**TRIBHUVAN UNIVERSITY**

**INSTITUTE OF ENGINEERING**

**PULCHOWK CAMPUS**

**DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING**

**THESIS NO.: 071/MSCS/663**

**A**

**THESIS REPORT**

**ON**

# AN APPROACH TO DEVELOP THE HYBRID ALGORITHM BASED ON SUPPORT VECTOR MACHINE AND NAÏVE BAYES FOR ANOMALY DETECTION


**By**

**Sandeep Sigdel**


**SUBMITTED TO THE DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING AS A PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE MASTER'S DEGREE IN COMPUTER SYSTEM AND KNOWLEDGE ENGINEERING**


**DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING**

**LALITPUR, NEPAL**



**OCTOBER, 2016**

# An Approach to Develop the Hybrid Algorithm Based On Support Vector Machine and Naïve Bayes for Anomaly Detection

**By**

**Sandeep Sigdel**

**Thesis Supervisor**

**Prof. Dr. Subarna Shakya**

**A thesis report submitted in partial fulfillment of the requirements for the degree of Master of Science in Computer System and Knowledge Engineering**

**Department of Electronics and Computer Engineering**

**Institute of Engineering, Pulchowk Campus**

**Tribhuvan University**

**Lalitpur, Nepal**

**October, 2016**

# COPYRIGHT ©

**TRIBHUVAN UNIVERSITY**
**INSTITUTE OF ENGINEERING**
**PULCHOWK CAMPUS**
**DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING**

The undersigned certify that they have read, and recommended to the Institute of Engineering for acceptance, a thesis mid-term report entitled "**AN APPROACH TO DEVELOP THE HYBRID ALGORITHM BASED ON SUPPORT VECTOR MACHINE AND NAÏVE BAYES FOR ANOMALY DETECTION**" submitted by **Sandeep Sigdel** in partial fulfillment of the requirements for the degree of Master of Science in Knowledge Base and Computer Engineering.

---

Supervisor: **Prof. Dr. Subarna Shakya**
**Faculty Member**
**Department of Electronics and Computer Engineering**

# DEPARTMENTAL ACCEPTANCE

The thesis entitled "**AN APPROACH TO DEVELOP THE HYBRID ALGORITHM BASED ON SUPPORT VECTOR MACHINE AND NAÏVE BAYES FOR ANOMALY DETECTION**", submitted by **Sandeep Sigdel** in partial fulfillment of the requirement for the award of the degree of "**Master of Science in Computer System and Knowledge Engineering**" has been accepted as a bonafide record of work independently carried out by him in the department.

---

**Dibakar Raj Pant**
Head of the Department
Department of Electronics and Computer Engineering,
Pulchowk Campus,
Institute of Engineering,
Tribhuvan University,
Nepal.

# ACKNOWLEDGEMENT

I am grateful to my Thesis supervisor **Prof. Dr. Subarna Shakya** for his incessant cooperative support, guidance and suggestion at various stages of the Thesis. I deeply value his work as my mentor by providing me with constant counseling and precious feedbacks on Thesis progress.

I would like to express my special thanks of gratitude to the Department of Electronics and Computer Engineering (DOECE) and to our Head of Department **Dr. Dibakar Raj Pant** for providing us with the golden opportunity to explore our interest and ideas in the field of engineering through this thesis. I would like to provide my sincere gratitude to our MSCSKE Coordinator **Dr. Sanjeeb Prasad Pandey** for providing us with necessary details and ideas for preparing the mid-term report of Thesis. I would also like to thank **Prof. Dr. Shashidhar Ram Joshi, Dr. Basanta Joshi** and **Dr. Aman Shakya** for their constant support and encouragement on research activity during master's program.

Finally, I would like to thank all our teachers and friends who have helped me directly or indirectly for encouraging me with this thesis topic and research decision.

# ABSTRACT

Anomaly detection is an important problem that has been researched within diverse research areas and application domains. Many anomaly detection techniques have been specifically developed, and the existing research have low detection rate. This research work proposes a weighted sum formulation for ensemble of Support Vector Machine and Naïve Bayes for anomaly detection, k-fold cross validation to evaluate the error metric associated with a candidate ensemble model and accuracy based weighting scheme to determine the weight values for member algorithms.

The experiment has been conducted in 10% (Knowledge Discovery and Data Mining) KDD dataset. The data has been preprocessed to remove the duplicate records. The categorical data in the 10% KDD dataset has been converted to numeric value using binary encoding scheme. The features of the dataset have been selected using information gain. The grid search has been applied to the dataset using 10-fold cross validation to determine the parameters for Support Vector Machine (SVM). The SVM has been implemented using RBF kernel and value of gamma and C of 0.0001 and 1 respectively.

The hybrid algorithm has been implemented to combine the outcome of prediction of SVM and Naïve Bayes classifiers using weight factors. The weights factors have been calculated using root mean square error of prediction as error metric. The classifier with high accuracy has been given higher weight and classifier with the lower accuracy has been given lower weight.

For the validation of result, ten-fold cross validation has been employed. The performance of SVM classifier, Naïve Bayes classifiers and hybrid algorithm has been compared using Receiver Operating Characteristic (ROC) curve and classification metrics.

**Keywords:** Anomaly Detection, SVM, Naïve Bayes, Hybrid algorithm, Cross Validation, ROC, Classification Metrics.

# TABLE OF CONTENT

# LIST OF FIGURES

# LIST OF TABLES

# LIST OF ABBREVATIONS

| | |
|---|---|
| AUC | Area Under Curve |
| FN | False Negative |
| FP | False Positive |
| HIDS | Host based Intrusion Detection Systems |
| HTTP | Hyper Text Transfer Protocol |
| IDS | Intrusion Detection System |
| KDD | Knowledge Discovery and Data Mining |
| KNN | K-Nearest Neighbor |
| LS-SVM | Least Squares Support Vector Machine |
| NIDS | Network Intrusion Detection Systems |
| OCSVM | One Class Support Vector Machine |
| RAM | Random Access Memory |
| RBF | Radial Basis Function |
| ROC | Receiver Operating Characteristic |
| SVM | Support Vector Machine |
| TN | True Negative |
| TP | True Positive |
| WEKA | Waikato Environment for Knowledge Analysis |

# CHAPTER 1: INTRODUCTION

## 1.1. Background

Anomaly detection refers to the problem of finding patterns in data that do not conform to expected behavior. These nonconforming patterns are often referred to as anomalies, outliers, discordant observations, exceptions, aberrations, surprises, peculiarities, or contaminants in different application domains [1]. Of these, anomalies and outliers are two terms used most commonly in the context of anomaly detection; sometimes inter-changeably. Anomaly detection finds extensive use in a wide variety of applications such as fraud detection for credit cards, insurance, or health care, intrusion detection for internal misuse, cyber-security, fault detection in safety critical systems, and military surveillance for enemy activities.

Due to worldwide proliferation in network environments, a variety of faster services have become a reality. However, the higher the reliance on computers, the more important security problems become. Particularly, the intrusions to the network infrastructures should be detected to minimize damage. Network intrusion detection systems (NIDS) are most efficient way of defending against network-based attack aimed at computer system. Basically there are two approaches of intrusion detection system: signature based (also known as misuse based) and anomaly based [2].

Anomalies are patterns in data that do not conform to a well defined notion of normal behavior. Figure 1 illustrates anomalies in a simple 2-dimensional data set. The data has two normal regions, N1 and N2, since most observations lie in these two regions. Points that are sufficiently far away from the regions, e.g., points O1 and O2, and points in region O3, are anomalies.

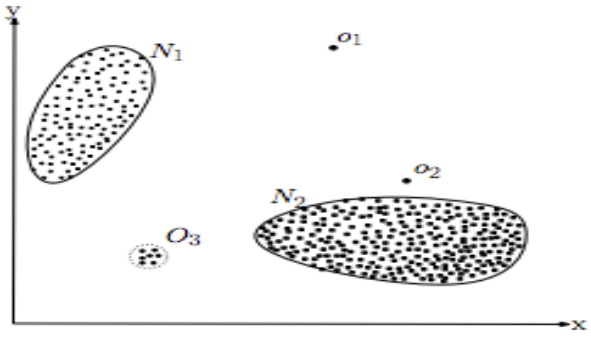


**Figure 1. 1 An example of anomalies in 2-dimensional dataset.**

## 1.2. Problem Definition

Anomaly detection is an important problem that has been researched within diverse research areas and application domains. Many anomaly detection techniques have been specifically developed, and the existing research have low detection rate. The selection of algorithm can largely impact the task of anomaly detection. The anomaly detection algorithms have different computational complexity and accuracies. The selection of improper anomaly detection algorithms can maximize the occurrence of false alarm rate, high resource consumption, and low intrusion detection rate and may result inefficiency to entire system and may even lead to security vulnerabilities. Moreover, one anomaly detection algorithm can outperform the other in particular dataset.

Therefore, ensemble methods can be used to combine the predictions from several candidate classification algorithms in order to improve the generalizability or robustness over a single estimator.

## 1.3. Objective

The main objective of the thesis is to develop the Hybrid algorithm based on SVM and Naïve Bayes for anomaly detection.

## 1.4. Scope

Insider misuse can be defined as the performance of activities where computers and networks in an organization are deliberately misused by those who are not authorized to use them. This scope of the research work is to develop a hybrid algorithm which is a weighted sum formulation for ensemble of Support Vector Machine and Naïve Bayes for anomaly detection. The k-fold cross validation will be used to evaluate the error metric associated with a candidate ensemble model and accuracy based weighting scheme to determine the weight values for member algorithms. The performance of the classifier will be compared to the candidate algorithm using classification metrics and ROC curve.

Therefore, this thesis presents an novel approach to anomaly detection which is applicable in a variety of domains, such as intrusion detection, fraud detection, fault detection, system health monitoring, event detection in sensor networks, and detecting Eco-system disturbances.

# CHAPTER 2: LITERATURE REVIEW

S. Mukkamala, G. Janoski, and A. Sung. On their research work "Intrusion detection using neural networks and support vector machines" were among the first researchers to experiment on anomaly intrusion detection using neural networks and SVMs [3]. They tested the performance of their classifiers on the KDDCUP'99 DARPA dataset. Their classifiers achieved highly accurate results which were greater than 99%. Their SVMs outperformed the neural networks in both training time and detection accuracy. The high detection accuracy might be attributed to the insufficiency of the dataset that they used.

M. Tavallaee, E. Bagheri, W. Lu, and A. Ghorbani in their research work "A detailed analysis of the KDD CUP 99 data set" observed that the KDDCUP'99 dataset had redundant records and it is difficult to compare the IDSs that had been evaluated using this dataset [4].

A. Sung and S. Mukkamala in their research work "Identifying important features for intrusion detection using support vector machines and neural networks" used the SVMs and neural networks for important feature selection and they still demonstrated high detection accuracies [5].

S. Mukkamala, A. Sung, and B. Ribeiro in their research work "Model Selection for Kernel Based Intrusion Detection Systems" performed evaluations of impact kernels on the accuracy of the SVM classifier in intrusion classification [6]. Their experiments still exhibited high detection accuracy rates. They also determined that the ability of SVM classifiers is highly dependent on the kernel type and parameter settings.

H. Chauhan, V. Kuma, S. Pundir and E. S. Pilli, in their research work "Comparative Analysis and Research Issues in Classification Techniques for Intrusion Detection" have presented a comparison between different classification techniques, which are worked to detect intrusions and classify them into normal and abnormal behaviors [7]. The algorithms that have been selected are J48, Naive Bayes, RIPPER (JRip), and One Rule (OneR). Their experiments were performed by using NSL-KDD dataset. WEKA platform was selected for the implementation of the selected algorithms. The results have showed that the best algorithm for classification purpose is OneR classifier, where it required the shortest time, which is around 0.45s with 10-fold cross-validation, and 0.32 s with supplied test set compared with others classifiers.

