cluster level is inversely proportional to the percentage of data in the main KDD data set.

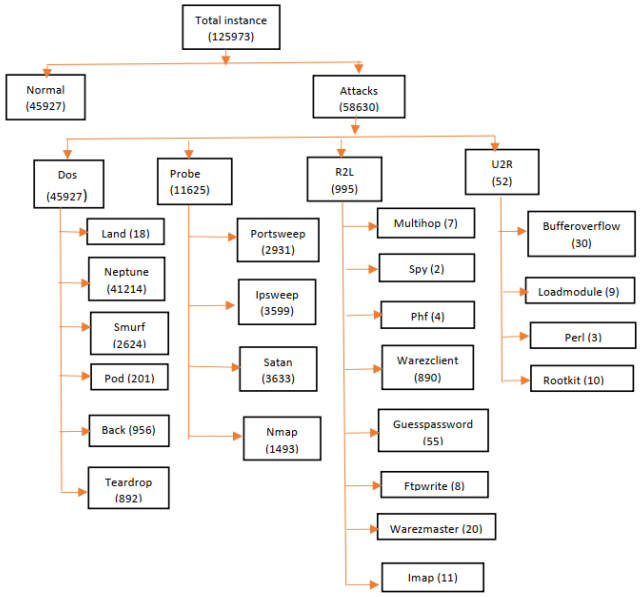

Figure 4.1: Data Set Details.

There are many versions available of this data set and used 20% of the training data for identified as KDDTrain+ 20Percent with a total number of 25192 instances. The test data set is recognized by the name KDDTest+ and has a total of 22544 instances. The total number of attributes in each case is 42[28]. The 42nd attribute is the "class" attribute. There are 4 types of Attacks in this data sets DoS, Probe, R2L and, U2R[29].

- DoS: Probe is any sort of attack that's organized consciously, therefore, it will detect its object and describe it with a big "fingerprint" within the report.
- Probe: Monitoring and other probing attacks. Its motive is to get information about the remote victim e.g. port scanning. Relevant features: "source bytes" and "duration of connection".
- U2R: Attackers have the regional association of sufferer machine and that they try to deliver the goods super user allowance.
- R2L: During this case, a hacker doesn't have an account on the victim machine. For that reason, they try to deliver goods access.

Table 4.1: Number of instances in data set.

Attack Types	Training Examples	Testing Examples
Normal	67343	13449
Denial of Service	45927	9234
Remote to User	995	209
User to Root	52	11
Probing	11656	2289
Total Examples	125973	25192

4.2 Experimental Results

Classification results of different classifiers using a feature selection algorithm and without using any feature selection algorithm are given below.

Table 4.2: Total number of features.

Feature selection algorithm	Number of features
CfsSubEval(Bestfirst)	19
CfsSubEval(GreedyStepwise)	10
ConsistancySubEval(Bestfirst)	13
ConsistancySubEval(Bestfirst)	14
ClassifierAttributeEval	25

After feature reduction, with the help of feature selection algorithms many features have removed from the original data-set. Important information may contain on those removed features from the data-set. This is a limitation. But to detect intrusions faster for less damage, this limitation has ignored.

Table 4.3: Selected feature number after using feature selection algorithm.

Feature selection algorithm	Feature Number
CfsSubEval(Bestfirst)	2,3,4,5,6,7,8,10,12,23,25,29,30,35,36,37,38,39,40
CfsSubEval(GreedyStepwise)	4,5,7,8,10,12,30,35,36,37
ConsistancySubEval(Bestfirst)	1,3,5,6,23,32,33,35,36,37,38,39,40
ConsistancySubEval(Bestfirst)	1,3,5,6,12,23,32,33,35,36,37,38,39,40
ClassifierAttributeEval	41,13,12,20,14,15,16,17,18,11,10,9,4,2,3,5,8,6,7,19,21,40,35,33,22

Table 4.4: Classification result without feature reduction.

Classifier	Accuracy	Precision (Weighted avg.)	F-Measure (Weighted avg.)	Recall (Weighted avg.)
Naiive Bayes	27.5194	0.675	0.311	0.275
J48	85.6816	0.827	0.813	0.857
CART	83.1595	0.819	0.797	0.832
SVM	82.3774	0.742	0.773	0.824
Random Forest	86.634	0.826	0.817	0.866
Bagging(Reptree)	82.2603	0.883	0.792	0.823
Adaboost(J48)	85.4847	0.821	0.808	0.855

Table 4.5: Classification result using CfsSubEval(Bestfirst).

