Improving the performance of IDS using improve feature selection method

Thesis Submitted in Partial Fulfillment of Requirements for the Award of the Degree
Of

Master of Technology
IN
INFORMATION SYSTEM

SUBMITTED BY
DEEPAK KUMAR
(2K16/ISY/03)

UNDER THE GUIDANCE OF
ANAMIKA CHAUHAN
ASSISTANT PROFESSOR

DEPARTMENT OF INFORMATION TECHNOLOGY DELHI TECHNOLOGICAL UNIVERSITY BAWANA ROAD, DELHI-110042 (2016-2018)

CERTIFICATE

This is to certify that Mr. **DEEPAK KUMAR (2K16/ISY/03)** has carried out the major project titled **"Improving the performance of IDS using improve feature selection method"** as a partial requirement for the award of **Master of Technology** degree in **Information System** by **Delhi Technological University, Delhi.**

The Major project is a bonafide piece of work carried out and completed under my supervision and guidance during the academic session 2016-2018. The Matter contained in this thesis has not been submitted elsewhere for the award of any other degree.

Date:

(Project Guide)

Ms. Anamika Chauhan

Assistant Professor

Department Of Information Technology

Delhi Technological University

ACKNOWLEDGEMENT

I express my gratitude to my major project guide **Ms. Anamika Chauhan, Assistant Professor** in **Department of Information technology** at **Delhi Technological University, Delhi** for the valuable support and guidance she provided in making this major project. It is my pleasure to record my sincere thanks to my respected guide for her constructive criticism and insight without which the project would not have shaped as it has.

I humbly extend my word of gratitude to Dr. Kapil Verma, Head of Department and other faculty members of department of Information Technology for providing their valuable help and time whenever it was required.

DEEPAK KUMAR

Roll No.: 2K16/ISY/03

M.Tech (Information System)

Department of Information Technology

Delhi Technological University, Delhi

ABSTRACT

Nowadays, the use of networks and especially the Internet has become a big part of daily life. According to rapid development and widespread use of network systems, diverse intrusive approaches have grown extensively in the recent years. Multiple protection techniques have been used in order to manage the security network risks. These methods do not suffice, as each of them have proven their inefficiency. Therefore, the use of intrusion detection systems as an additional defense mechanism is almost indispensable. An Intrusion Detection System (IDS) dynamically monitors the events taking place in a system, and decides whether these events are symptomatic of an attack (intrusion) or constitute a legitimate use of the system. Since the appearance of IDS multiple techniques have been proposed in order to improve the performances of these. Recently, several machine learning techniques and optimization techniques have been applied to make it efficient and to improve accuracy.

In this project I am using Hybrid Binary PSO with SVM to find best subset of dataset to train our prediction model, and using this I'm improving the performance of prediction model. In this project I'm also comparing the performance of different classification algorithms like SVM, Random Forest, and Naïve baye's.

LIST OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION

The tremendous development in communication technology has empowered the vision of consistent connectivity. This has provided for inter-connectivity between wired and wireless (remote) networks, infrastructure based and ad- hoc networks, thus Facilitate heterogeneous devices to communicate with each other. Further with the introduction of IoTs, all virtual and physical devices shall be able to connect and communicate. Though this has led to many opportunities for development, it also faces numerous challenges and short comings. With the number of devices and applications expanding exponentially, networking infrastructure has to deal with excessive amount of data traffic. This information will consist of helpful as well as malicious information. And with increased advantages the burden of increasing number of types of attacks and threats is also to be faced. This threat is manifold and must be mitigated with deployment of new security techniques. This techniques must ensure security of data as well as administrative and legitimate privileges of users. A large portion of the attacks are basically intrusions into the system in the form of malware, bots, viruses, worms and Trojans.

If security issues are not addressed then the confidential information might be leaked at any time. In this manner, the security issue must be address to.

Confidentiality: An attacker can easily intercept the information passing from sender to the receiver that results privacy can be leaked and content of the information can be modified. So that secure message passing is required to maintain confidentiality of user.

Integrity: The message must not be modified during transmission; it should be received at receiver side same as it is sent at sender node. Integrity guarantees that message has not been modified by unauthorized persons during transmission.

Availability: Information or resources must be available when required. Attackers can flood the channel to damage the availability. By using malicious attacks like Denial of service (DOS) attack, flooding attack, black hole attack, jamming attacks etc. Availability can be damage.

Authenticity: Authenticity involves proof of identity. Users should be able to identify each other's identity with which they are communicate. It can be verified through authentication process so the unauthorized user cannot participate in the communication.

Non-Repudiation: In Non- Repudiation sender node and receiver node both cannot deny having sent and received the message respectively.

With the huge growth in technology encourages researcher to create various security methods to make security of network non -vulnerable and to ensures the privacy and data of user from attackers but attackers comes with various different complex ideas to crack those mechanism. So we need to develop something different and complex and non-vulnerable security mechanism and security models so we can ensures individual privacy. Now this is possible with the concept of supervised learning. Using supervised learning or unsupervised learning or both (which is subset of Artificial Intelligence) techniques it's possible to develop.

With the use of machine learning technique's we can develop complex and non-vulnerable prediction model and train that model with available datasets. After making our prediction security model, it can dynamically monitor the data flows in network and alert for suspicious attack. But there is a limitation with machine learning based prediction, is false alarm. In past few years various machine learning algorithms and techniques have been developed to reduce the shortcomings of previous algorithms and to reduce false alarm.

For building a prediction model the data plays an important role. The data which we use for our prediction model is collected from various sources and after simulating various types of attacks in lab. The dataset is huge, it has various features but it not necessary that all the features are important. Some features in the dataset may be irrelevant for our model because it degrades the performance of security model and increases the false alarm rate. So before use of data for training purpose, we have to use data preprocessing techniques to remove the noise from data.
To select relevant data and to remove irrelevant and unnecessary data, we need feature selection techniques. Using feature selection methods we can select only those features which are required for training our classification algorithm to build intelligent security model. Using feature selection algorithms with nature inspired algorithms we can improve the performance of security model and improve their accuracy.

Chapter 2

RELATED THEORY

2.1. INTRUSION

Computer security can be exceptionally entangled and may be significantly puzzling to various people. It can even be a disputable subject. Network administrators had strong believe that their system is secure and complex and nobody can break-in and attackers who have past involvement of breaking into systems May get a kick out of the chance to trust that they can break into any sort of system. Thusly intrusion detection system gives a safe and protective wall to protect computer system inside system. The key components for intrusion detection are: system resources to be shielded in a target system from unauthorized access, i.e., accounts, system documents, kernel, and so on; models that determine the behavior of these assets as 'typical' or ‘legitimate’; strategies which we use to compare the normal system activities and the built up models, and distinguish those that are "irregular" or "suspicious". It is essential for us to construct a security instruments for system, which is intended to protect system resources and information from unauthorized access. We can, however, we can try to detect these unauthorized attempts so that necessary action may be taken to reduce the damage later. This area of research is called Intrusion Detection.

In simple words, monitoring the activities of user in the network and revealing user activity as normal or abnormal are referred as Intrusion Detection.

Sometimes intrusion detection system considered as obscene experimental, continuous research in network security space, intrusion detection system has reached to great height and it secures its place in network security domain along with threat protection system and firewalls. But in real scenario the implementation of intrusion detection system is a complex task and, sometimes its simplest one: examining all the incoming and outgoing packets through network and activities in network.

Classification of intrusions are discuss below

There are mainly six types of intrusions, they are listed below

- Attempted break-ins: It’s detected by unusual behavior or if there is violation of security rules.
- Masquerade attacks: in this type of attack, attacker pose as authentic user i.e. User profile seems suspicious or their will be security rules violates.
- Control system get penetrated: it’s detected by examining the specific activity patterns.
- Leakage: it’s detected if usage of resource seems suspicious.

- Denial of service: it’s detected, if system or server has resource but it’s unable to provide services for which it’s designs or its resources get blocked in unusual way.

2.2. INTRUSION DETECTION SYSTEM

Intrusion detection system (IDS) is a security tool, which is designed to perform automatically monitor of incoming and outgoing packets in networks to classify the packet is malicious or legitimate. If ids founds that the network packet is malicious then it alert the network administrator to take necessary actions and also maintain the log record.

By examining the availability of system vulnerabilities, checking integrity of system files, and to check with available attack patterns , an intrusion detection system detects the behaviors of system is normal or intrusive. It automatically monitors the internet by itself to search for any of newest attacks or threats which cause trouble or attack in future.