H. Om and A. Kundu, in their research work "A hybrid system for reducing the false alarm rate of anomaly intrusion detection system" proposed a hybrid intrusion detection

system, which combines K- Means and two classifiers K-Nearest Neighbor (K-NN) and Naïve Bayes [8]. KDD-Cup 1999 dataset are used for evaluation purpose. First, the entropy based feature selection algorithm is used to select the appropriate features. Then, k-means clustering algorithm is applied on the selected features to split the data records into normal and abnormal clusters. After that, the obtained data are classified into normal or abnormal clusters by using the hybrid classifier. The main goals in their approach were to reduce the FAR, detect the intrusions, and further classify them into four categories: DoS, U2R, R2L, and probe. As a result, they have found that the proposed approach is better than the other conventional approaches such as kMeans, kNN, and Naïve Bayes in terms of accuracy, DR, and FAR.

Mukkamala and Sung in their research work "Significant feature selection using computational intelligent techniques for intrusion detection" proposed a novel feature selection algorithm to reduce the feature space of KDD Cup 99 dataset from 41 dimensions to 6 dimensions and evaluated the 6 selected features using an IDS based on SVM [9]. The results show that the classification accuracy increases by 1% when using the selected features.

S. Chebrolu, A. Abraham, J. P. Thomas in their research work "Feature deduction and ensemble design of intrusion detection systems" investigated the performance in the use of a Markov blanket model and decision tree analysis for feature selection, which showed its capability of reducing the number of features in KDD Cup 99 from 41 to 12 features [10].

Y. Chen, A. Abraham, B. Yang in their research work "Feature selection and classification flexible neural tree" proposed an IDS based on Flexible Neural Tree (FNT) [11]. The model applied a pre-processing feature selection phase to improve the detection performance. Using the KDD Cup 99, FNT model achieved 99.19% detection accuracy with only 4 features.

F. Amiri, M. Rezaei Yousefi, C. Lucas, A. Shakery, N. Yazdani in their research work "Mutual information-based feature selection for intrusion detection systems" proposed a forward feature selection algorithm using the mutual information method to measure the relation among features. The optimal feature set was then used to train the LS-SVM classifier and build the IDS [12].

S.-J. Horng, M.-Y. Su, Y.-H. Chen, T.-W. Kao, R.-J. Chen, J.-L. Lai, C. D. Perkasa in their research work "A novel intrusion detection system based on hierarchical clustering and support vector machines" proposed an SVM-based IDS, which combines a

hierarchical clustering and the SVM [13]. The hierarchical clustering algorithm was used to provide the classifier with fewer and higher quality training data to reduce the average training and testing time and improve the classification performance of the classifier. Experimented on the corrected labels KDD Cup 99 dataset, which includes some new attacks, the SVM-based IDS scored an overall accuracy of 95.75% with a false positive rate of 0.7%.

A. H. Katherine in research work "One Class Support Vector Machines for Detecting Anomalous Windows Registry Accesses" presents a new Host-based Intrusion Detection System (IDS) that monitors accesses to the Microsoft Windows Registry using Registry Anomaly Detection (RAD). The system uses a one class Support Vector Machine (OCSVM) to detect anomalous registry behavior and detect outliers in new (unclassified) data generated from the same system [14].

M. Zhang et al. on research work "An Anomaly Detection Model Based on One-Class SVM to Detect Network Intrusions" proposes an anomaly detection model based on One-class SVM to detect network intrusions. The one-class SVM adopts only normal network connection records as the training dataset. But after being trained, it is able to recognize normal from various attacks [15].

R. Chitrakar et al. on research work "Anomaly detection using Support Vector Machine classification with k-Medoids clustering" proposed a better combination to address problems of the previously proposed hybrid approach of combining k-Means/k-Medoids clustering technique with Naïve Bayes classification. In this approach, the need of large samples by the previous approach is reduced by using Support Vector Machine while maintaining the high quality clustering of k-Medoids [16].

N. B. Amor et al. on research work "Naive bayes vs decision trees in intrusion detection systems" performed a comparison between two classifiers native Bayes networks and decision tree using KDD Cup dataset 1999 [17]. Native Bayes and decision tree having their own decision capable to detect the intrusion. Both performed equally however, while detecting U2R and probe native bayes performed better and in normal, DOS and R2L decision tree performed better.

R. Jain et al. on research work "Network attacks, classification and models for anomaly based network intrusion detection system" presents a selective survey of incremental approaches for detecting anomaly in normal system and network traffic [18].

S. S. Murtaza et al. research on "A host-based anomaly detection approach by representing system calls as states of kernel modules" attempts to reduce the false alarm

rate and processing time while increasing the detection rate. The paper presents a novel anomaly detection technique based on semantic interactions of system calls which analyzes the state interactions, and identifies anomalies by comparing the probabilities of occurrences of states in normal and anomalous traces [19].

C. Manikopoulos and S. Papavassiliou research on "Network intrusion and fault detection: a statistical anomaly approach" applies neural network and SVM classifiers that was used to detect the anomalies [20]. The main objective of this paper was to create robust, effective and efficient classifiers which detects the intrusion in the real-time. The idea was to discover patterns or features that describe the user behavior. In this approach both neural network and SVM perform better rather than another technique of classifiers.

X. Yingchao et al. research on "Parameter Selection of Gaussian Kernel for One-Class SVM" proposes a novel method to solve the problem of kernel parameter selection in one class classifier, specifically, one-class SVM (OCSVM) [21].

D. P. Gaikwad et al. research on "Intrusion Detection System Using Bagging Ensemble Method of Machine Learning" presents a novel intrusion detection technique based on ensemble method of machine learning is proposed. The Bagging method of ensemble with REPTree implement intrusion detection system [22].

A. J. Hoglund et al. research on "A computer host-based user anomaly detection system using the self-organizing map" aimed at designing a system that contains an automatic anomaly detection component [23]. A prototype UNIX anomaly detection system was constructed for anomaly detection attempts to recognize abnormal behavior to detect intrusions. The component for detection used a test based on the self-organizing map to test if user behavior is anomalous.

I. S. Thaseen paper titled "Intrusion detection model using fusion of PCA and optimized SVM" proposes a novel method of integrating principal component analysis (PCA) and support vector machine (SVM) by optimizing the kernel parameters using automatic parameter selection technique. This technique reduces the training and testing time to identify intrusions thereby improving the accuracy. The proposed method was tested on KDD data set [24].

L. Lin et al. research on "SVM ensemble for anomaly detection based on rotation forest" proposes a new intelligent intrusion detection system using SVM ensemble. The ensemble was made of two-layer, one is composed by five SVM network decided by winner-take-all, the other is a ensemble network composed of five classifier decided by majority voting [25].

S. Peddabachigari et al. paper on “Modeling intrusion detection system using hybrid intelligent systems” provides a new hybrid approach called DTSVM (Decision trees - SVM) in which two classifiers decision tree and SVM were used as an individual base classifier. The motive of this hybrid technique was to increase the detection accuracy and reduce the computational complexity. This hybrid approach was provided better accuracy than the individual classifier. The paper gives a great idea or a new concept of using multiple classifiers to improve the detection accuracy and reduce the computational complexity [26].

P. Amudha et al. paper titled “Intrusion detection based on Core Vector Machine and ensemble classification methods” proposes a combined algorithm based on Principal Component Analysis (PCA) and Core Vector Machine (CVM), which is an extremely fast classifier, for intrusion detection [27]. PCA was used as feature extraction technique to select principal features from the intrusion detection KDDCup'99 dataset and an intrusion detection model was constructed by CVM algorithm. The effectiveness of the features selected was also tested on ensemble based classifiers and the results are compared with the standard classifiers.

P. Sornsuwit and S. Jaiyen research titled “Intrusion detection model based on ensemble learning for U2R and R2L attacks” concentrates on ensemble learning for detecting network intrusion data, which are difficult to detect. In addition, correlation- based algorithm was used for reducing some redundant features. Adaboost algorithm was adopted to create the ensemble of weak learners in order to create the model that can protect the security and improve the performance of classifiers. The U2R and R2L attacks in KDD Cup'99 intrusion detection dataset were used to train and test the ensemble classifiers. The experimental results show that reducing features can improve efficiency in attack detection of classifiers in many weak leaners [28].

Hu, C., Youn, B. D., & Wang, P. in their research work “Ensemble of data-driven prognostic algorithms with weight optimization and k-fold cross validation” have proposed ensemble of data-driven prognostic algorithms with weight optimization and k-fold cross validation. They have calculated the weights for candidate algorithms using cross-validation error of the candidate algorithm [29].

Zerpa et al. “An optimization methodology of alkaline–surfactant–polymer flooding processes using field scale numerical simulation and multiple surrogates” proposed weighting average model based on variance of prediction to combine the individual surrogates model to build a hybrid model [30].

Guang, Yang, and Nie Min.in research work "Anomaly intrusion detection based on wavelet kernel LS-SVM" proposed an intrusion detection method based on wavelet kernel Least Square Support Vector Machine (LS-SVM) that has higher detection accuracy rate and lower false alarm rate [31].

Pervez, Muhammad Shakil, and Dewan Md Farid in their research work “Feature selection and intrusion classification in NSL-KDD cup 99 dataset employing SVMs”, have applied SVM classifier on several feature subsets of NSL- KDD Cup 99 dataset and the experimental analysis shown that the proposed method achieved 91% classification accuracy using only three input features and 99% classification accuracy using 36 input features, while all 41 input features achieved 99% classification accuracy. They have planned to ensemble other mining classifiers with SVM to achieve the good classification accuracy of the minority class instances [32].

# CHAPTER 3: RELATED THEORY

## 3.1. Intrusion Detection Systems

Intrusion detection is defined as any set of actions that compromise the integrity, confidentiality and availability of a resource [33]. Any system that therefore ensures that the integrity, confidentiality and availability of its resources are not compromised is said to be a secure system. IDS therefore, attempts to recognize activities as either legitimate or intrusive and notify users. There are two types of intrusion detection systems namely Host based IDS and Network based IDS. The host based IDS (HIDS) monitors the characteristics of a single host and the events occurring within that host for any suspicious activity. Network based IDS (NIDS) on the other hand monitors network traffic for particular network segments or devices and analyses the network and application protocol activity to identify any sign of suspicious activity. There are two main strategies for detection. One is misuse detection, which attempts to match patterns and signatures of already known attacks in network traffics [33]. And another is anomaly detection, which is adaptive in nature. They attempt to identify behaviors that do not conform to normal behaviors [33].

## 3.2. Anomaly Detection

Anomaly detection is essentially the classification problem. It is the task of finding data points that deviate from the rest of the data. It can also be called outlier detection or deviation detection [34]. Presently, a number of various automatic methods exist and can be used instead. Anomaly detection can be applied to a variety of different problems, such as using it for an intrusion detection system, for finding instances of fraud [35], or for finding out if a safety critical system is running in an abnormal way before any major harm is done [34]. Hill and Minsker stress the necessity of not using abnormal data to make predictions, as it would cause the predictions to in part view abnormal events as normal [36]. It is also important to keep in mind how the alarms raised by the anomaly detection are to be handled. There are a number of different methods that can be used to perform anomaly detection, and most can be grouped into being statistical, supervised, or unsupervised.

## 3.3. Supervised Anomaly Detection

Supervised anomaly detection requires pre-classified training data that defines normal behavior. Supervised algorithms may not be able to properly handle data from an unexpected region, as it was not contained in the training data [34]. Thus, it is important for the training data to cover as much of normal behavior as possible, including examples of normal and abnormal data, which may be difficult and resource draining to assemble.