Classifier	Accuracy	Precision (Weighted avg.)	F-Measure (Weighted avg.)	Recall (Weighted avg.)
Naiive Bayes	51.9794	0.871	0.571	0.520
J48	86.3946	0.772	0.813	0.864
CART	83.1276	0.815	0.795	0.831
SVM	81.8719	0.803	0.767	0.819
Random Forest	86.2829	0.769	0.811	0.863
Bagging(Reptree)	82.388	0.821	0.794	0.824
Adaboost(J48)	85.5645	0.765	0.805	0.856

Table 4.6: Classification result CfsSubEval(GreedyStepwise).

Classifier	Accuracy	Precision (Weighted avg.)	F-Measure (Weighted avg.)	Recall (Weighted avg.)
Naiive Bayes	59.4977	0.845	0.629	0.595
J48	84.1226	0.793	0.797	0.841
CART	84.4046	0.781	0.792	0.844
SVM	80.5257	0.735	0.755	0.805
Random Forest	84.7026	0.785	0.796	0.847
Bagging(Reptree)	84.7292	0.776	0.795	0.847
Adaboost(J48)	83.6756	0.785	0.789	0.837

Table 4.7: Classification result using ConsistancySubEval(Bestfirst).

Classifier	Accuracy	Precision (Weighted avg.)	F-Measure (Weighted avg.)	Recall (Weighted avg.)
Naiive Bayes	50.9152	0.804	0.568	0.509
J48	84.6334	0.805	0.798	0.846
CART	85.4741	0.820	0.808	0.855
SVM	81.2014	0.735	0.770	0.812
Random Forest	83.7501	0.826	0.804	0.838
Bagging	83.5054	0.823	0.802	0.835
Adaboost(J48)	85.0005	0.816	0.802	0.850

Table 4.8: Classification result using ConsistancySubEval(GreedyStepwise).

Classifier	Accuracy	Precision (Weighted avg.)	F-Measure (Weighted avg.)	Recall (Weighted avg.)
Naiive Bayes	53.8523	0.773	0.590	0.539
J48	84.5855	0.799	0.796	0.846
CART	85.4794	0.820	0.809	0.855
SVM	81.4302	0.728	0.767	0.814
Random Forest	83.7927	0.795	0.801	0.838
Bagging(Reptree)	83.5054	0.790	0.835	0.802
Adaboost(J48)	85.7933	0.808	0.809	0.858

Table 4.9: Classification result using ClassifierAttributeEval.

Classifier	Accuracy	Precision (Weighted avg.)	F-Measure (Weighted avg.)	Recall (Weighted avg.)
Naiive Bayes	43.4181	0.664	0.471	0.434
J48	84.1758	0.758	0.794	0.842
CART	84.5483	0.841	0.792	0.845
SVM	83.1116	0.800	0.777	0.831
Random Forest	85.5965	0.764	0.805	0.856
Bagging(Reptree)	82.9626	0.824	0.792	0.830
Adaboost(J48)	84.4525	0.757	0.793	0.845

Table 4.5 shows Naiive Bayes classifier performed much better with feature reduction than the accuracy without any feature reduction. Also the accuracy of J48, CART, and Bagging classifiers has increased.

Table 4.6 shows the accuracy of Naiive Bayes, CART, and Bagging has increased with feature reduction algorithm named CfsSubEval(GreedyStepwise) than the accuracy without any feature reduction.

Table 4.7 shows the accuracy of Naiive Bayes, CART, and Bagging has increased with feature reduction algorithm named ConsistancySubEval(Bestfirst) than the accuracy without any feature reduction.

Table 4.8 shows the accuracy of Naiive Bayes, CART, SVM and Adaboost has increased with feature reduction algorithm named ConsistancySubEval(GreedyStepwise) than the accuracy without any feature reduction.

Table 4.9 shows the accuracy of Naiive Bayes, CART, SVM and Adaboost has increased with feature reduction algorithm named ClassifierAttributeEval than the accuracy without any feature reduction.