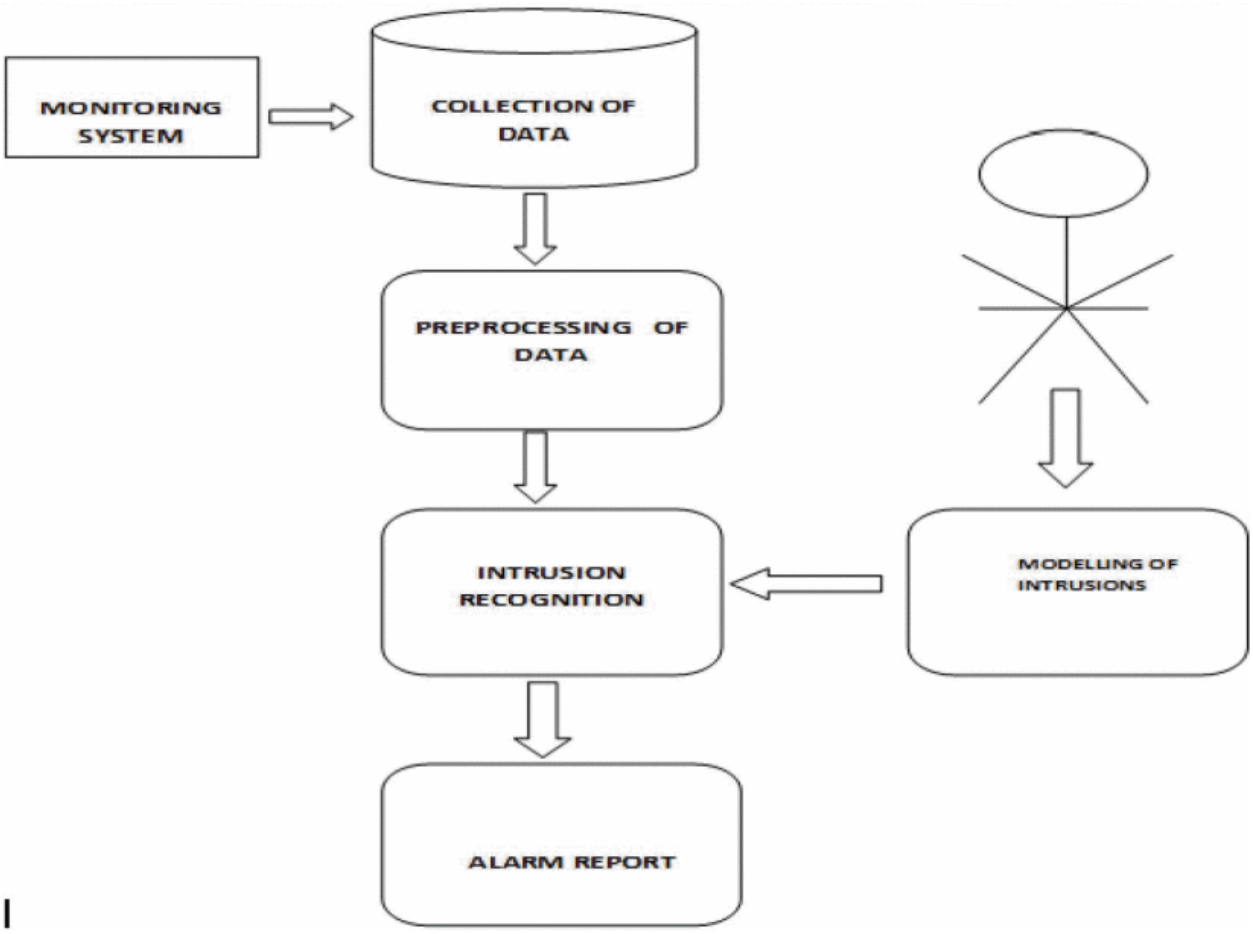


Fig 2.1. General architecture of IDS

Working of Intrusion detection systems

Steps for generalizing the working of IDS:

- Collection of Data: It includes a collection of network activity using particular software and it helps to collect the information about the network traffic like sorts of packets, hosts and protocol details.
- Selection of Feature: Data which is been gathered, significantly in a huge amount because of the excessive network traffic ratio, that contains just necessary information, like packet type, layer 4 protocol type, etc.
- Data Analysis: The data is been analyzed in this step to discover whether data is abnormal or not.
- Action Performed: IDS alarm or alert is been made by the system administrator to alert if an attack has occurred and it tells about the character of the attack. IDS also participate in controlling the attacks by closing the network access and destroy the processes.

There are various ways for detection of malicious activity in network by an IDS. Based on their detection approach, the intrusion detection system techniques grouped into three types, they are discussed below.

1. SIGNATURE BASED IDS (SIDS) :

   A Signature based IDS has a knowledge repository of known patterns or signature of known threats that are provided by human experts. SIDS will monitor the network for incoming and outgoing packets and compare them against known repository, if the pattern matched then it confirms the attack. Decisive advantage of signature based Intrusion detection system is already known attack pattern is stored, the future state of intrusion can be efficiently and effectively recognized.
   The main disadvantage is it can’t detect the new threat.

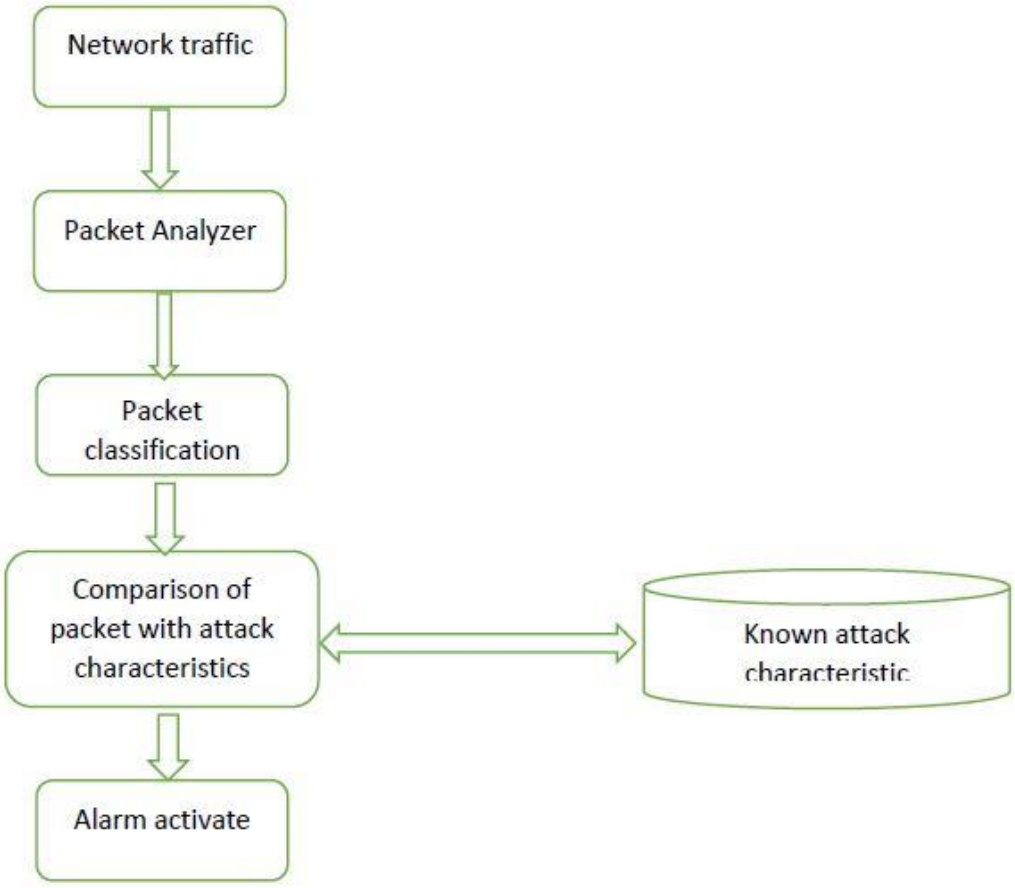


Fig 2.2. Signature based IDS

2. ANOMALY BASED IDS (AIDS): In Anomaly based IDS the system builds a normal behavior profile based on available data. AIDS compares the user activity with the build pattern, if the difference is more than acceptable, user activity will be consider as intrusive or malicious. The major advantage of Anomaly based IDS is they are able to detect unknown intrusion which is new for IDS and it doesn't require prior knowledge of a particular intrusion.

3. SPECIFICATION BASED IDS: This strategy is fairly like anomaly detection system. In this technique, the normal behavior of the network is characterized by physically, so it provides less incorrect positives rate. This method endeavors to selection best between signature-based and anomaly based detection systems by endeavoring to clear up deviations from normal behavior of conduct that are made neither by the training information nor by the machine learning techniques. The improvement of attack or protocol

specification is finished by physically so it requires greater investment. Thus, this can be a drawback of this approach.