## 3.4. Unsupervised Anomaly Detection

Unsupervised anomaly detection does not require any pre-classified training data or input from a human teacher, it learns about the normality of the data in an unsupervised manner. It is able to find outliers without having any prior knowledge of the data by processing a batch of data [34]. An advantage of unsupervised methods as compared to supervised methods is the possibility of detecting previously unknown types of faults [37]. One popular unsupervised approach is clustering, a method that tries to segment data into groups. It can be used to find natural groupings and patterns in data. It is one of the most popular unsupervised techniques, but it can suffer from choosing data metrics poorly [37]. When used to detect malicious anomalies, for example during network surveillance, it relies of the observation that malicious events often are related, rather than occurring separately [38].

## 3.5. Classification

Consider the problem of separating the set of training vectors belong to two separate classes, $(x_1, x_2), (x_2, y_2), \ldots, (x_n, y_n)$ where $x_i \in R^p$ and $y_i \in \{-1, +1\}$ is the corresponding class label, $1 \leq i \leq n$. The main task is to find a classifier with a decision function $f(\mathbf{x}, \theta)$ such that $y = f(\mathbf{x}, \theta)$, where y is the class label for $\mathbf{x}$, $\theta$ is a vector of unknown parameters in the function [39].

## 3.6. Support Vector Machine

Support vector machines (SVMs) is a method for the classification of both linear and nonlinear data. It uses a nonlinear mapping to transform the original training data into a higher dimension. Within this new dimension, it searches for the linear optimal separating hyperplane (i.e., a “decision boundary” separating the tuples of one class from another).

With an appropriate nonlinear mapping to a sufficiently high dimension, data from two classes can always be separated by a hyperplane. The SVM finds this hyperplane using support vectors ("essential" training tuples) and margins (defined by the support vectors) [40].

The theory of SVM is from statistics and the basic principle of SVM is finding the optimal linear hyperplane in the feature space that maximally separates the two target classes [41]. Geometrically, the SVM modeling algorithm finds an optimal hyperplane with the maximal margin to separate two classes, which requires to solve the following constraint problem can be defined as

$$minimize_{w,b}\frac{1}{2}\left||w|\right|^2 \tag{1}$$

subject to:

$$y_i(w^T x_i + b) \geq 1, i = 1,2,3,\dots,n \tag{2}$$

To allow errors, the optimization problem now becomes:

$$minimize_{w,b}\frac{1}{2}\left||w|\right|^2 + C\sum_{i=1}^{n}\xi_i \tag{3}$$

subject to:

$$y_i(w^T x_i + b) \geq 1 - \xi_i, i = 1,2,3,\dots,n \tag{4}$$

$$\xi_i \geq 0, i = 1,2,3,..,n \tag{5}$$

Using the method of Lagrange multipliers, we can obtain the dual formulation which is expressed in terms of variable $\alpha_i$

$$maximize_{\alpha}\sum_{i=1}^{n}\alpha_i - \sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j x_i^T x_j \tag{6}$$

Subject to:

$$\sum_{i=1}^{n} y_i\alpha_i = 0, 0 < \alpha_i < C, i = 1,2,3,\dots,n \tag{7}$$

Finally, the linear classifier based on a linear discriminant function takes the following form

$$f(x) = \sum_{i=1}^{n}\alpha_i x_i^T x + b \tag{8}$$

In many applications a non-linear classifier provides better accuracy. The naive way of making a non-linear classifier out of a linear classifier is to map our data from the input space X to a feature space F using a non-linear function $\emptyset: X \rightarrow F$. In the space $F$, the optimization takes the following form using kernel function [19]:

$$maximize_{\alpha} \sum_{i=1}^{n} \alpha_i - \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j k(x_i, x_j) \tag{9}$$

Subjected to:

$$\sum_{i=1}^{n} y_i \alpha_i = 0, 0 < \alpha_i < C\,, i = 1,2,3, \dots, n \tag{10}$$

Finally, in terms of the kernel function the discriminant function takes the following form:

$$f(x) = \sum_{i=1}^{n} \alpha_i k(x, x_i) + b \tag{11}$$

The RBF or Gaussian kernel is given by:

$$\mathrm{K(x, y)} = \mathrm{e}^{-||\mathrm{x-y}||^2/2\sigma^2} \tag{12}$$

where $\sigma$ is the width of function.

Equivalently in terms of $\gamma$ parameter, where $\gamma = \frac{1}{2\sigma^2}$

$$\mathrm{K(x, y)} = \mathrm{e}^{-\gamma||\mathrm{x-y}||^2} \tag{13}$$

Instead of predicting the label, many applications require a posterior class probability $\mathrm{p_r(y = 1|x)}$. Platt proposes approximating the posterior by a sigmoid function [42]

$$p_r(y = 1|x) \approx p_{A,B}(f) = \frac{1}{1 + e^{Af+B}} \tag{14}$$

where $\mathrm{f = f(x)}$

Let each $\mathrm{f_i}$ be an estimate of $\mathrm{f(x_i)}$. The best parameter setting $\mathrm{z* = (A*, B*)}$ is determined by solving the following regularized maximum likelihood problem (with N+ of the $\mathrm{y_i's}$ positive, and N− negative) [42]:

$$\min_{z} F(z) = \sum_{i=1}^{l} (t_i \log(p_i) + (1 - t_i) \log(1 - p_i)) \tag{15}$$

for $\mathrm{p_i = P_{A,B}(f_i)}$, and $\mathrm{t_i} = \begin{cases} \frac{\mathrm{N_+ + 1}}{\mathrm{N_+ + 2}} \text{ if } \mathrm{y_i} = 1 \\ \frac{1}{\mathrm{N_- + 2}} \text{ if } \mathrm{y_i} = -1 \end{cases}$ $\mathrm{i = 1,2,3, \dots, l}$

## 3.7. Naive Bayes

Bayesian classifiers are statistical classifiers. They can predict class membership probabilities such as the probability that a given tuple belongs to a particular class. Bayesian classification is based on Bayes' theorem. Bayesian classifiers has exhibited high accuracy and speed when applied to large databases.

Naive Bayes methods are a set of supervised learning algorithms based on applying Bayes' theorem with the "naive" assumption of independence between every pair of features. Given a class variable $\boldsymbol{y}$ and a dependent feature vector $(\boldsymbol{x_1}, \boldsymbol{x_2}, \ldots, \boldsymbol{x_n})$ **,** Bayes' theorem states the following relationship [43]:

$$p(y|x_1, \ldots\ldots, x_n) = \frac{p(y)p(x_1, \ldots x_n|y)}{p(x_1, \ldots., x_n)} \tag{16}$$

Using the naive independence assumption that

$$p(x_i|y, x_1, \ldots x_n) = p(x_i|y) \tag{17}$$

For all *i*, this relationship is simplified to

$$p(y|x_1, \ldots\ldots, x_n) = \frac{p(y)\prod_{i=1}^{n} p(x_i|y)}{p(x_1, \ldots., x_n)} \tag{18}$$

Since $p(y|x_1, \ldots\ldots, x_n)$ is constant given the input, we can use the following classification rule:

$$p(y|x_1, \ldots\ldots, x_n) \propto p(y) \prod_{i=1}^{n} p(x_i|y) \tag{19}$$

Assuming that each attribute follows a normal distribution, the likelihood of feature is given by [44]

$$p(x_i|y) = \frac{1}{\sqrt{2\pi}\sigma_y} \exp\left(-\left(x_i - \mu_y\right)^2/2\sigma_y^2\right) \tag{20}$$

The mean $\mu_y$ and and standard deviation $\sigma_y$ are estimated from training data.

## 3.8. Information Gain

Information gain measures the amount of information in bits about the class prediction, if the only information available is the presence of a feature and the corresponding class distribution. Concretely, it measures the expected reduction in entropy (uncertainty associated with a random feature). The information gain of an attribute tells how much information with respect to the classification target the attribute gives you. That is, it measures the difference in information between the cases where you know the value of the attribute and where you don't know the value of the attribute.

Let *S* be a set of training set samples with their corresponding labels. Suppose there are *m* classes and the training set contains $s_i$ samples of class *I* and *s* is the total number of samples in the training set. Expected information needed to classify a given sample is calculated by [45]:

$$I(s_1, s_2, \dots, s_m) = -\sum_{i=1}^{m} \frac{s_i}{s} \log_2 (\frac{s_i}{s}) \tag{21}$$

A feature *F* with values $\{f_1, f_2, .., f_v\}$ can divide the training set into v subsets $\{S_1, S_2, .., S_v\}$ where $S_j$ is the subset which has the value $f_j$ for feature *F*. Furthermore, let $S_j$ contains $s_{ij}$ samples of class *i*. Entropy of feature *F* is [44]

$$E(F) = \sum_{j=1}^{v} \frac{(s_{1j} + .. + s_{mj})}{s} * I(s_{1j}, s_{2j}, .., s_{mj}) \tag{22}$$

Information Gain for *F* can be calculated as follow [44]:

$$Gain(F) = I(s_1, s_2, .., s_m) - E(F) \tag{23}$$

# CHAPTER 4: METHODOLOGY

## 4.1. Data Collection

The experiment has been conducted on 10% KDD labeled dataset. The 10% KDD dataset consist of 10% of original dataset that has approximately 494,021 records. The dataset has 19.69% normal and 80.31% attack connections [39]. Each record has 41 attributes describing different features and a label assigned to each either as an 'attack' type or as 'normal'.

**Table 4. 1 Features of KDD dataset.**

| SN | Feature | Description |
|---|---|---|
| 1 | duration | duration of connection in seconds |
| 2 | protocol_type | connection protocol (tcp, udp, icmp) |
| 3 | service | dst port mapped to service (e.g. http, ftp, ..) |
| 4 | flag | normal or error status flag of connection |
| 5 | src_bytes | number of data bytes from src to dst |
| 6 | dst_bytes | bytes from dst to src |
| 7 | land | 1 if connection is from/to the same host/port; else 0 |
| 8 | wrong_fragment | number of ‘wrong’ fragments (values 0,1,3) |
| 9 | urgent | number of urgent packets |
| 10 | hot | number of ‘hot’ indicators (bro-ids feature) |
| 11 | num_failed_logins | number of failed login attempts |
| 12 | logged_in | 1 if successfully logged in; else 0 |
| 13 | num_compromised | number of ‘compromised’ conditions |
| 14 | root_shell | 1 if root shell is obtained; else 0 |
| 15 | su_attempted | 1 if ‘su root’ command attempted; else 0 |
| 16 | num_root | number of ‘root’ accesses |
| 17 | num_file_creations | number of file creation operations |
| 18 | num_shells | number of shell prompts |
| 19 | num_access_files | number of operations on access control files |
| 20 | num_outbound_cmds | number of outbound commands in an ftp session |
| 21 | is_hot_login | 1 if login belongs to ‘hot’ list (e.g. root, adm); else 0 |
| 22 | is_guest_login | 1 if login is ‘guest’ login (e.g. guest, anonymous); else 0 |

| | | |
|---|---|---|
| 23 | count | number of connections to same host as current connection in past two seconds |
| 24 | srv_count | number of connections to same service as current connection in past two seconds |
| 25 | serror_rate | % of connections that have 'SYN' errors |
| 26 | srv_serror_rate | % of connections that have 'SYN' errors |
| 27 | rerror_rate | % of connections that have 'REJ' errors |
| 28 | srv_rerror_rate | % of connections that have 'REJ' errors |
| 29 | same_srv_rate | % of connections to the same service |
| 30 | diff_srv_rate | % of connections to different services |
| 31 | srv_diff_host_rate | % of connections to different hosts |
| 32 | dst_host_count | count of connections having same dst host |
| 33 | dst_host_srv_count | count of connections having same dst host and using same service |
| 34 | dst_host_same_srv_rate | % of connections having same dst port and using same service |
| 35 | dst_host_diff_srv_rate | % of different service on current host |
| 36 | dst_host_same_src_port_rate | % of connections to current host having same src port |
| 37 | dst_host_srv_diff_host_rate | % of connections to same service coming from diff. hosts |
| 38 | dst_host_serror_rate | % of connections to current host that have an S0 error |
| 39 | dst_host_srv_serror_rate | % of connections to current host and specified service that have an S0 error |
| 40 | dst_host_rerror_rate | % of connections to current host that have an RST error |
| 41 | dst_host_srv_rerror_rate | % of connections to current host and specified service that have an RST error |

## 4.2. Data preprocessing

The duplicate records are removed. The features should be normalized if necessary. SVM and Naïve Bayes classification systems are not able to process the KDD'99 dataset in its current format as there are categorical features present in the dataset [39]. SVM requires that each data instance is represented as a vector of real numbers. Hence preprocessing is required before SVM classification system could be built.