After feature reduction, the accuracy of most classifiers has improved. The accuracy of some classifiers has decreased but the amount of decreasing accuracy is very low.

4.3 Comparison

Following line charts will give a clear comparison between the accuracy of different classifiers for using a feature selection algorithm and not for using any feature selection algorithm. Different classifiers have performed differently for different feature selection algorithm.

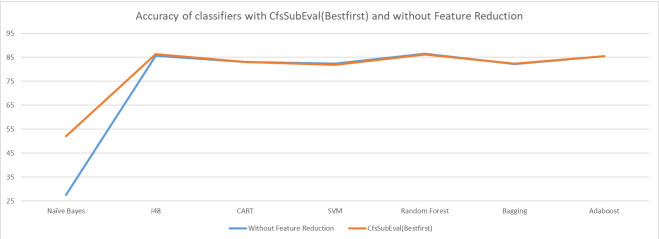

Figure 4.2: Accuracy of classifiers with CfsSubEval(Bestfirst) and without feature reduction.

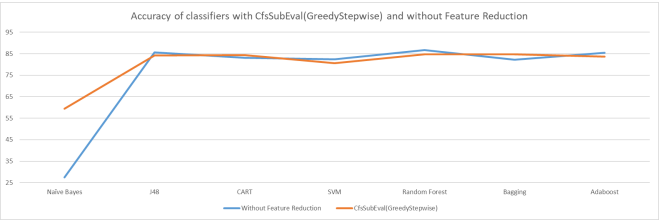

Figure 4.3: Accuracy of classifiers with CfsSubEval(GreedyStepwise) and without feature reduction.

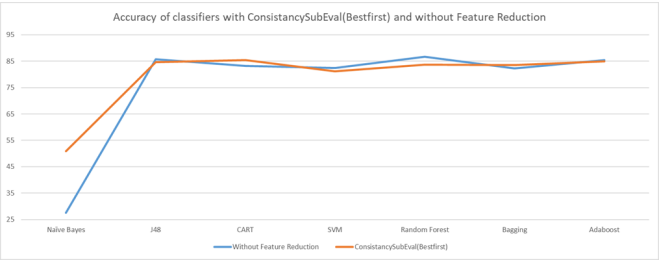

Figure 4.4: Accuracy of classifiers with ConsistencySubEval(Bestfirst) and without feature reduction.

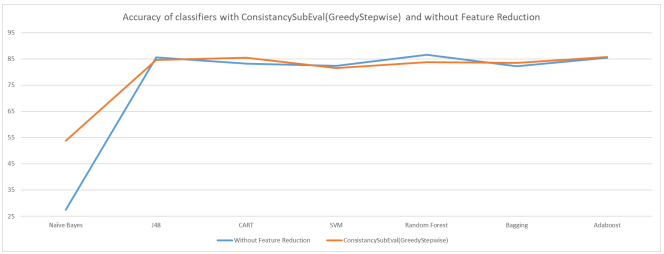

Figure 4.5: Accuracy of classifiers with ConsistencySubEval(GreedyStepwise) and without feature reduction.

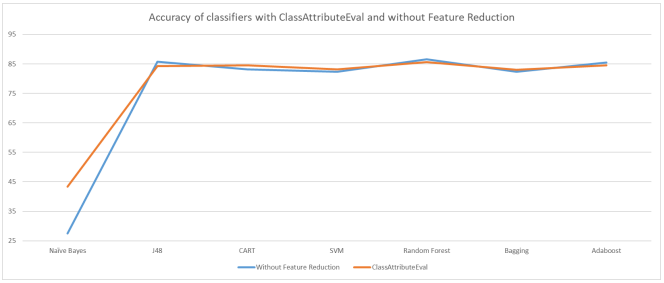

Figure 4.6: Accuracy of classifiers with ClassAttributeEval and without feature reduction.

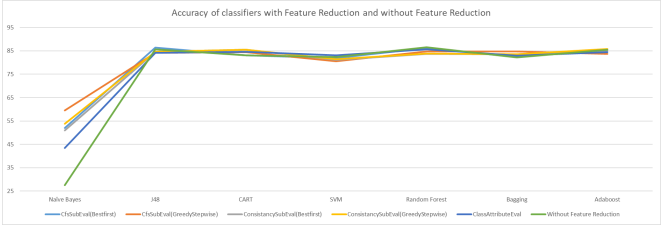

Figure 4.7: Accuracy of classifiers with feature reduction and without feature reduction.