Based on its data storage and analysis, IDS can be categorize into two parts, they are discussed below:

1. HOST-BASED IDS (HIDS): HIDS referred as host based intrusion detection method in which an IDS runs on a single host and monitor the incoming and outgoing network traffic for suspicious activity. In this approach system collects data as records of various activities including event logs, system logs, memory usage etc. For working of Host based IDS, we don't need to install any kind of hardware or additional software. The main advantage of HIDS is it confirms that the attack is successful or failed, monitors system activity, detects attacks. We cannot see the near real-time detection and response in network based IDS.

2. NETWORK BASED IDS (NIDS): NIDS are placed at a strategic point or somewhere within the network so that it can monitor traffic to and from all devices on the network. Ideally, NIDS will be placed in network were we can monitor all inbound and outbound traffic. The main disadvantage of NIDS is high false alarm rate. Mostly NIDS is OS independent and their implementation are easy.

The effectiveness of an IDS depends on data and classification algorithm. Ids collects and analyze the information from different areas and deals with large amount of data. Data with irrelevant and redundant features reduces the performance of IDS. For that reason, data preprocessing is applied on data to reduce the errors and to select meaningful features in data to improve the accuracy of IDS.
Selection of classification algorithm is also important. During selection process of an algorithm, it's very important to check for few parameters like accuracy and resources consumption of an algorithm etc.

2.3. FEATURE SELECTION

In machine learning and data mining areas Feature selection is a critical preprocessing strategy. This method can be utilized not only to reduce the amount of data for analysis but also to build prediction models which has more accuracy and with more grounded inter-operability based on lesser features. Feature selection is a process of selecting a subset of whole features according to certain set of rules, is a critically important and mostly used as a reduction technique in data

mining. A normally used feature section algorithm or techniques consist of four stages, which are subset generation, subset evaluation, stopping criterion, and result validation.

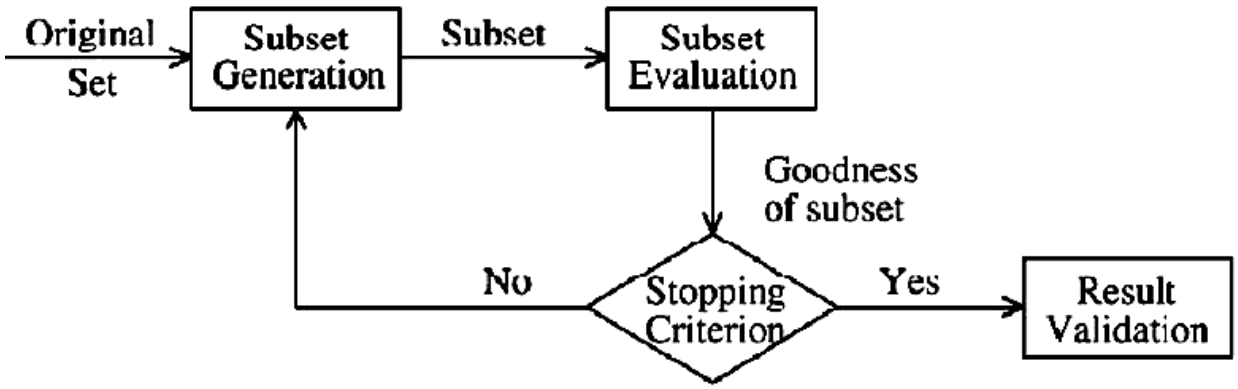


Fig 2.3. Feature selection framework for IDS

In subset generation we use a search method which generates candidate feature subset for analysis based on some search techniques. Every single candidate feature subset is examined and compared with previous best candidate feature subset according to a certain examine criteria. If new candidate feature subset is better in evaluation criterion, the previous one is replaced by this new one. This process of subset generation is repeated until we find best feature subset or defined stopping criteria is meet.

Feature selection is very important technique used to enhance the performance of machine learning algorithms. Its importance is discussed below:

Machine learning follow a simple rule- on the off chances that you place junk in, you will just inspire waste to turn out. Noise in information is referred as junk.

This turns out to be considerably more critical when there is huge number of features. You require not utilize each feature available to you for making a prediction. You can help your prediction by sustaining in just those features that are extremely relevant. In various research work i have seen that relevant feature subset gives you better prediction than all available features for same algorithm.

In the rivalries as well as this can be exceptionally helpful in modern applications also. You not just reduces the time required for training and the assessment time, you likewise have less things to stress over!

Top motivations to utilize feature selection are:

- Feature selection helps machine learning algorithm to prepare quicker.
- It lessens the intricacy of a model and makes it less demanding to translate.
- It enhances the precision of a model if the correct subset is picked.
- It diminishes over fitting.

FILTER METHOD

Filter methods are generally used for the performance evaluation of the selected features.
It's used as a preprocessing step. The selection of feature subset is independent of any other machine learning algorithms.
Instead, selection of features are done on the basis of their performance in various statistical tests and keeps or remove low performance features from the available dataset.
Filter methods:

- Information gain
- Chi-square test
- Fisher score
- Correlation coefficient
- Variance threshold

WRAPPER METHOD

Wrapper method selects a set of features, prepares different combination, evaluate each one of them and compares with each other.
This method can provide the best subset of features but it's computationally expensive. There are some
Common examples of wrapper method:

- FORWARD SELECTION: It's a repetitive strategy. As named suggest we move forward to add features which are boost the performance of classifier or machine learning algorithms. In this methodology we begin with an empty feature subset. In each cycle we include features which improve the accuracy and performance of prediction model till an

expansion of another feature doesn't improve the accuracy and performance of prediction model.

- BACKWARD SELECTION: It's a repetitive strategy. As named suggest we move backward to remove features so the remaining features in feature subset boost the performance of classifier or machine learning algorithms. In this methodology we begin with all available feature subset. In each cycle we exclude features which improve the accuracy and performance of prediction model till an inclusion of another feature doesn't improve the accuracy and performance of prediction model.

- RECURSIVE FEATURE SELECTION: It is a voracious optimization algorithm which intends to find the best performing feature subset. It literately makes models and keeps aside the best or the most observably awful performing features at each accentuation. It builds up the accompanying model with the left feature until the point that each one of the features are depleted. It at that point positions the features in light of the request of their disposal.

Pros and cons of wrapper method are discuss below

Pros: feature subset search and model selection phases interacting very well and it has abilities to check for feature dependencies.

Cons: it has higher risk of feature over-fitting than filter methods. Wrapper method is very computationally intensive especially when computation cost for building classifier is expensive.

EMBEDDED METHOD

Embedded method has its characteristics comes from filter method and wrapper methods, because in embedded method we use some characteristics of both and its implementation is done by those algorithms which has their own feature selection methods libraries.

LASSO AND RIDGE regression are some of the most used examples. LASSO and RIDGE regression have its own built-in 'penalization function' for reducing over-fitting.

- To perform L1 regularization, LASSO regression used. It adds penalty which is equivalent to the value of coefficients magnitude.
- To perform L2 regularization, Ridge regression is used. It adds penalty which is equivalent to the square of value of coefficients magnitude.

Difference between filter and wrapper methodologies are:

As we have earlier discussed the filter and wrapper methods, now the major differences between them are discussed below

- In filter strategy we measure significance of features by checking their connection with dependent variable however in wrapper technique we measure the significance of feature subset via preparing a model on it. The performance of filter method is faster than the performance of wrapper methods because in filter method we doesn't train model. On other hand, wrapper methods are very expensive in terms of computation.
- In filter method we use statistical techniques to evaluate the importance of features while in wrapper method we evaluate it by cross validation.
- In filter method, it's not always possible to find best feature subset but in wrapper method, it always guarantee to provide best possible feature subset.
- Using features subset generated by wrapper methods for training model leads over-fitting problem but the feature subset generated by filter methods make model less prune.

2.4. NSL-KDD DATASET

As discussed before, we utilize NSL-KDD dataset in our work. The dataset is an enhanced and reduced version of the "KDD Cup 99 dataset" The KDD Cup dataset was captured utilizing the network traffic captured by 1998 DARPA IDS assessment program. The network traffic incorporates typical and various types of suspicious activity, for example, DoS, Probing. Training the network traffic was gathered for seven weeks took after by the two weeks accumulation of activity for testing reason as raw tcp dump format. The test information contains numerous attacks that were not infused during the training data gathering stage to influence the intrusion detection task more practical. It is trusted that the vast majority of the novel attacks can be gotten from the known attacks. From that point, the training and test information were handled into the datasets of five million and two million TCP/IP association records, separately.