Preprocessing contains the following processes: The features in columns 2, 3, and 4 in the KDD'99 dataset are the protocol type, the service type, and the flag, respectively. The value of the protocol type may be tcp, udp, or icmp; the service type could be one of the 66 different network services such as http and smtp; and the flag has 11 possible values such as SF or S2. Hence, the categorical features in the KDD dataset must be converted into a numeric representation. This is done by the usual binary encoding; each categorical variable having possible **m** values is replaced with **m − 1** dummy variables [39]. Here a dummy variable has value one for a specific category and having zero for all category.

**Table 4. 2 Example of dataset with categorical features.**

| Protocol | Packet count | Sent byte | Received byte |
|---|---|---|---|
| tcp | 5 | 224 | 325 |
| udp | 6 | 435 | 223 |
| icmp | 22 | 525 | 415 |

**Table 4. 3 Final dataset after binary encoding of features.**

| Protocol=tcp | Protocol=udp | Protocol=icmp | Packet count | Sent byte | Received byte |
|---|---|---|---|---|---|
| **1** | 0 | 0 | 5 | 224 | 325 |
| 0 | **1** | 0 | 6 | 435 | 223 |
| 0 | 0 | **1** | 22 | 525 | 415 |

## 4.3. Feature Selection

Feature selection is a form of search in the training data. It selects a subset of input features d from a total of D original input features in the training data by using an optimization of scientific theorem to improve the classification accuracy of a learning classifier [32]. In general terms feature selection is the process of searching through the subsets of features in training data, and tries to find the best one. In complex classification area like IDS, feature selection is really necessary as irrelevant and redundant input features in training data make a complex classification model and also reduce the classification rate.

The result of feature selection using the information gain technique and a ranker is presented in Table 4.4 [46].

**Table 4. 4 Feature selected using information gain.**

| Rank | Value | Feature |
|---|---|---|
| 1 | 0.8162 | Src_bytes |
| 2 | 0.6715 | Service |
| 3 | 0.633 | Dst_bytes |
| 4 | 0.5193 | flag |
| 5 | 0.5186 | Diff_srv_rate |
| 6 | 0.5098 | Same_srv_rate |
| 7 | 0.4759 | Dst_host_srv_count |
| 8 | 0.4382 | Dst_host_same_srv_rate |
| 9 | 0.4109 | Dst_host_diff_srv_rate |
| 10 | 0.4059 | Dst_host_serror_rate |
| 11 | 0.4047 | Logged_in |
| 12 | 0.398 | Dst_host_srv_serror_rate |
| 13 | 0.3927 | Serror_rate |
| 14 | 0.3835 | count |
| 15 | 0.3791 | Srv_serror_rate |

## 4.4. SVM Model Selection

There are two parameters for an RBF kernel: C and γ. It is not known beforehand which C and γ are best for a given problem; consequently, some kind of model selection (parameter search) must be done [47]. The gamma parameter defines how far the influence of a single training example reaches, with low values meaning ‘far’ and high values meaning ‘close’. The C parameter trades off misclassification of training examples against simplicity of the decision surface. A low C makes the decision surface smooth, while a high C aims at classifying all training examples correctly by giving the model freedom to select more samples as support vectors. The goal of grid search is to identify good (C, γ) so that the classifier can accurately predict unknown data (i.e. testing data).

In k-fold cross-validation, the training set is divided into k subsets of equal size. Sequentially one subset is tested using the classifier trained on the remaining k−1 subsets. Thus, each instance of the whole training set is predicted once so the cross-validation accuracy is the percentage of data which are correctly classified.

The recommended approach is to perform a "grid-search" on C and γ using cross-validation. Various pairs of (C, γ) values are tried and the one with the best cross-validation accuracy is picked. The exponentially growing sequences of C and γ is a practical method to identify good parameters [47].

It is to be noted that, although RBF SVM cannot learn well when a very low value of C is used, its performance largely depends on the γ value if a roughly suitable C is given [48]. Clearly, changing γ leads to larger variation on test error than changing C. This means that over a large range of C, the performance of RBF SVM can be adjusted by simply changing the value of γ [49]. It is known that, in a certain range, a large γ often leads to a reduction in classifier complexity but at the same lowers the classification performance. Also, a smaller γ often increases the learning complexity and leads to higher classification performance in general.

## 4.5. Hybrid Algorithm

This proposed hybrid algorithm is a weighted sum formulation for ensemble of Support Vector Machine and Naïve Bayes for anomaly detection. An ensemble of prognostic member algorithms for target prediction can be expressed in a weighted-sum formulation as [30]:

$$y_h(x) = \sum_{i=1}^{M} w_i y_i(x) \tag{24}$$

where $y_h(\boldsymbol{x})$ is the hybrid algorithm predicted response, M is the number of classifiers in the hybrid algorithm, $w_i$ is the weight factor for the $i^{th}$ classifier, $y_i(x)$ is the response estimated by the $i^{th}$ classifier, and $\boldsymbol{x}$ is the vector of independent input variables.

The error in prediction of a classifier is defined as

$$e_i = y^i - y_p^i \tag{25}$$

where $y^i, y_p^i$ represent the actual value and predicted value respectively.

The process is repeated for all n observations and the root mean square error (*RMSE*) is computed as

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n} e_i^2} \tag{26}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y^i - y_p^i\right)^2} \tag{27}$$

The prediction accuracy of the j[th] member algorithm is quantified by Cross Validation root mean square error, expressed as

$$\varepsilon_{cv}^{j} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y_j^i - y_{j,p}^i\right)^2} \tag{28}$$

The weight $w_j$ of the j[th] member algorithm can then be defined as the normalization of the corresponding inverse CV error [29], expressed as

$$w_j = (1/\varepsilon_{cv}^{j}) \sum_{i=1}^{M} \frac{1}{\varepsilon_{cv}^{i}} \tag{29}$$

This definition indicates that a larger weight is assigned to a member algorithm with higher prediction accuracy. Thus, a member algorithm with better prediction accuracy has a larger influence on the ensemble prediction. This weighting scheme relies exclusively on the prediction accuracy to determine the weights of member algorithms.

In the proposed hybrid algorithm, the member algorithms are SVM and Naïve Bayes. The prediction error for SVM and Naïve Bayes are $e_{svm}$ and $e_{nb}$respectively are calculated as follow:

$$e_{svm}^{i} = y^i - y_{svm}^{i} \tag{30}$$

$$e_{nb}^{i} = y^i - y_{nb}^{i} \tag{31}$$

The root mean square error of prediction for SVM and Naïve Bayes at k fold of cross validation are $RMSE_{svm}$ and $RMSE_{nb}$ respectively are defined by

$$RMSE_{svm}^{k} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(e_{svm}^{i}\right)^2} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y^i - y_{svm}^{i}\right)^2} \tag{32}$$

$$RMSE_{nb}^{k} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(e_{nb}^{i}\right)^2} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y^i - y_{nb}^{i}\right)^2} \tag{33}$$

The weight factors for SVM and Naïve Bayes are $w_{svm}$and $w_{nb}$respectively at k fold of cross validation and are defined by

$$w_{svm}^{k} = \frac{RMSE_{nb}}{RMSE_{svm} + RMSE_{nb}} \tag{34}$$

$$w_{nb}^{k} = \frac{RMSE_{svm}}{RMSE_{svm} + RMSE_{nb}} \tag{35}$$

The weights choosen has the constraint defined by

$$w_{svm}^{k} + w_{nb}^{k} = 1 \tag{36}$$

Finally, the prediction of hybrid algorithm is given by

$$y_{hyb}^{i} = w_{nb}^{k} * y_{nb}^{i} + w_{svm}^{k} * y_{svm}^{i} \tag{37}$$

$$y_{hyb}^{i} = \frac{RMSE_{svm}^{k}}{RMSE_{svm}^{k} + RMSE_{nb}^{k}} * y_{nb}^{i} + \frac{RMSE_{nb}^{k}}{RMSE_{svm}^{k} + RMSE_{nb}^{k}} * y_{svm}^{i} \tag{38}$$

## 4.6. System Flow Diagram

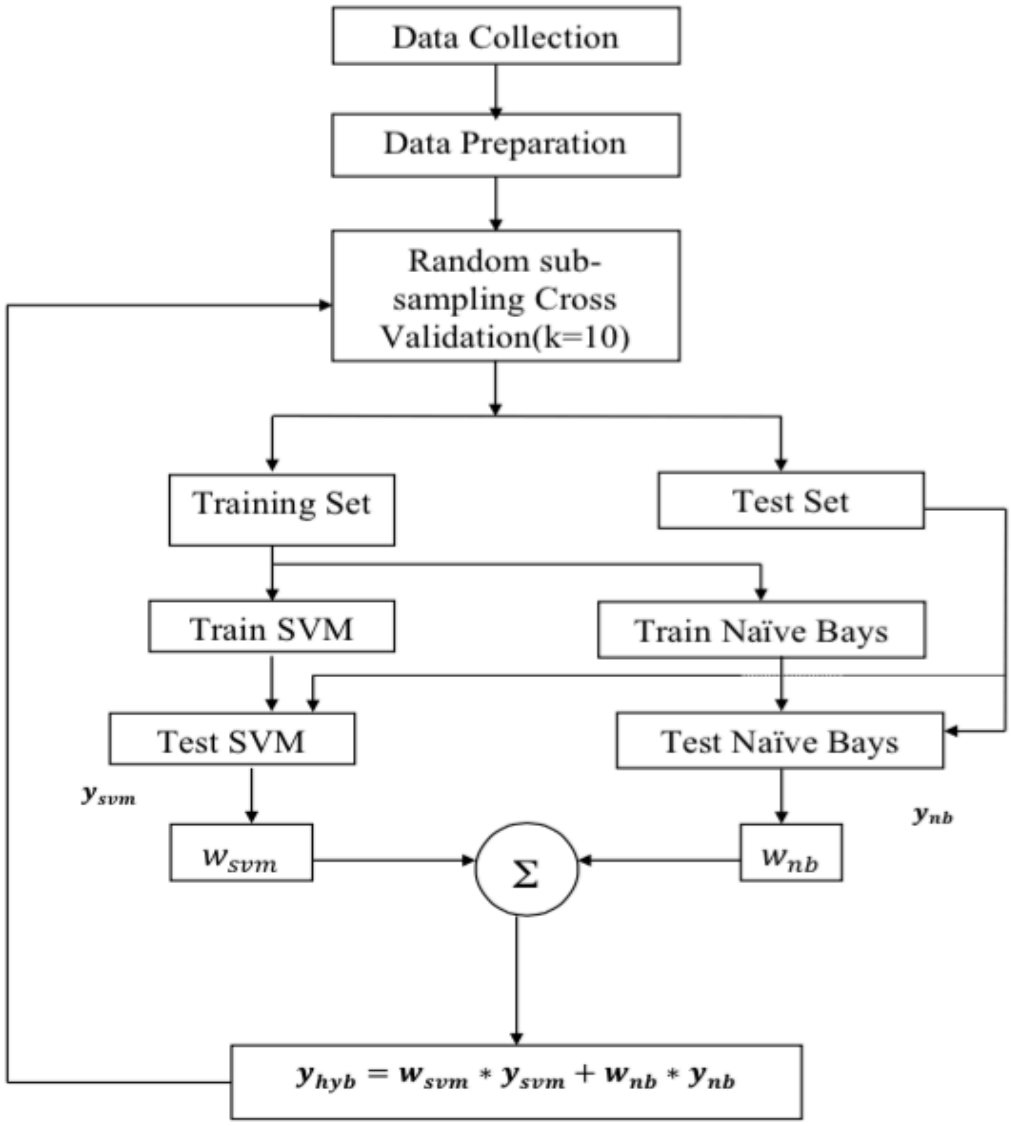


**Figure 4. 1 System flow diagram of hybrid algorithm.**

Figure 4.1 shows the block diagram of Hybrid algorithm. The preprocessed data is randomly partitioned into 10 equal size subsamples. Of the 10 subsamples, a single subsample is retained as the validation data for testing the model, and the remaining 9 subsamples are used as training data. The cross-validation process is then repeated 10 times (the folds), with each of the 10 subsamples used exactly once as the validation data. The 10 results from the folds can then be averaged (or otherwise combined) to produce a single estimation.