4.4 Summary

This chapter provides a clear view of the performance of different classifiers. The performance of different classifiers shows that most of the classifiers performed well with feature reduction compared to without feature reduction. Some of the classifiers did not performed well with feature reduction. The accuracy of those classifiers are almost same as without feature reduction. Figure 4.7 clearly shows that except Naiive Bayes classifier, the performance of other classifiers with feature reduction is as same as without any feature reduction.

Chapter 5

Conclusions and Future Work

5.1 Conclusions

Several numbers of machine learning algorithms like single, hybrid and ensemble classifiers are applied to detect and prevent intrusions. These algorithms are very useful to detect intrusions but no methods can detect or prevent all types of intrusions. And till now, various sorts of intrusions have been detected but it is not possible to prevent all of them, as the system cannot be absolutely flawless. As we are working with big data and trying to detect intrusion, it takes too much time and space. After our analysis with some feature reduction algorithms, we have seen that the accuracy has improved from the previous accuracy or tends to be the same. If we can reduce some features and keep the accuracy higher or the same as before, we can detect intruders much faster and time and space complexity can be reduced.

5.2 Future Work

Future work will be improving previous accuracy and precision with the help of feature selection. And also try to extend our work to build an ensemble method which can easily detect attacks, consume less memory space, and can perform in less execution time.

Bibliography

[1] S. R. Snapp, J. Brentano, G. Dias, T. L. Goan, L. T. Heberlein, C.-L. Ho, and K. N. Levitt, "Dids (distributed intrusion detection system)-motivation, architecture, and an early prototype," *Proceedings of the 14th national computer security conference*, April 1991. 1

[2] C.-F. Tsai, Y.-F. Hsu, C.-Y. Lin, and W.-Y. Lin, "Intrusion detection by machine learning: A review," *expert systems with applications*, vol. 36, no. 10, pp. 11 994–12 000, 2009. 1

[3] M. A. Hall and L. A. Smith, "Feature subset selection: a correlation based filter approach," *International Conference on Neural Information Processing and Intelligent Information Systems (pp. 855-858), Berlin.*, JUNE 1997. 2, 13

[4] W.-C. Lin, S.-W. Ke, and C.-F. Tsai, "Cann: An intrusion detection system based on combining cluster centers and nearest neighbors," *Knowledge-based systems*, vol. 78, pp. 13–21, 2015. 4

[5] N. Chand, P. Mishra, C. R. Krishna, E. S. Pilli, and M. C. Govil, "A comparative analysis of svm and its stacking with other classification algorithm for intrusion detection," *2016 International Conference on Advances in Computing, Communication, & Automation (ICACCA)(Spring)*, pp. 1–6, 2016. 6

[6] C. Kruegel, D. Mutz, W. Robertson, and F. Valeur, "Bayesian event classification for intrusion detection," *19th Annual Computer Security Applications Conference, 2003. Proceedings.*, pp. 14–23, 2003. 6

[7] N. J. Miller, "Benchmarks for evaluating anomaly-based intrusion detection solutions," *Presented to the Department of Computer Engineering and Computer Science California State University, Long Beach*, 2018. 6

[8] M. Panda and M. R. Patra, "Network intrusion deteciton using naive bayes," *International Journal of Computer Science and Network Security(IJCSNS)*, vol. 7, pp. 258–263, December 2007. 7

[9] N. H. Dewan Md. Farid and M. Z. Rahman, "Combining naive bayes and decision tree for adaptive intrusion detection," *International Journal of Network Security Its Applications (IJNSA)*, pp. 12–25, April 2010. 7

[10] S. Peddabachigari, A. Abraham, C. Grosan, and J. Thomas, "Modeling intrusion detection system using hybrid intelligent systems," *Journal of network and computer applications*, vol. 30, no. 1, pp. 114–132, 2007. 7

[11] C. M. Rahman, D. M. Farid, N. Harbi, E. Bahri, and M. Z. Rahman, "Attacks classification in adaptive intrusion detection using decision tree," *International Conference on Computer Science (ICCS)*, 2010. 8