The KDD Cup dataset has been broadly used as a benchmark dataset for a long time for performance evaluation of the NIDS. With KDD cup dataset, there was one major drawback that it contains a tremendous measure of excess records both in the training and test data. It was observed that just about 78% and 75% records are excess in the training and test data, individually. This redundancy makes the learning calculations one-sided towards the incessant attack records and leads to poor classification results for the less frequent, but harmful records. The training and test data were characterized with the minimum accuracy of 98% and 86% individually utilizing an exceptionally simple machine learning algorithms. It made the comparison of different machine learning algorithm complex task.
NSL-KDD was proposed to overcome the confinement of KDD Cup dataset. The dataset is gotten from the KDD Cup dataset. It enhanced the past dataset in two different ways. To start with, it dispensed with all the repetitive records from the preparation and test information. Second, it

parceled every one of the records in the KDD Cup dataset into different difficulty levels in view of the quantity of learning algorithm that can effectively arrange the records. From that point forward, it chose the records by irregular testing of the particular records from every difficulty level in a small amount of that is contrarily corresponding to their portion in the unmistakable records.

These multi-steps handling of KDD Cup dataset made the quantity of records in NSL-KDD dataset sensible for the preparation of any learning algorithm and reasonable also. Each record in the NSL-KDD dataset comprises of 41 includes and is marked with either normal or a specific sort of attack. These feature incorporate fundamental features got straightforwardly from a TCP/IP connection, activity features accumulated in a window interval, either time, e.g. two seconds or number of connection, and substance features extricated from the application layer information of connection. Out of 41 feature, three are nominal, four are binary, and remaining 34 features are continuous.

The training information contains 23 traffic classes that incorporate 22 classes of abnormal and one normal class. The test data contains 38 traffic classes that incorporate 21 attacks classes from the training data, 16 novel attacks, and one normal class. All these attacks are grouped into four categories based on the purposes, for example, DoS, Probing, U2R (client toroot), R2L (remote-to-nearby). Table-1 demonstrates the measurements of records for the preparation and test information for ordinary and diverse assault classes.

1. **Denial of Service Attack (DoS):** Deny legitimate request to a system.
2. **Users to Root Attack (U2R):** unofficial access to local super user or root.
3. **Remote to Local Attack (R2L):** Unauthorized local access from a remote machine.
4. **Probing Attack (RPOBE):** Information gathering.

TABLE 2.1. Traffic records distribution in the training and testing data for normal and attack

Traffic		Training	Testing
Normal		67343	9711
Attack	DoS	45927	7458
	U2R	52	67
	R2L	995	2887
	Probe	11656	2421

TABLE 2.2. NSL KDD Data set attack types

Class	Known Attack Subclass
DoS	Back, land, Neptune, pod, smurf, teardrop
Probe	Ipsweep, nmap, portsweep, satan
U2R	Buffer_overflow, loadmodule, perl, rootkit
R2L	ftp_write,guess_passwd,imap,multihop,phf,spy,warezclient, warezmaster

To understand and to perform evaluation for appropriate classification, it's important to know that that total number of instances in each class. The given fig below illustrate the class analysis done.

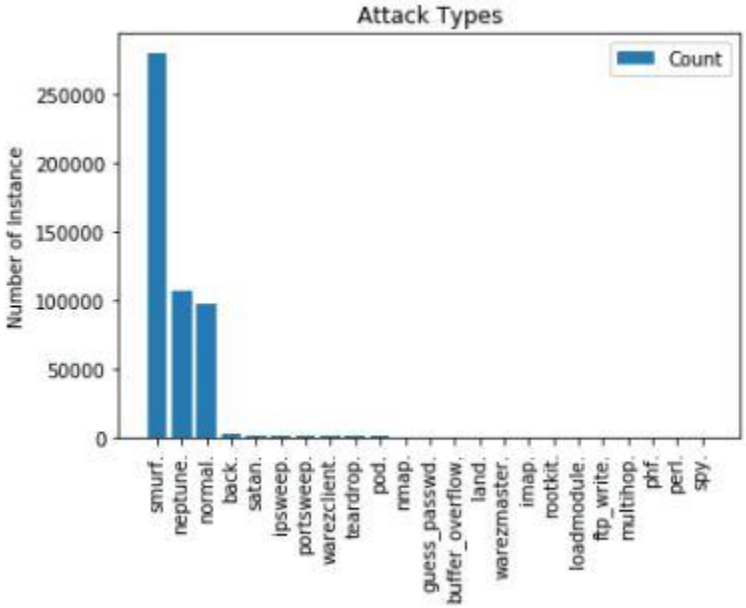


Fig 2.4. Attack types

Chapter 3

PROPOSED MODEL

An IDS continuously monitors the incoming and outgoing packets which pass through network for suspicious activity and enables the alarm to alert network administrator for abnormal pattern or suspicious activity of user to take indirect action to protects the network and maintain security. An IDS can be categorized in various ways based on their working location and the way it trained with dataset. A signature based intrusion detection system takes advantage of existing and pre-defined attacks signature of known attacks for training, the prediction model to protect the network. A signature based intrusion detection system has less false alarm rate but it unable to detects those threats which are not pre-defined or whose signature are not available while training of model. The general architecture of signature based IDS is shown in fig 2. An anomaly based IDS classify suspicious system behavior as abnormal or anomaly. These anomalies are mostly considered as anomaly. New attacks can be easily tracked down by anomaly based intrusion detection system.

For designing of any machine learning based prediction models or pattern recognition model, feature selection play a very important role. For training a prediction model we train it using data, data is collected from various source, it may be contain noise or impure data which leads to false prediction. For an accurate model we need to train our model with suitable data. For building a model, feature selection is very important, because if our training data contain noise or irrelevant feature, it increases the false alarm rate and reduces the performance of prediction model. Currently feature subset selection is become very important and interesting topic for researchers and their community because it's vast applications in the domain of classification, regression analysis and clustering. In other words feature subset selection for machine learning algorithm is most famous research area nowadays. Due to its easiness and effectiveness, the feature selection algorithm is accepted in many applications by researchers. Many data scientists are working on feature selection algorithm to improve the performance, accuracy and to reduce data over fitting problem in machine learning algorithms. Generally the existing feature selection algorithms are fail to perform or to generate relevant feature subset for unconventional data like interval, multi-valued, model and categorical dataset.

So in this project I proposed a model which has majorly two steps viz, designing a novel feature selection method using Binary PSO for interval valued dataset and train classification algorithm with those filtered relevant features.

3.1. Methodology

In this project I'm using Hybrid PSO algorithm with SVM classification algorithm to reduce the dataset. Using proposed feature selection method I'm selecting only those features which relevant for my prediction model. Using Hybrid Binary PSO algorithm with SVM classification algorithm, we build a wrapper based feature selection algorithm and use those selected features to train our classification algorithm. After training we evaluate the performance of prediction model, it shows that the performance has been improved.

In stage first I use available NSL KDD cup dataset for building my prediction model, this dataset contain improper values or noise. So to proceed forward we use some data preprocessing techniques to clean the data and to remove noise the data. Once we get noise free data, we have to perform some feature selection techniques because this dataset contains both relevant and irrelevant features, which reduce the performance of prediction model or machine learning model. Irrelevant features leads to increase false alarm rate and creates ambiguous model. So we need to select relevant features, for selecting features traditional methods are not that much effective? So we use our proposed feature selection method in stage 2. In this stage we use an objective function which to remove irrelevant features. In this objective function I use a binary mask to select random subset of features. These random subset of features are used to train an SVM classifier and calculates the performance of the classifier. Repeatedly it add and remove the features to maximize the performance of classifier. Once the performance of classifier is not affected by adding new features, it stops. In stage 3, we use these selected subset of features to train our prediction model to build an anomaly based IDS.

In this project we use various machine learning algorithm to build prediction model which use selected feature subset and we compare their accuracy and performance.

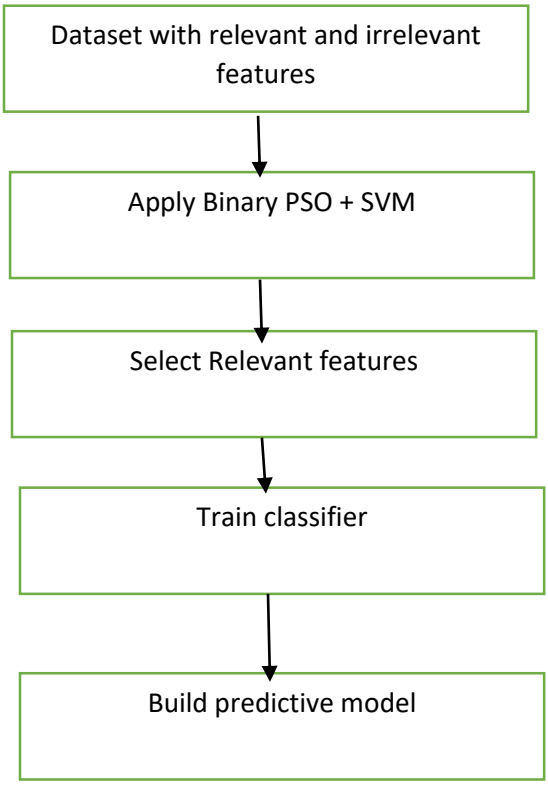


Fig 3.1. General architecture of proposed model

3.2. CLASSIFICATION ALGORITHM

The main idea in classification is to predict the target class by examining training dataset. this can be done by using appropriate machine learning algorithm. Selection of wrong supervised machine learning algorithm leads to misinterpretation of target class. While solving a classification problem, the selection of right classification algorithm is very important because the accuracy of the classification model is depends on the classification algorithm. Training time, data fitting, computational efficiency, stability and learning rate, these are some important factors for selection of classification algorithm. In this project, I use SVM algorithm to build predictive model and Binary PSO (particle swarm optimization) algorithm for wrapper based feature selection algorithm.