At each stage of cross validation, training data is used to train SVM and Naïve Bayes classifier. The test data is used by SVM and Naïve Bayes for predicting the target output. At each fold of cross validation root mean square error is calculated for prediction of both SVM and Naïve Bayes. These error values are used to calculate the weight to be used with the member (SVM and Naïve Bayes) classifiers of Hybrid algorithm. Finally, the output of hybrid algorithm is calculated as weight sum of output from SVM and Naïve Bayes classifier.

## 4.7. Evaluate Receiver Operating Characteristic Curve

A ROC curve provides the possibility to observe, how the change of a free parameter influences the performance of the system. It represents the relation between the true positive rate and the false positive rate of a retrieval algorithm for each parameter value in one plot [50]. The quality of a ROC curve is often summarized using the area under the curve (AUC). Higher the value of AUC scores better is the classification. A perfect classification has an area of 1.00.

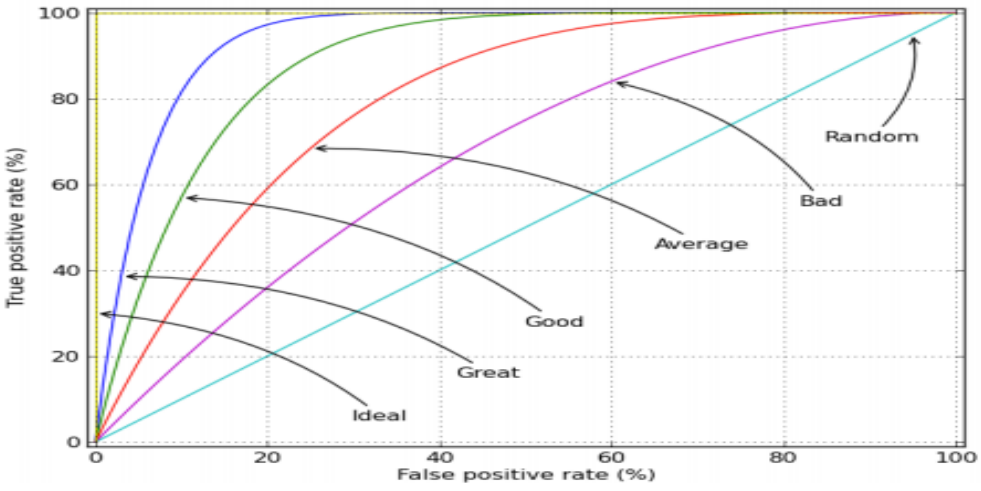


**Figure 4. 2 Illustration of ROC curve.**

## 4.8. Evaluate Classification Metrics

The performance measures of a classifier measures how accurate the classifier predicting the class label of instances (both training and testing instances) [32]. To compute the performance of learning classifiers we need to know the four terms: (1) True positives, TP (the positive instances are correctly classified by the learning algorithm), (2) True negatives, TN (the negative instances are correctly classified by the learning algorithm), (3) False positives, FP (the negative instances are misclassified as positive by the learning algorithm), and (4) False negatives, FN (the positive instances are misclassified as negative by the learning algorithm).

**Table 4. 5 Confusion matrix for anomaly detection.**

| | | Prediction | |
|---|---|---|---|
| | | Normal | Abnormal |
| Actual Class | Normal | TN | FP |
| | Abnormal | FN | TP |

There are many metrics that can be used to measure the performance of a classifier or predictor.

**Accuracy:**

The accuracy of a learning algorithm on a given test data is the percentage of test instances that are correctly classified by the learning algorithm, which is measured as the ratio of correctly classified data to the total classified data.

$$accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{39}$$

**Error rate:**

The error rate of a learning algorithm on a given test data is the percentage of test set instances that are misclassified by the learning algorithm.

$$error\ rate = \frac{FP + FN}{TP + FP + TN + FN} \tag{40}$$

**Sensitivity:**

The sensitivity is referred to as the true positive rate (i.e., the proportion of positive instances that are correctly identified by learning algorithm).

$$sensitivity = \frac{TP}{P} = \frac{TP}{TP + FN} \tag{41}$$

**Specificity:**

The specificity is referred to as the true negative rate (i.e., the proportion of negative instances that are correctly identified by learning algorithm).

$$specificity = \frac{TN}{N} = \frac{TN}{TN + FP} \tag{42}$$

**Precision:**

Precision can be thought of as a measure of exactness (i.e., what percentage of instances labeled as positive are actually such).

$$precision = \frac{TP}{TP + FP} \tag{43}$$

**Recall:**

Recall is a measure of completeness (what percentage of positive instances are labeled as such).

$$recall = \frac{TP}{P} = \frac{TP}{TP + FN} \tag{44}$$

**F1-Score:**

It is basically used to measure the effectiveness of the classifiers. F1-scores is the harmonic mean of precision and recall.

$$F1 - score = \frac{2(precision * recall)}{precision + recall} \tag{45}$$

## 4.9. Validation

To estimate the area under curve (AUC) performance of a two-class classifier, a technique called cross validation is employed. In k-fold cross-validation, the original sample is randomly partitioned into k equal size subsamples. Of the k subsamples, a single subsample is retained as the validation data for testing the model, and the remaining k-1 subsamples are used as training data [47]. The cross-validation process is then repeated k times (the folds), with each of the k subsamples used exactly once as the validation data. The k results from the folds can then be averaged (or otherwise combined) to produce a single estimation. The advantage of this method is that all observations are used for both training and validation, and each observation is used for validation exactly once.

# CHAPTER 5: RESULT ANALYSIS AND COMPARISION

The experiments were carried out by a MacBook Pro with Retina display with 2.5 GHz quad-core Intel Core i5 Processor and 4 GB of RAM. The proposed algorithm was tested by applying the receiver operating characteristics on classification output of Hybrid algorithm, SVM and Naïve Bayes on the 10% KDD Cup 99 dataset.

## 5.1. Result

### 5.1.1. SVM Grid Search

**Table 5. 1 Model selection of RBF SVM using grid search.**

| Gamma(γ) | C | Accuracy |
|---|---|---|
| 1.00E-09 | 1 | 0.9190 |
| 1.00E-09 | 10 | 0.9436 |
| 1.00E-09 | 100 | 0.9465 |
| 1.00E-08 | 1 | 0.9436 |
| 1.00E-08 | 10 | 0.9464 |
| 1.00E-08 | 100 | 0.9470 |
| 1.00E-07 | 1 | 0.9587 |
| 1.00E-07 | 10 | 0.9630 |
| 1.00E-07 | 100 | 0.9663 |
| 1.00E-05 | 1 | 0.9734 |
| 1.00E-05 | 10 | 0.9727 |
| 1.00E-05 | 100 | 0.9790 |
| 0.0001 | 1 | 0.9758 |
| 0.0001 | 10 | 0.9779 |
| 0.0001 | 100 | 0.9785 |
| 0.001 | 1 | 0.9749 |
| 0.001 | 10 | 0.9749 |
| 0.001 | 100 | 0.9753 |
| 0.01 | 1 | 0.9715 |
| 0.01 | 10 | 0.9720 |
| 0.01 | 100 | 0.9720 |
| 0.1 | 1 | 0.8928 |
| 0.1 | 10 | 0.8958 |
| 0.1 | 100 | 0.8958 |
| 1 | 1 | 0.7981 |
| 1 | 10 | 0.8096 |
| 1 | 100 | 0.8096 |

| | | |
|---|---|---|
| 10 | 1 | 0.7399 |
| 10 | 10 | 0.7400 |
| 10 | 100 | 0.7400 |

Table 5.1 shows the variation of accuracy with C and γ parameter for SVM. The results show that γ parameter has high effect on accuracy than C parameter of RBF SVM. Though high value of C improves the accuracy, it increases the computational complexity. Therefore, value of γ and C are choosen to be 0.0001 and 1 respectively.

### 5.1.2. Metrics Calculation

**Table 5. 2 Calculation of metrics for Naive Bayes**

| Folds | TN | FP | FN | TP | Accuracy | Precision | Recall | F1-Score |
|---|---|---|---|---|---|---|---|---|
| 0 | 7817 | 0 | 895 | 4646 | 0.9330 | 1.0000 | 0.8385 | 0.9121 |
| 1 | 7816 | 1 | 895 | 4646 | 0.9329 | 0.9998 | 0.8385 | 0.9121 |
| 2 | 7665 | 152 | 144 | 5397 | 0.9778 | 0.9726 | 0.9740 | 0.9733 |
| 3 | 7799 | 18 | 1361 | 4180 | 0.8968 | 0.9957 | 0.7544 | 0.8584 |
| 4 | 7642 | 175 | 0 | 5541 | 0.9869 | 0.9694 | 1.0000 | 0.9845 |
| 5 | 7816 | 1 | 895 | 4646 | 0.9329 | 0.9998 | 0.8385 | 0.9121 |
| 6 | 7745 | 72 | 232 | 5309 | 0.9772 | 0.9866 | 0.9581 | 0.9722 |
| 7 | 7752 | 65 | 78 | 5463 | 0.9893 | 0.9882 | 0.9859 | 0.9871 |
| 8 | 6143 | 1673 | 51 | 5490 | 0.8709 | 0.7664 | 0.9908 | 0.8643 |
| 9 | 7756 | 60 | 196 | 5345 | 0.9808 | 0.9889 | 0.9646 | 0.9766 |
| **Average** | | | | | 0.9479 | 0.9667 | 0.9143 | 0.9353 |

Table 5.2 shows the calculation of Accuracy, Precision, Recall and F1-Score at each fold of 10-fold cross validation for Naïve Bayes classifier. The average values of Accuracy, Precision, Recall and F1-Score was found to be 0.9479, 0.9667, 0.9143 and 0.9353 respectively.