[12] A. A. Srilatha Chebrolu and J. P. Thomas, "Feature deduction and ensemble design of intrusion detection systems," *International Journal of Computers and Security*, vol. 24, pp. 295–307, 2005. 8

[13] W.-H. Chen, S.-H. Hsu, and H.-P. Shen, "Application of svm and ann for intrusion detection," *Computers & Operations Research*, vol. 32, no. 10, pp. 2617–2634, 2005. 9

[14] S. T. Miller and C. Busby-Earle, "Multi-perspective machine learning a classifier ensemble method for intrusion detection," *Proceedings of the 2017 International Conference on Machine Learning and Soft Computing*, pp. 7–12, 2017. 10

[15] D. Gaikwad and R. C. Thool, "Intrusion detection system using bagging ensemble method of machine learning," *2015 International Conference on Computing Communication Control and Automation*, pp. 291–295, 2015. 11

[16] M. J. R. Dewan Md. Farid and C. M. Rahman, "Adaptive intrusion detection based on boosting and naÃŕve bayesian classifier," *International Journal of Computer Applications (0975 âĂŞ 8887) Volume 24âĂŞ No.3*, June 2011. 11

[17] W. H. Weiming Hu and S. Maybank, "Adaboost-based algorithm for network intrusion detection," *IEEE Transactions on Systems, Man, and Cybernetics, Part B(Cybernetics)*, vol. 38, pp. 577–583, April 2008. 12

[18] X. H. Mohammad A. Ambusaidi and P. Nanda, "Unsupervised feature selection method for intrusion detection system," *2015 IEEE Trustcom/BigDataSE/ISPA, Findland*, August 2015. 13

[19] E. Emary, H. M. Zawbaa, C. Grosan, and A. E. Hassenian, "Feature subset selection approach by gray-wolf optimization," *Afro-European Conference for Industrial Advancement*, pp. 1–13, 2015. 13

[20] N. K. Akashdeep, Ishfaq Manzoor, "A feature reduced intrusion detection system using ann classifier," *Expert Systems With Applications*, vol. 88, pp. 249–257, 2017. 13

[21] H. M. A., "Correlation-based feature subset selection for machine learning," *This thesis is submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy at The University of Waikato.*, April 1999. 14

[22] A. Onan, "A fuzzy-rough nearest neighbor classifier combined with consistency-based subset evaluation and instance selection for automated diagnosis of breast cancer," *Expert Systems with Applications Volume 42, Issue 20,*, November 2015. 14

[23] H. Liu, R. Setiono *et al.*, "A probabilistic approach to feature selection-a filter solution," vol. 96, pp. 319–327, 1996. 14

[24] M. A. Hall and G. Holmes, "Benchmarking attribute selection techniques for discrete class data mining," 2002. 14

[25] A. Megha Aggarwal, "Performance analysis of different feature selection method in intrusion detection," *INTERNATIONAL JOURNAL OF SCIENTIFIC TECHNOLOGY RESEARCH VOLUME 2, ISSUE 6*, JUNE 2013. 16, 17

[26] N. F. Haq, A. R. Onik, M. A. K. Hridoy, M. Rafni, F. M. Shah, and D. M. Farid, "Application of machine learning approaches in intrusion detection system: a survey," *IJARAI-International Journal of Advanced Research in Artificial Intelligence*, vol. 4, no. 3, pp. 9–18, 2015. 18

[27] M. S. M. LAHEEB M. IBRAHIM, DUJAN T. BASHEER, "A comparison study for intrusion database (kdd99, nsl-kdd) based on self organization map (som) artificial neural network," *Journal of Engineering Science and Technology Vol. 8, No. 1 School of Engineering, Taylor s University*, April 2013. 18

[28] S. K. S. Preeti Aggarwal, "Analysis of kdd dataset attributes - class wise for intrusion detection," *3rd International Conference on Recent Trends in Computing (ICRTC-2015)*, April 2015. 19

[29] M. H. Aghdam and P. Kabiri, "Feature selection for intrusion detection system using ant colony optimization," *International Journal of Network Security, Vol.18, No.3, PP.420-432*, May 2016. 19