3.2.1 SUPPORT VECTOR MACHINE (SVM)

Support vector machine (SVM) falls under supervised machine learning, it can be used for both classification as well as regression analysis but it's originally designed for binary classification

problems. SVM is eager learning algorithm which gives preferable speculation capability over other classification algorithm for the information not ordered appropriately but rather it experiences low computational proficiency and high preparing time confines its utilization. SVM has much better capacity to detect outliers. The SVM are based on the concept of hypothesis planes that define decision boundaries. A hypothesis plane is one that separate input data to different class memberships.

In SVM the precision of classifier changes with changes of kernel. In SVM polynomial kernel gives lowest precision, linear and sigmoid kernel gives nearly same precision but RBF gives highest precision.

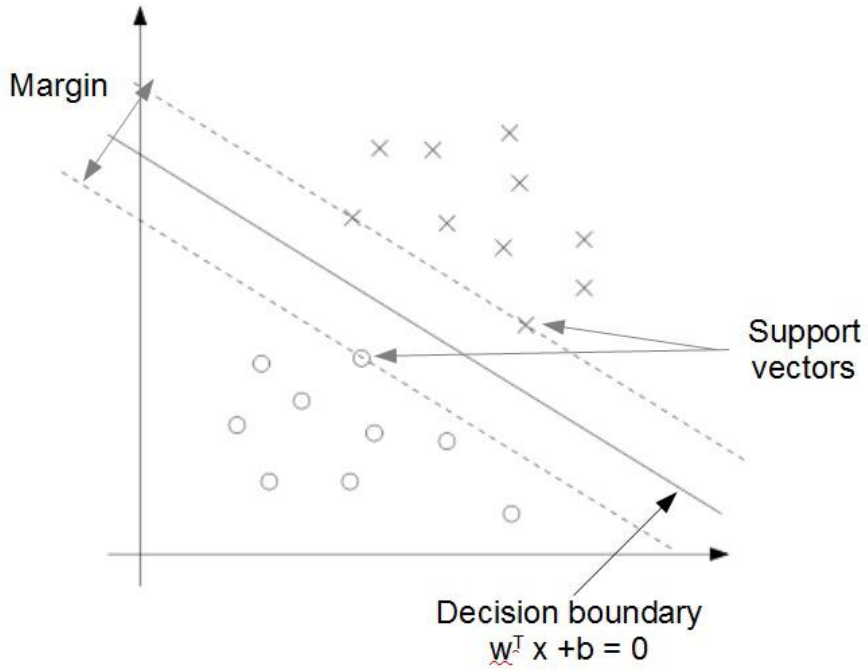


Fig 3.2. SVM binary classification

3.2.2 Naïve Bayes

Based on the strong independence presumptions between the features, Naïve bayes classifier falls under the family of probabilistic classifier. In Naive bayes classification algorithm we assumes that the presence or absence of any attribute of a class is not related with the presence or absence of other features.

Bayesian classification gives reasonable learning algorithm and prior information and observed information can be consolidated. Bayesian Classification gives an accommodating point of view for comprehension and assessing different machine learning algorithms. It registers unequivocal

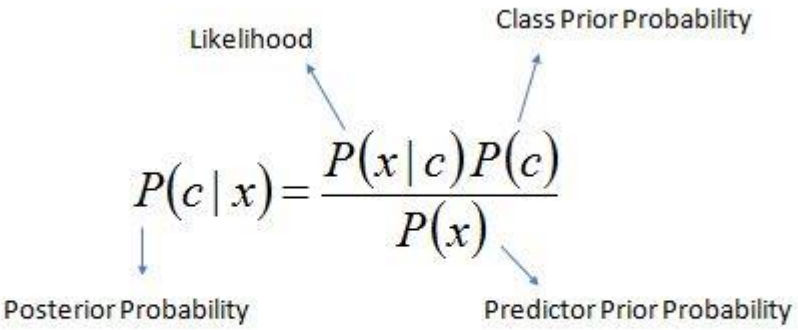
probabilities for speculation and it is hearty to polluting influence in input information. Training is fast because only the likelihood of each class and the likelihood of each class given distinctive input (x) values should be calculated. Coefficients need not to be fitted by optimization procedures.

In this, classification can be expressed on the arrangement of a hypothesis that is the data belongs to a specific class. Then the probability for the hypothesis for being true is been calculated. Bayes theorem calculates the posterior probability P.

Fundamentally, this Algorithm is utilized for the assumption that the impact of the estimation of a predictor (x) on a given class (c) is independent of the estimation of other predictors. The Formula used for the classification is:

Algorithm:

Bayes theorem provides a method for calculating the posterior probability, P(c|x), from P(c), P(x), and P(x|c). Naive Bayes classifier assume that the impact of the estimation of a predictor (x) on a given class (c) is not dependent on the values of other predictors. We called it as class conditional independence.



$$P(c \mid X) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

Fig 3.3.

- $P(c|x)$ is the posterior probability of the class (target) given predictor (attribute
- P(c) is the prior probability of the class.
- $P(x|c)$ is the likelihood, which is the probability of predictor given class.
- P(x) is the prior probability of the predictor.

There is no predictor in Zeros model, we try to find the best predictor in OneR model. Naive bayes includes all predictors using Bayes' rule and the independence assumptions between predictors.

3.2.3 RANDOM FOREST

Random forest (RF) algorithm is a supervised machine learning algorithm. RF classifier ensemble utilized for both regression analysis and classification problems. It's proposed by 'Breiman'. RF is a variation of bagging ensemble methods. In certain cases, its performance is better than boosting ensemble method and it performs faster than both boosting and bagging ensemble methods. In original version we RF is taught as version of bagging ensemble where random tree is a base classifier. However, we considered random forest as a ensemble learning whose base classifier is decision tree.

Besides, RF is a classifier which contains forest of tree composed classifiers, every tree developed as per a random vector and they are indistinguishably and independently distributed. A vote based mechanism is used to find most prominent class for input vector. In RF each tree in forest gives their vote to decide the input vector class and class having maximum vote can be defined as final class. Random forest algorithm acquire its diversity from samples of input attributes, from the input dataset, or by altering some parameters of the decision tree arbitrary.

Random forest has two parameters, which needs to be balanced: one is the quantity of variables which is chosen for each node and it's fixed for all nodes, and number of trees required to build the forest. In fig2 we have shown a demographic view of RF which we use in our project work. In random forest there will be various individual trees which is used in prediction model for prediction of input class as either normal or abnormal.