**Table 5. 3 Calculation of metrics for SVM**

| Folds | TN | FP | FN | TP | Accuracy | Precision | Recall | F1-Score |
|---|---|---|---|---|---|---|---|---|
| 0 | 7792 | 25 | 445 | 5096 | 0.9648 | 0.9951 | 0.9197 | 0.9559 |
| 1 | 7799 | 18 | 11 | 5530 | 0.9978 | 0.9968 | 0.9980 | 0.9974 |
| 2 | 7754 | 63 | 306 | 5235 | 0.9724 | 0.9881 | 0.9448 | 0.9660 |
| 3 | 7814 | 23 | 201 | 5317 | 0.9832 | 0.9957 | 0.9636 | 0.9794 |
| 4 | 7727 | 90 | 0 | 5541 | 0.9933 | 0.9840 | 1.0000 | 0.9919 |
| 5 | 7785 | 32 | 25 | 5516 | 0.9957 | 0.9942 | 0.9955 | 0.9949 |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 6 | 7771 | 46 | 215 | 5326 | 0.9805 | 0.9914 | 0.9612 | 0.9761 |
| 7 | 7797 | 120 | 762 | 4679 | 0.9340 | 0.9750 | 0.8600 | 0.9139 |
| 8 | 7214 | 602 | 52 | 5489 | 0.9510 | 0.9012 | 0.9906 | 0.9438 |
| 9 | 7745 | 71 | 111 | 5430 | 0.9864 | 0.9871 | 0.9800 | 0.9835 |
| **Average** | | | | | 0.9759 | 0.9809 | 0.9613 | 0.9703 |

Table 5.3 shows the calculation of Accuracy, Precision, Recall and F1-Score at each fold of 10-fold cross validation for SVM classifier. The average values of Accuracy, Precision, Recall and F1-Score was found to be 0.9759, 0.9809, 0.9613 and 0.9703 respectively.

**Table 5. 4 Calculation of metrics for Hybrid Algorithm**

| Folds | TN | FP | FN | TP | Accuracy | Precision | Recall | F1-Score |
|---|---|---|---|---|---|---|---|---|
| 0 | 7816 | 1 | 0 | 5541 | 0.9999 | 0.9998 | 1.0000 | 0.9999 |
| 1 | 7804 | 13 | 493 | 5048 | 0.9621 | 0.9974 | 0.9110 | 0.9523 |
| 2 | 7733 | 84 | 166 | 5375 | 0.9813 | 0.9846 | 0.9700 | 0.9773 |
| 3 | 7799 | 18 | 234 | 5307 | 0.9811 | 0.9966 | 0.9578 | 0.9768 |
| 4 | 7695 | 122 | 0 | 5541 | 0.9909 | 0.9785 | 1.0000 | 0.9891 |
| 5 | 7806 | 11 | 20 | 5521 | 0.9977 | 0.9980 | 0.9964 | 0.9972 |
| 6 | 7746 | 71 | 218 | 5323 | 0.9784 | 0.9868 | 0.9607 | 0.9736 |
| 7 | 7753 | 64 | 525 | 5016 | 0.9559 | 0.9874 | 0.9053 | 0.9445 |
| 8 | 7325 | 491 | 51 | 5490 | 0.9594 | 0.9179 | 0.9908 | 0.9530 |
| 9 | 7741 | 75 | 112 | 5429 | 0.9860 | 0.9864 | 0.9798 | 0.9831 |
| **Average** | | | | | 0.9793 | 0.9833 | 0.9672 | 0.9747 |

Table 5.4 shows the calculation of Accuracy, Precision, Recall and F1-Score at each fold of 10-fold cross validation for Hybrid Algorithm. The average values of Accuracy, Precision, Recall and F1-Score was found to be 0.9793, 0.9833, 0.9672 and 0.9747 respectively.

## 5.2. Analysis

### 5.2.1. Model Selection for SVM

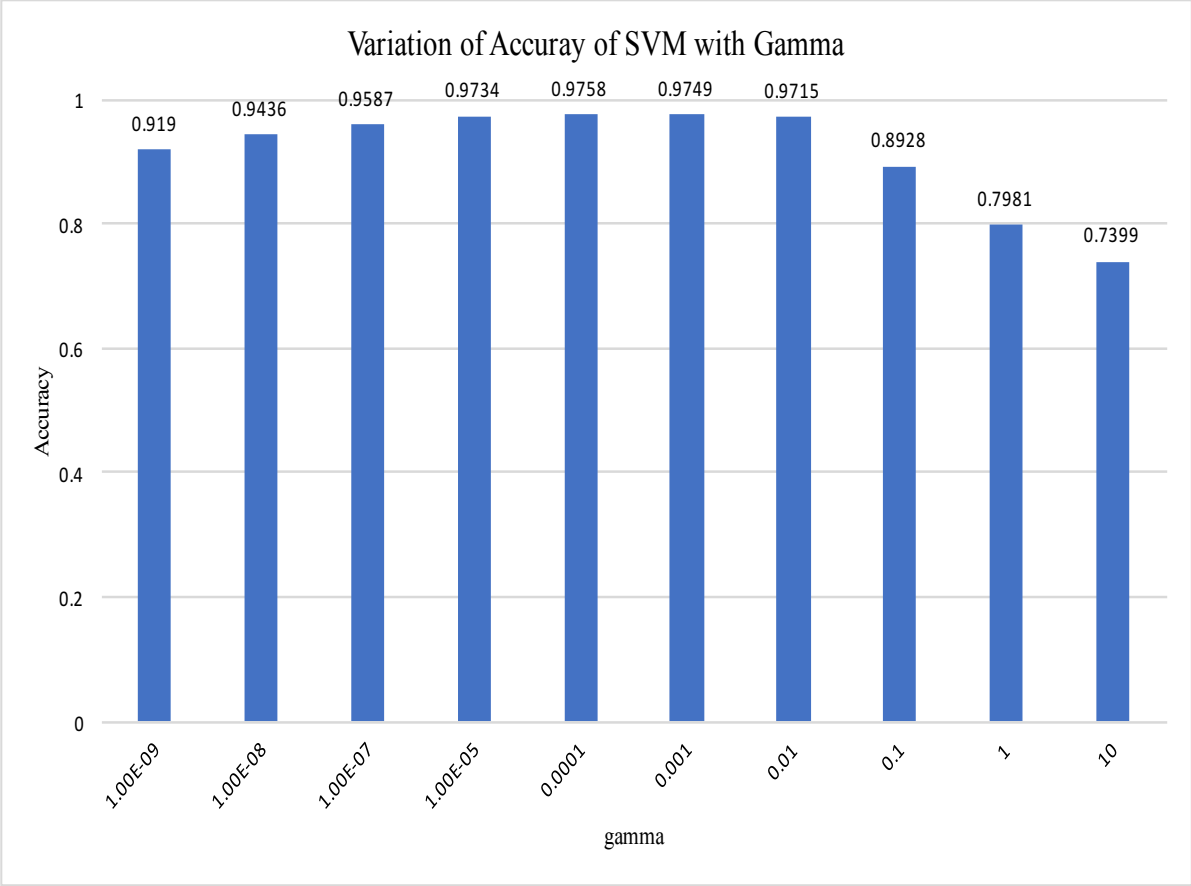


**Figure 5. 1 Effect of gamma on accuracy of SVM classifier.**

Figure 5.1 shows the variation of accuracy with the value of gamma(γ). The value of gamma is choosen to be exponentially growing sequence in the range 1e-9 to 10. The maximum accuracy of 0.9758 is obtained for value of γ=0.0001. To compute the accuracy, the average of accuracy from each fold of 10-fold cross validation was taken.

### 5.2.2. ROC Curve

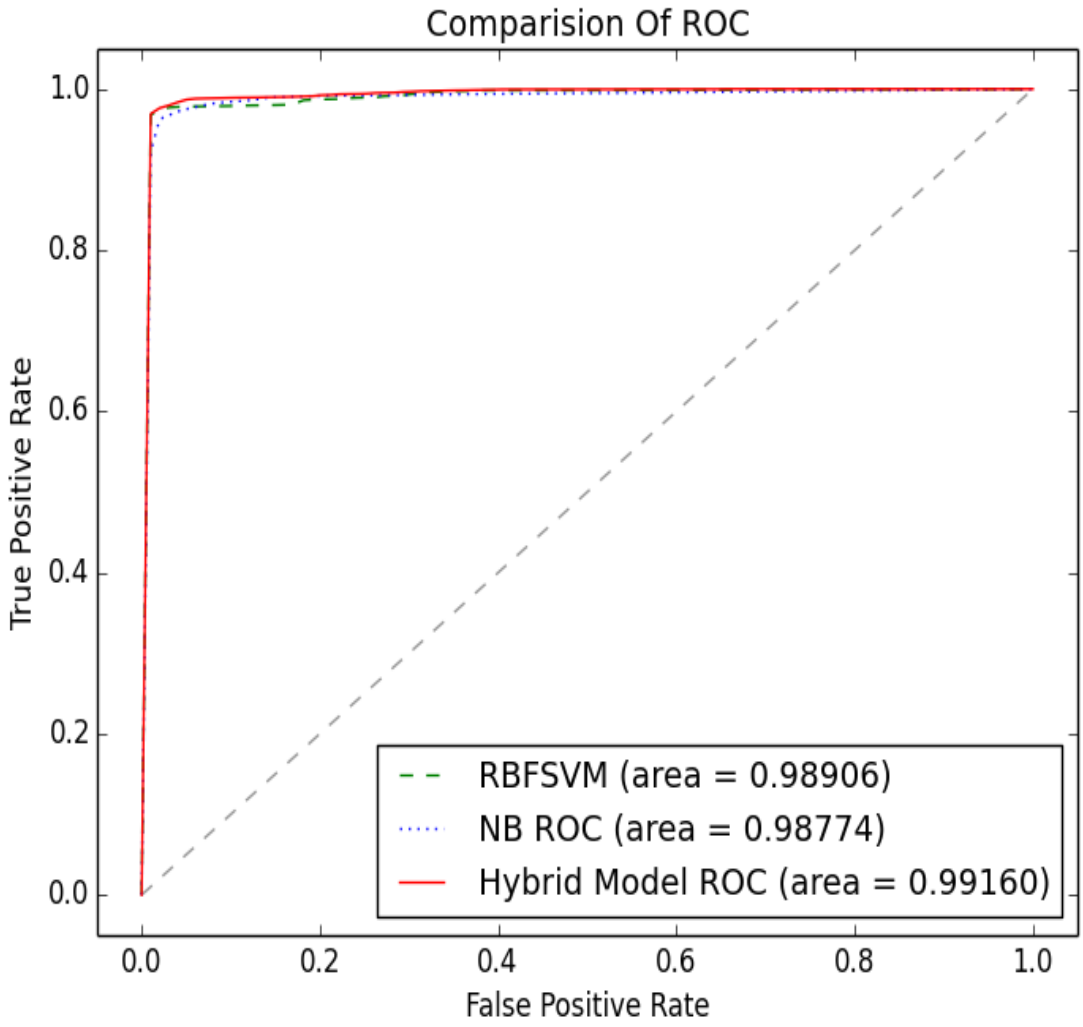


**Figure 5. 2 Mean ROC curve for SVM, Naive Bayes and Hybrid algorithm.**

Figure 5.2 shows the comparison of mean roc plot for SVM, Naïve Bayes and Hybrid algorithm. The mean area under curve for SVM, Gaussian Naïve Bayes and Hybrid algorithm was found to be 0.98906, 0.98774 and 0.99160 respectively. As higher the value of area under curve, better the classifier. Hybrid algorithm is better than SVM and Gaussian Naïve Bayes classifier as it has higher area under curve value than SVM and Naïve Bayes classifier.