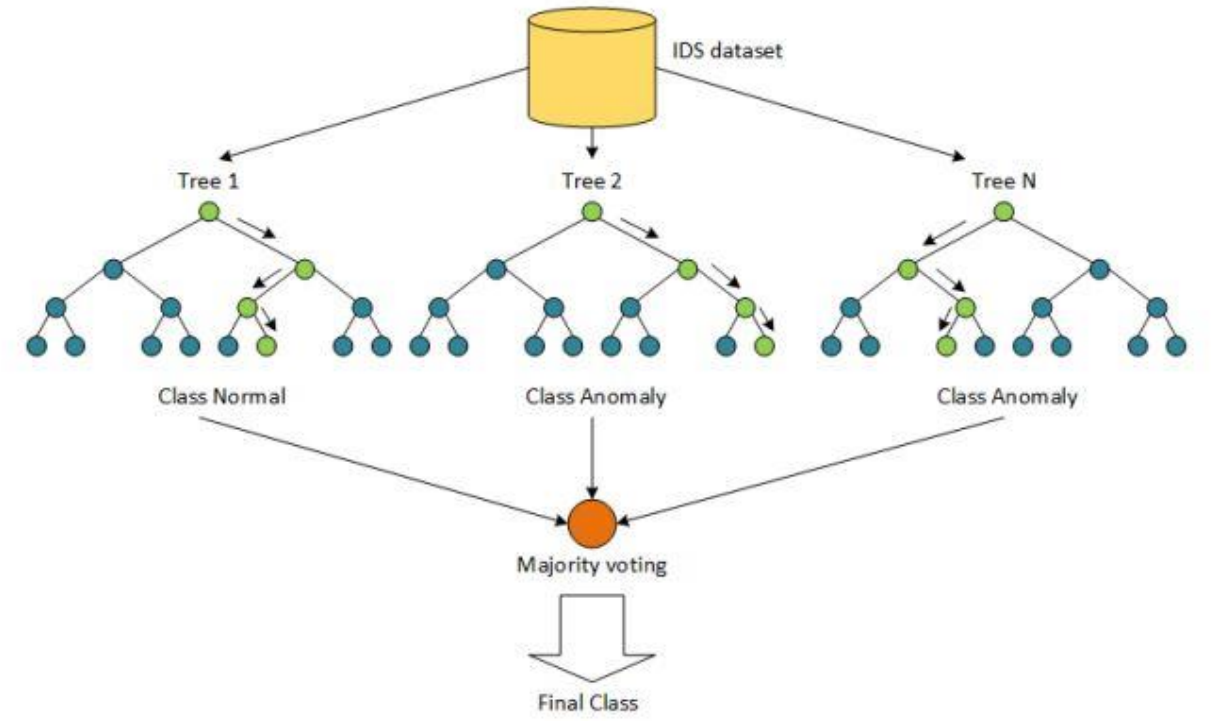


Fig 3.4. Illustration of Random Forest classifier

3.2.4 Binary Particle Swarm Optimization (BPSO)

Particle swarm optimization (PSO), which was inspired by studies of bird predation behavior, is an evolutionary algorithm developed by Kennedy and Eberhart. Particles are used to optimize the solutions in the search space and to record the best location on the current path. Each particle considers its own current position and velocity and records its own optimal solution (optimal position), pbest. Then, it adjusts its current position according to the global optimal solution among the population, gbest. The specific updating of each particle is performed as shown in Eqs (1) and (2)

$$v_h^{t+1} = wv_h^t + c_1 \times rand\left(pbest_h - x_h^t\right) + c_2 \times rand\left(gbest - x_h^t\right) \quad \text{Eq(1)}$$

$$x_h^{t+1} = x_h^t + v_h^{t+1} \quad \text{Eq(2)}$$

Where v_h^t the velocity of the hth particle in iteration t, w is is the inertia coefficient, and is the position of the hth particle in iteration t. The acceleration coefficients c_1 and c_2 are nonnegative constants that control the influence of pbest and gbest on the search process. In formula (1), wv_h^t represents the search capabilities of particles, whereas $c_1 \times rand(pbest_h - x_h^t)$ and $c_2 \times rand(gbest - x_h^t)$ represent the evolution of the particles themselves and the cooperation among particles, respectively.

The original PSO algorithm was developed for solving problems in a continuous space. Kennedy later adjusted the method used to update velocity and position and proposed binary particle swarm optimization (BPSO), which is suitable for solving discrete problems. In this approach, the particles in the population can search in a binary space. That is to say, the position vectors of the particles are represented by values of 0 or 1. The most important component of the BPSO algorithm is the transfer function, which converts continuous velocity values into discrete positions. The velocity obtained using Eq (1) is transformed into a vector in the interval [0,1] by means of the sigmoid function T, as given in Eq (3)

$$T\left(v_h^k(t)\right) = \frac{1}{1 + e^{-v_h^k(t)}} \qquad \text{Eq(3)}$$

Where $v_h^k(t)$ is the velocity of the hth particle in iteration t for the kth dimension? Hence, the position of a particle is updated to its new value using the following Eq(4)

$$x_h^k(t+1) = \begin{cases} 0 & if \ \text{rand} < \mathrm{T}\left(v_h^k(t+1)\right) \\ 1 & if \ \text{rand} \geq \mathrm{T}\left(v_h^k(t+1)\right) \end{cases} \qquad \text{Eq(4)}$$

3.3. PERFORMANCE MEASURE

Performance of classification model is very important. While solving a problem, accuracy of classifier is an important point to decide whether a model is good enough to solve the problem or not. Performance of classifier model is predicted by how many input instances are correctly

predicted by classifier. After implementing a machine learning algorithm, how effective is the model should be find out based on metrics and dataset. The prediction efficiency of classifier is measured by using common performance metrics like accuracy, specificity, sensitivity, training time etc. The number of test data are correctly classified by classification model, it can be tabulated in the form of confusion matrix. The parameters of confusion matrices for binary classifier is shown on table I.

Table 3.1 Intrusion detection taxonomy

Actual Data Class	Predicted Data Class	
	Positive	Negative
Positive	True Positive (tp)	False Negative (fn)
Negative	False Positive (fp)	True Negative (tn)

True Positive (tp): Correctly predicted the instance as normal.

True Negative (tn): Correctly predicted the instance as attack.

False Positive (fp): Incorrectly predict normal instance as attack.

False Negative (fn): Incorrectly predict attack instance as normal.

ROC (Receiver operating characteristics): We use this term to draw a graph between false positive rate and true positive rate. The term AUC is defined as the area under the curve in ROC graph, this gives the ROC value.

SENSEVITY: Sensitivity is also called as TPR (true positive rate). It gives how much actual normal instances are correctly classified.

$$\text{Sensitivity} = tp/(tp + fn) \qquad \text{Eq(5)}$$

SPECIFICITY: specificity is also called as TNR (true negative rate). It gives how actual attack instances are correctly classified.

$$\text{Specificity} = tn/(fp + tn) \quad \text{Eq(6)}$$

PRECISION: It's defined as the ratio of the number of true positive (tp) records divided by number of true positive (tp) and false positive (fp) records.

$$\text{Precision} = tp/(tp + fp) \quad \text{Eq(7)}$$

ACCURACY: It's defined as how many instances are correctly distinguish over the total number of instances.

$$\text{Accuracy} = (tp + tn)/(tp + tn + fp + fn) \quad \text{Eq(8)}$$

Training time: It's time taken by classifier to build predictive model on given dataset to predict the class label of instances.

Chapter 4

SIMULATION AND RESULT

The analysis is performed by using proposed feature selection algorithm and different classification algorithms. We use SVM and hybrid binary PSO algorithm to generate relevant feature subset. These selected features are used to train different classification algorithms. Using different classification algorithms which are trained using reduced feature, we have intrusion and normal traffic records, and also we have train these classification algorithm with all dataset without using proposed feature selection algorithm. Then we compare the performance of these two approaches. In first approach we are using reduced dataset to train classification algorithms and in other approach we are using reduced feature subset to train classification model. The comparison after building the model of these two approaches can be seen in table 4.1.

In addition the system resources required for training the classification algorithms without feature selection algorithm is reduced using proposed feature selection approach. Using proposed feature selection algorithm we will improve the performance of SVM classifier from 76% to 82%, increment of 6 percent in performance. Proposed algorithm also improve the performance of other classification algorithm. This indicates that anomaly based intrusion detection system based on proposed feature selection technique is better.

To calculate the performance of the proposed model, a performance measure- classification accuracy is used. It's defined as the ratio of correctly classified input sample to the total number of input samples.

Table 4.1. Classification accuracy with proposed model and without proposed model

Accuracy	Performance using proposed model in percent	Performance without using proposed model in percent
SVM	82	76
Random Forest	79	77
Naïve Bayes	72	67

4.1. Screenshot of results:

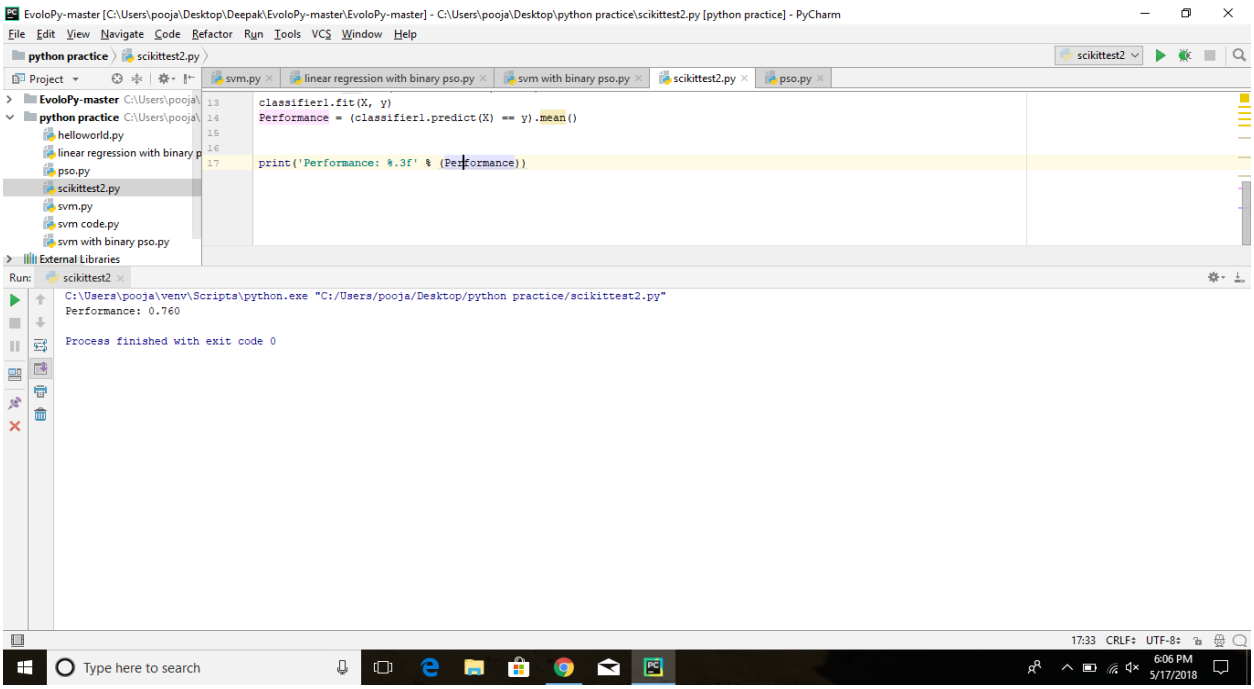


fig 4.1.

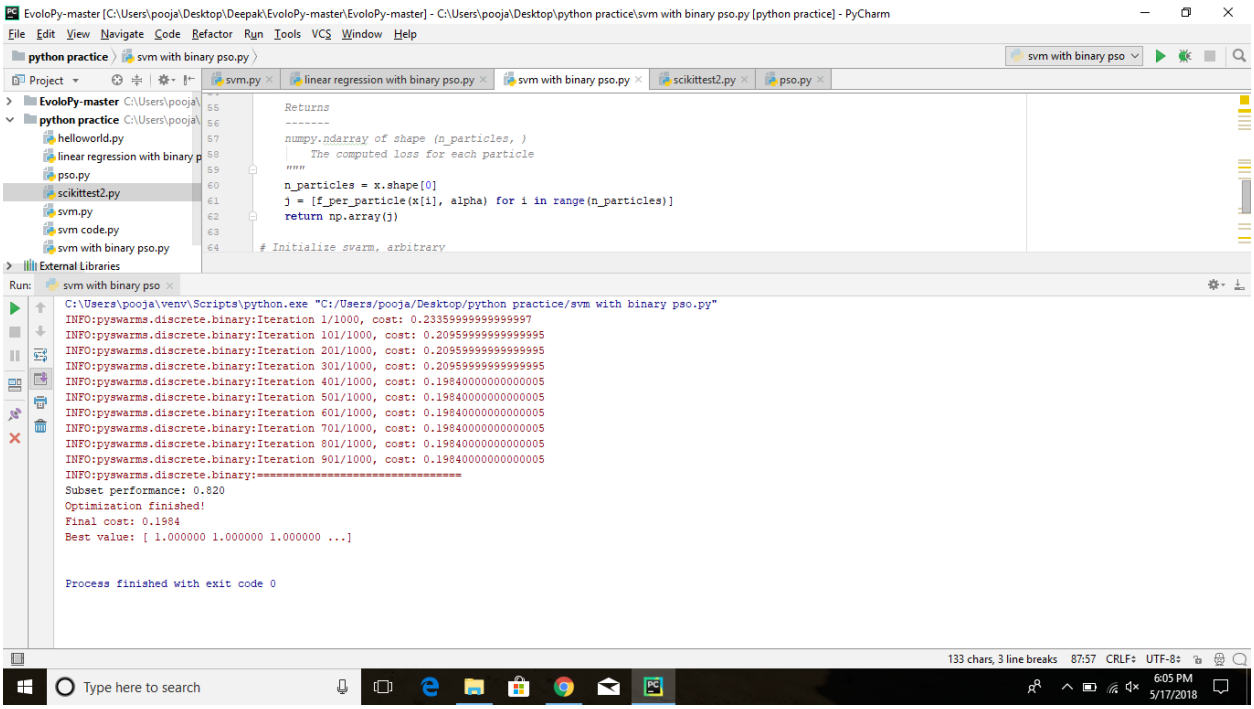


fig 4.2.

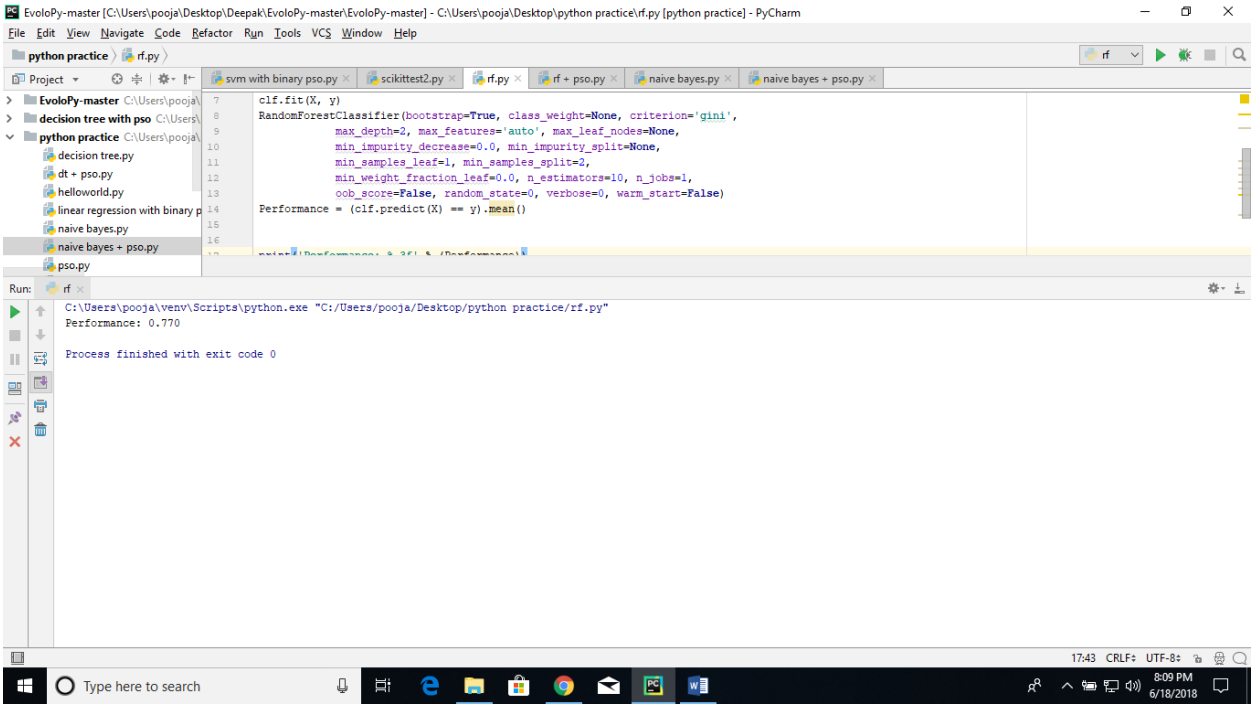

EvoloPy-master [C:\Users\pooja\Desktop\Deepak\EvoloPy-master\EvoloPy-master] - C:\Users\pooja\Desktop\python practice\rf.py [python practice] - PyCharm
File Edit View Navigate Code Refactor Run Tools VCS Window Help
python practice rf.py
svm with binary pso.py
scikittest2.py
rf.py
rf + pso.py
naive bayes.py
naive bayes + pso.py
clf.fit(X, y)
RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini',
max_depth=2, max_features='auto', max_leaf_nodes=None,
min_impurity_decrease=0.0, min_impurity_split=None,
min_samples_leaf=1, min_samples_split=2,
min_weight_fraction_leaf=0.0, n_estimators=10, n_jobs=1,
oob_score=False, random_state=0, verbose=0, warm_start=False)
Performance = (clf.predict(X) == y).mean()
Run: rf
C:\Users\pooja\venv\Scripts\python.exe "C:/Users/pooja/Desktop/python practice/rf.py"
Performance: 0.770
Process finished with exit code 0
Type here to search
8:09 PM
6/18/2018

fig 4.3.