### 5.2.3. Classification Metrics

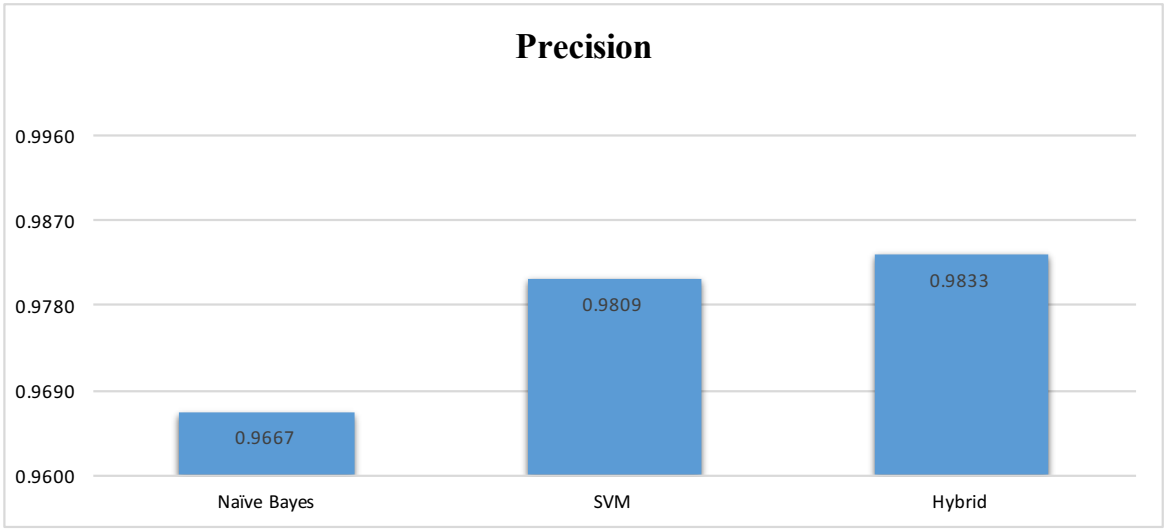


**Figure 5. 3 Average precision of SVM, Naïve Bayes and Hybrid Algorithm.**

Figure 5.3 shows the comparison of average precision of SVM, Naïve Bayes and Hybrid Algorithm using 10-fold cross validation. The average value of precision for SVM, Naïve Bayes and Hybrid Algorithm was found to be 0.9809, 0.9667 and 0.9833 respectively. The results showed that Hybrid algorithm has better precision than SVM and Naïve Bayes.

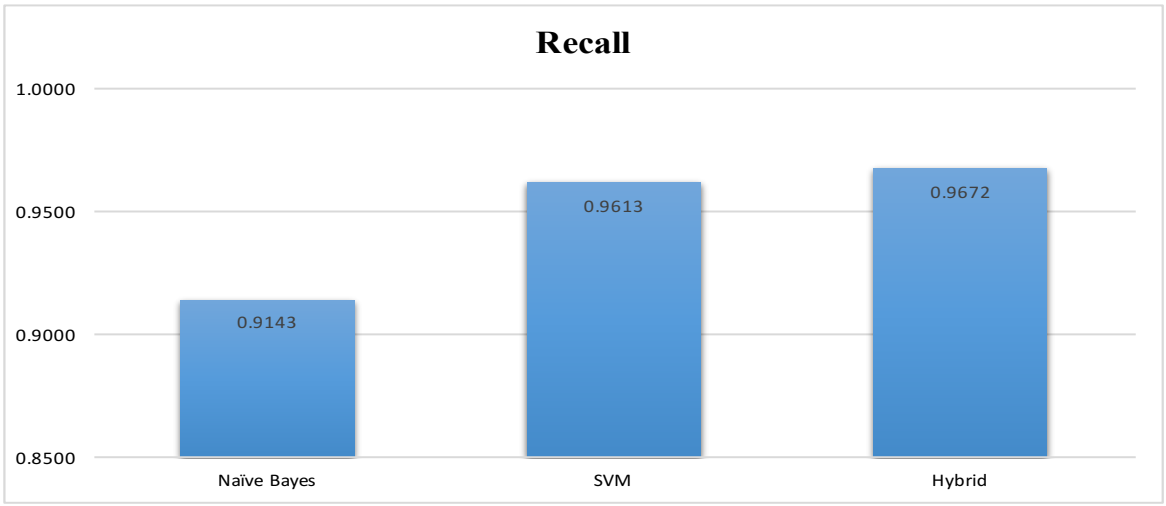


**Figure 5. 4 Average Recall of SVM, Naïve Bayes and Hybrid Algorithm.**

Figure 5.4 shows the comparison of average Recall of SVM, Naïve Bayes and Hybrid Algorithm using 10-fold cross validation. The average value of Recall for SVM, Naïve Bayes and Hybrid Algorithm was found to be 0.9613, 0.9143 and 0.9672 respectively. The results showed that Hybrid algorithm has better Recall than SVM and Naïve Bayes.

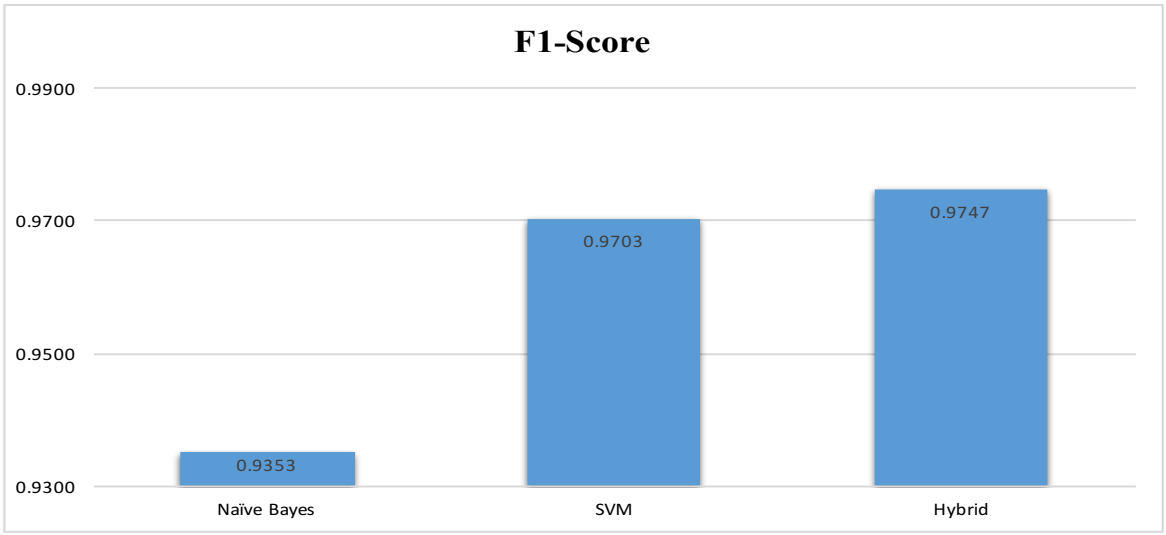


**Figure 5. 5 Average F1-Score of SVM, Naïve Bayes and Hybrid Algorithm.**

Figure 5.5 shows the comparison of average F1-Score of SVM, Naïve Bayes and Hybrid Algorithm using 10-fold cross validation. The average value of F1-Score for SVM, Naïve Bayes and Hybrid Algorithm was found to be 0.9703, 0.9353 and 0.9747 respectively. The results showed that Hybrid algorithm has better F1-Score than SVM and Naïve Bayes.

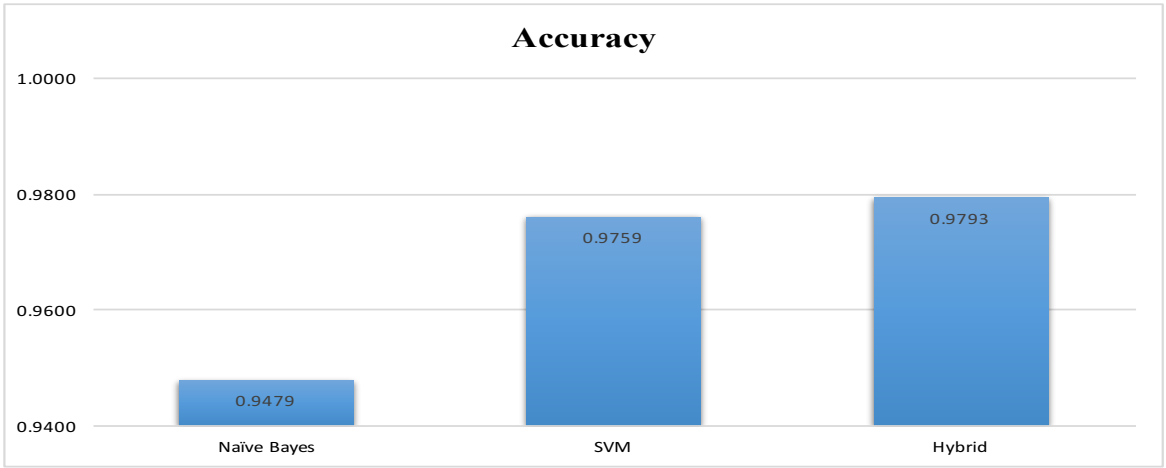


**Figure 5. 6 Average Accuracy of SVM, Naïve Bayes and Hybrid Algorithm.**

Figure 5.6 shows the comparison of average Accuracy of SVM, Naïve Bayes and Hybrid Algorithm using 10-fold cross validation. The average value of Accuracy for SVM, Naïve Bayes and Hybrid Algorithm was found to be 0.9759, 0.9479 and 0.9793 respectively. The results showed that Hybrid algorithm has better Accuracy than SVM and Naïve Bayes.

### 5.2.4. Computational Complexity

The SVM classifier has time complexity $O(pN^3)$, and the time complexity for Naïve Bayes classifier is $O(pN)$, where N is the number of training examples and $p$ is the number of features. The time complexity to compute weights is $O(N)$.

$$Total\ computational\ complexity = O(pN^3) + O(pN) + O(N) = O(pN^3)$$

## 5.3. Comparison

**Table 5. 5 Comparison of SVM, Naive Bayes and Hybrid algorithm.**

| SN | Parameter | SVM | Naïve Bayes | Hybrid Algorithm |
|---|---|---|---|---|
| 1 | ROC AUC | 0.9891 | 0.9877 | 0.9916 |
| 2 | Accuracy | 0.9759 | 0.9479 | 0.9793 |
| 3 | Precision | 0.9809 | 0.9667 | 0.9833 |
| 4 | Recall | 0.9613 | 0.9143 | 0.9672 |
| 5 | F1-Score | 0.9703 | 0.9353 | 0.9747 |
| 6 | Computational Complexity | $O(pN^3)$ | $O(pN)$ | $O(pN^3)$ |

Table 5.5 shows the comparison of SVM, Naïve Bayes and Hybrid Model based on AUC, classification metrics and computational complexity. The results show that Hybrid algorithm has improved performance upon AUC and classification metrics. In addition to this, the computational complexity of SVM and Hybrid algorithm are comparable and higher than that of Naïve Bayes.

# CHAPTER 6: CONCLUSION

This research work proposed a novel Hybrid Algorithm for the anomaly detection using SVM and Naïve Bayes, which employed the k-fold cross validation and error based weighting schemes. By combining the predictions of all member algorithms, the Hybrid algorithm achieves better performance in anomaly detection compared to any sole member algorithm.

The results of the hybrid algorithm are compared with the results of SVM and Naïve Bayes in terms of classification metrics and ROC curve. It is seen that hybrid approach outperforms SVM and Naïve Bayes. The new approach is effective during detection of anomalies. The detection ratio of the hybrid algorithm is better than other techniques. The hybrid algorithm properly classifies the data either as normal or abnormal. Therefore, it can be concluded that this hybrid approach is simple and efficient in terms of reducing the false alarm ratio.

There are many possibilities to exploit and extend the learning approach used in this thesis. This research work uses the information gain to select the important features from the KDD dataset. Many techniques for feature selection has been proposed by researchers. Other feature selection techniques can be employed to improve the performance of classifier. The model selection for SVM can be improved by choosing the values of C and $\gamma$ such that the grid size is small.

The hybrid algorithm has been applied to 10% KDD dataset. The performance of proposed hybrid algorithm can be employed to many new datasets like yahoo anomaly detection dataset, http csic dataset, etc. The hybrid algorithm can be employed to detect anomaly in IT enterprise after preparation of labeled dataset using real scenarios of anomalies. Moreover, hybrid model can be further tuned to improve accuracy with ensemble of more improved classifier.

# CHAPTER 7: LIMITATIONS

The proposed hybrid algorithm is weighted ensemble of SVM and Naïve Bayes. Therefore, the model suffers from limitations of SVM and Naïve Bayes.

The finding of right kernel for SVM is a major challenge. The kernel models can be quite sensitive to over-fitting the model selection criterion. The SVM has no standardized way for dealing with multi-class problems, fundamentally SVM is a binary classifier. Likewise, in many classification problems the probability of class membership is required, so it would be better to use a method like Kernel Logistic Regression, rather than post-process the output of the SVM to get probabilities. Although SVMs have good generalization performance, they can be slow in training as well as test phase. However, from a practical point of view perhaps the most serious problem with SVMs is the high algorithmic complexity and extensive memory requirements of the required quadratic programming in large-scale tasks.

The Naïve Bayes has very simple representation doesn't allow for rich hypotheses. Moreover, assumption of independence of attributes is too constraining for Naïve Bayes. Another problem happens due to data scarcity. For any possible value of a feature, you need to estimate a likelihood value by a frequency approach. This can result in probabilities going towards 0 or 1, which in turn leads to numerical instabilities and worse results.

# REFERENCES


[1] Chandola, Varun, Arindam Banerjee, and Vipin Kumar. "Anomaly detection: A survey." *ACM computing surveys (CSUR)* 41.3 (2009): 15.

[2] Jyothsna, V., VV Rama Prasad, and K. Munivara Prasad. "A review of anomaly based intrusion detection systems." *International Journal of Computer Applications* 28.7 (2011): 26-35.

[3] Mukkamala, Srinivas, Guadalupe Janoski, and Andrew Sung. "Intrusion detection using neural networks and support vector machines." *Neural Networks, 2002. IJCNN'02. Proceedings of the 2002 International Joint Conference on*. Vol. 2. IEEE, 2002.

[4] Tavallaee, Mahbod, et al. "A detailed analysis of the KDD CUP 99 data set."*Proceedings of the Second IEEE Symposium on Computational Intelligence for Security and Defence Applications 2009*. 2009.

[5] Sung, Andrew H., and Srinivas Mukkamala. "Identifying important features for intrusion detection using support vector machines and neural networks."*Applications and the Internet, 2003. Proceedings. 2003 Symposium on*. IEEE, 2003.

[6] Mukkamala, Srinvas, A. H. Sung, and B. M. Ribeiro. "Model selection for kernel based intrusion detection systems." *Adaptive and Natural Computing Algorithms*. Springer Vienna, 2005. 458-461.

[7] Chauhan, Himadri, et al. "Comparative Analysis and Research Issues in Classification Techniques for Intrusion Detection." *Intelligent Computing, Networking, and Informatics*. Springer India, 2014. 675-685.

[8] Om, Hari, and Aritra Kundu. "A hybrid system for reducing the false alarm rate of anomaly intrusion detection system." *Recent Advances in Information Technology (RAIT), 2012 1st International Conference on*. IEEE, 2012.

[9] Mukkamala, Srinivas, and Andrew H. Sung. "Significant feature selection using computational intelligent techniques for intrusion detection." *Advanced Methods for Knowledge Discovery from Complex Data*. Springer London, 2005. 285-306.

[10] Chebrolu, Srilatha, Ajith Abraham, and Johnson P. Thomas. "Feature deduction and ensemble design of intrusion detection systems." *Computers & Security* 24.4 (2005): 295-307.

[11] Chen, Yuehui, Ajith Abraham, and Bo Yang. "Feature selection and classification using flexible neural tree." *Neurocomputing* 70.1 (2006): 305-313.

[12] Amiri, Fatemeh, et al. "Mutual information-based feature selection for intrusion detection systems." *Journal of Network and Computer Applications*34.4 (2011): 1184-1199.

[13] Horng, Shi-Jinn, et al. "A novel intrusion detection system based on hierarchical clustering and support vector machines." *Expert systems with Applications* 38.1 (2011): 306-313.

[14] Heller, Katherine A., et al. "One class support vector machines for detecting anomalous windows registry accesses." *Proc. of the workshop on Data Mining for Computer Security*. Vol. 9. 2003.

[15] Zhang, Ming, Boyi Xu, and Jie Gong. "An Anomaly Detection Model Based on One-Class SVM to Detect Network Intrusions." *2015 11th International Conference on Mobile Ad-hoc and Sensor Networks (MSN)*. IEEE, 2015.

[16] Chitrakar, Roshan, and Huang Chuanhe. "Anomaly detection using Support Vector Machine classification with k-Medoids clustering." *2012 Third Asian Himalayas International Conference on Internet*. IEEE, 2012.

[17] Amor, Nahla Ben, Salem Benferhat, and Zied Elouedi. "Naive bayes vs decision trees in intrusion detection systems." *Proceedings of the 2004 ACM symposium on Applied computing*. ACM, 2004.

[18] Jain, Rahul, Tejpal Singh, and Amit Sinhal. "A SURVEY ON NETWORK ATTACKS, CLASSIFICATION AND MODELS FOR ANOMALY-BASED NETWORK INTRUSION DETECTION SYSTEMS." (2013).

[19] Murtaza, Syed Shariyar, et al. "A host-based anomaly detection approach by representing system calls as states of kernel modules." *2013 IEEE 24th International Symposium on Software Reliability Engineering (ISSRE)*. IEEE, 2013.

[20] Manikopoulos, Constantine, and Symeon Papavassiliou. "Network intrusion and fault detection: a statistical anomaly approach." *IEEE Communications Magazine* 40.10 (2002): 76-82.

[21] Xiao, Yingchao, Huangang Wang, and Wenli Xu. "Parameter selection of Gaussian kernel for one-class SVM." *IEEE transactions on cybernetics* 45.5 (2015): 941-953.

[22] Gaikwad, D. P., and Ravindra C. Thool. "Intrusion Detection System Using Bagging Ensemble Method of Machine Learning." *Computing Communication Control and Automation (ICCUBEA), 2015 International Conference on*. IEEE, 2015.

[23] Hoglund, Albert J., Kimmo Hatonen, and Antti S. Sorvari. "A computer host-based user anomaly detection system using the self-organizing map." *Neural Networks, 2000. IJCNN 2000, Proceedings of the IEEE-INNS-ENNS International Joint Conference on*. Vol. 5. IEEE, 2000.

[24] Thaseen, I. Sumaiya, and Ch Aswani Kumar. "Intrusion detection model using fusion of PCA and optimized SVM." *Contemporary Computing and Informatics (IC3I), 2014 International Conference on*. IEEE, 2014.

[25] Lin, Liyu, et al. "SVM ensemble for anomaly detection based on rotation forest." *Intelligent Control and Information Processing (ICICIP), 2012 Third International Conference on*. IEEE, 2012.

[26] Peddabachigari, Sandhya, et al. "Modeling intrusion detection system using hybrid intelligent systems." *Journal of network and computer applications*30.1 (2007): 114-132.

[27] Amudha, P., S. Karthik, and S. Sivakumari. "Intrusion detection based on Core Vector Machine and ensemble classification methods." *Soft-Computing and Networks Security (ICSNS), 2015 International Conference on*. IEEE, 2015.

[28] Sornsuwit, Ployphan, and Saichon Jaiyen. "Intrusion detection model based on ensemble learning for U2R and R2L attacks." *2015 7th International Conference on Information Technology and Electrical Engineering (ICITEE)*. IEEE, 2015.

[29] Hu, Chao, Byeng D. Youn, and Pingfeng Wang. "Ensemble of data-driven prognostic algorithms with weight optimization and k-fold cross validation."*ASME 2010 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. American Society of Mechanical Engineers, 2010.

[30] Zerpa, Luis E., et al. "An optimization methodology of alkaline–surfactant–polymer flooding processes using field scale numerical simulation and multiple surrogates." *Journal of Petroleum Science and Engineering* 47.3 (2005): 197-208.

[31] Guang, Yang, and Nie Min. "Anomaly intrusion detection based on wavelet kernel LS-SVM." *Computer Science and Network Technology* (ICCSNT), 2013 3rd International Conference on. IEEE, 2013.

[32] Pervez, Muhammad Shakil, and Dewan Md Farid. "Feature selection and intrusion classification in NSL-KDD cup 99 dataset employing SVMs."*Software, Knowledge, Information Management and Applications (SKIMA), 2014 8th International Conference on*. IEEE, 2014.

[33] Mukherjee, Saurabh, and Neelam Sharma. "Intrusion detection using naive Bayes classifier with feature reduction." *Procedia Technology* 4 (2012): 119-128.

[34] Hodge, Victoria J., and Jim Austin. "A survey of outlier detection methodologies." *Artificial intelligence review* 22.2 (2004): 85-126.

[35] Bolton, Richard J., and David J. Hand. "Statistical fraud detection: A review."*Statistical science* (2002): 235-249.

[36] Hill, David J., and Barbara S. Minsker. "Anomaly detection in streaming environmental sensor data: A data-driven modeling approach."*Environmental Modelling & Software* 25.9 (2010): 1014-1022.

[37] Bolton, Richard J., and David J. Hand. "Unsupervised profiling methods for fraud detection." *Credit Scoring and Credit Control VII* (2001): 235-255.

[38] Sample, Char, and Kim Schaffer. "An overview of anomaly detection." *IT Professional* (2013): 8-11.

[39] Hasan, Md Al Mehedi, et al. "Support vector machine and random forest modeling for intrusion detection system (IDS)." *Journal of Intelligent Learning Systems and Applications* 6.1 (2014): 45.

[40] Han, Jiawei, Jian Pei, and Micheline Kamber. *Data mining: concepts and techniques*. Elsevier, 2011.

[41] Vapnik, Vladimir Naumovich. "The nature of statistical learning theory, ser. Statistics for engineering and information science." *New York: Springer* 21 (2000): 1003-1008.

[42] Lin, Hsuan-Tien, Chih-Jen Lin, and Ruby C. Weng. "A note on Platt's probabilistic outputs for support vector machines." *Machine learning* 68.3 (2007): 267-276.

[43] Zhang, Harry. "The optimality of naive Bayes." *AA* 1.2 (2004): 3.

[44] Metsis, Vangelis, Ion Androutsopoulos, and Georgios Paliouras. "Spam filtering with naive bayes-which naive bayes?." *CEAS*. 2006.

[45] Kayacik, H. Günes, A. Nur Zincir-Heywood, and Malcolm I. Heywood. "Selecting features for intrusion detection: A feature relevance analysis on KDD 99 intrusion detection datasets." *Proceedings of the third annual conference on privacy, security and trust*. 2005.

[46] Calix, Ricardo A., and Rajesh Sankaran. "Feature Ranking and Support Vector Machines Classification Analysis of the NSL-KDD Intrusion Detection Corpus." *FLAIRS Conference*. 2013.

[47] Hsu, Chih-Wei, Chih-Chung Chang, and Chih-Jen Lin. "A practical guide to support vector classification." (2003): 1-16.

[48] Valentini, Giorgio, and Thomas G. Dietterich. "Bias-variance analysis of support vector machines for the development of SVM-based ensemble methods." *Journal of Machine Learning Research* 5.Jul (2004): 725-775.

[49] Li, Xuchun, Lei Wang, and Eric Sung. "AdaBoost with SVM-based component classifiers." *Engineering Applications of Artificial Intelligence* 21.5 (2008): 785-795.

[50] Lukashevich, Hanna, Stefanie Nowak, and Peter Dunker. "Using one-class SVM outliers detection for verification of collaboratively tagged image training sets." *2009 IEEE International Conference on Multimedia and Expo*. IEEE, 2009.