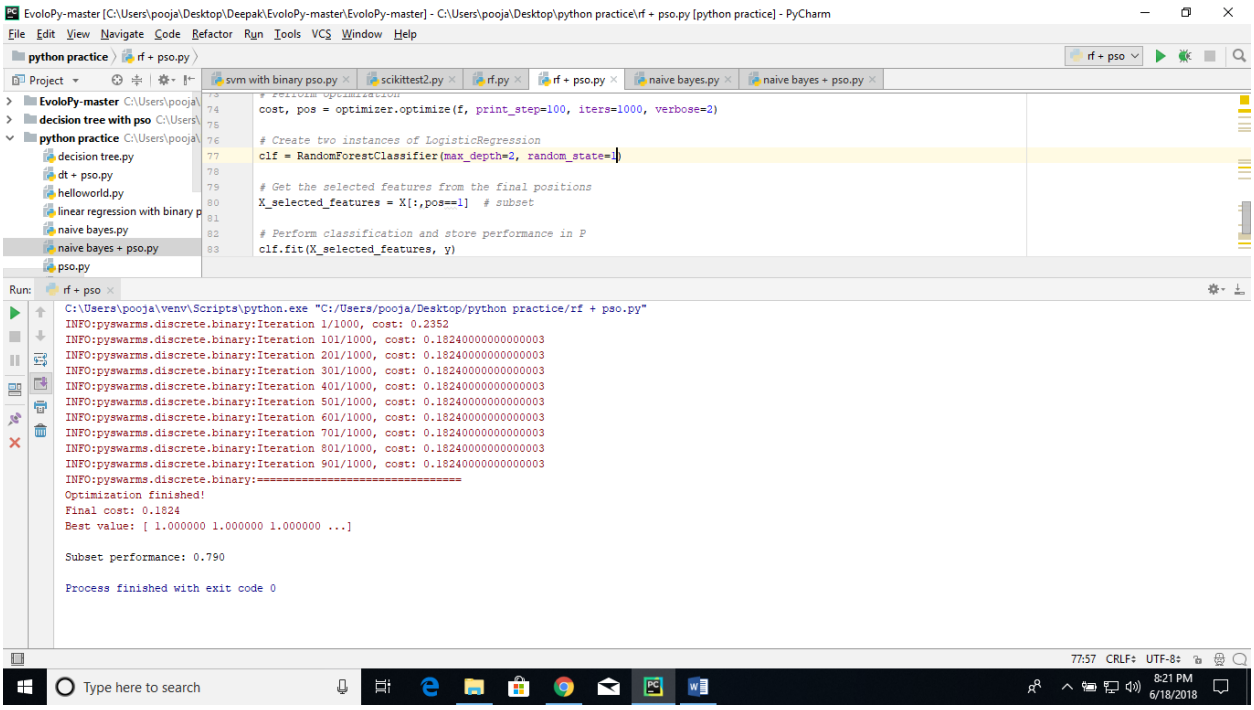

EvoloPy-master [C:\Users\pooja\Desktop\Deepak\EvoloPy-master\EvoloPy-master] - C:\Users\pooja\Desktop\python practice\rf + pso.py [python practice] - PyCharm
File Edit View Navigate Code Refactor Run Tools VCS Window Help
python practice rf + pso.py
rf + pso
cost, pos = optimizer.optimize(f, print_step=100, iters=1000, verbose=2)
# Create two instances of LogisticRegression
clf = RandomForestClassifier(max_depth=2, random_state=1)
# Get the selected features from the final positions
X_selected_features = X[:,pos==1] # subset
# Perform classification and store performance in P
clf.fit(X_selected_features, y)
Run: rf + pso
C:\Users\pooja\venv\Scripts\python.exe "C:/Users/pooja/Desktop/python practice/rf + pso.py"
INFO:pyswarms.discrete.binary:Iteration 1/1000, cost: 0.2352
INFO:pyswarms.discrete.binary:Iteration 101/1000, cost: 0.18240000000000003
INFO:pyswarms.discrete.binary:Iteration 201/1000, cost: 0.18240000000000003
INFO:pyswarms.discrete.binary:Iteration 301/1000, cost: 0.18240000000000003
INFO:pyswarms.discrete.binary:Iteration 401/1000, cost: 0.18240000000000003
INFO:pyswarms.discrete.binary:Iteration 501/1000, cost: 0.18240000000000003
INFO:pyswarms.discrete.binary:Iteration 601/1000, cost: 0.18240000000000003
INFO:pyswarms.discrete.binary:Iteration 701/1000, cost: 0.18240000000000003
INFO:pyswarms.discrete.binary:Iteration 801/1000, cost: 0.18240000000000003
INFO:pyswarms.discrete.binary:Iteration 901/1000, cost: 0.18240000000000003
INFO:pyswarms.discrete.binary:================================
Optimization finished!
Final cost: 0.1824
Best value: [1.000000 1.000000 1.000000 ...]
Subset performance: 0.790
Process finished with exit code 0
Type here to search
8:21 PM
6/18/2018

fig 4.4.

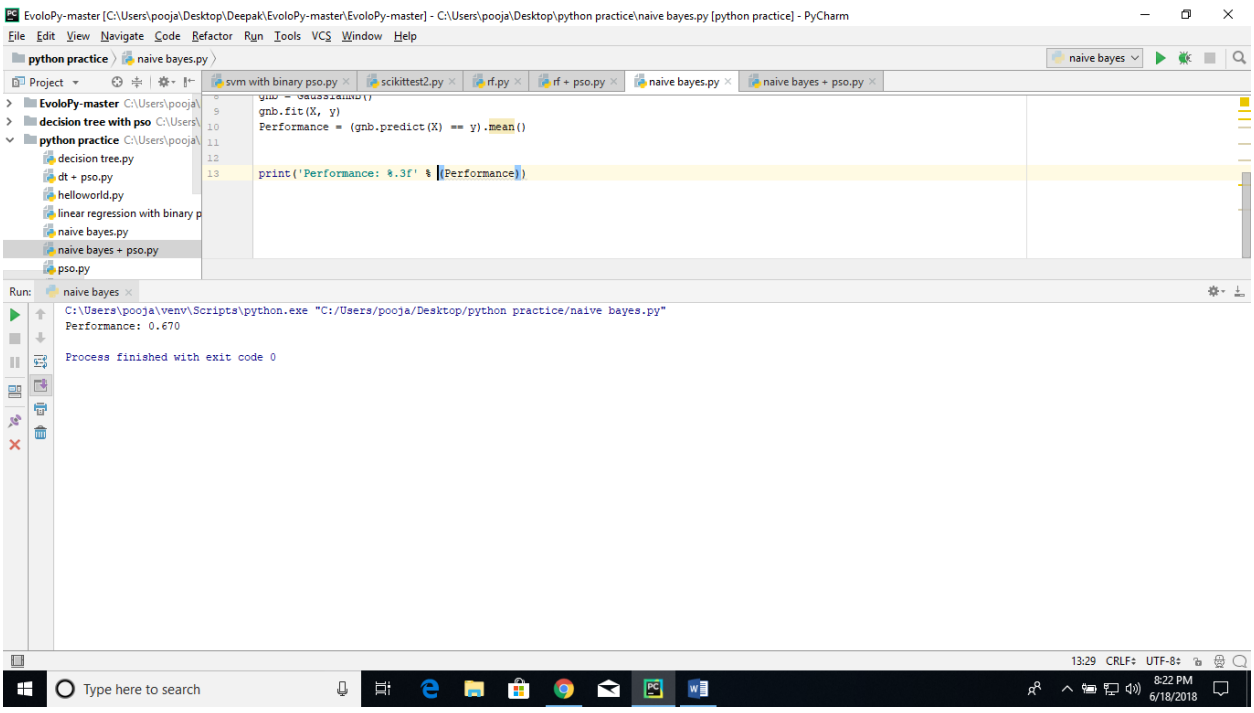

EvoloPy-master [C:\Users\pooja\Desktop\Deepak\EvoloPy-master\EvoloPy-master] - C:\Users\pooja\Desktop\python practice\naive bayes.py [python practice] - PyCharm
File Edit View Navigate Code Refactor Run Tools VCS Window Help
python practice > naive bayes.py
naive bayes
Project
svm with binary pso.py
scikittest2.py
rf.py
rf + pso.py
naive bayes.py
naive bayes + pso.py
EvoloPy-master C:\Users\pooja\
decision tree with pso C:\Users\
python practice C:\Users\pooja\
decision tree.py
dt + pso.py
helloworld.py
linear regression with binary p
naive bayes.py
naive bayes + pso.py
pso.py
gnb.fit(X, y)
Performance = (gnb.predict(X) == y).mean()
print('Performance: %.3f' % (Performance))
Run:
naive bayes
C:\Users\pooja\venv\Scripts\python.exe "C:/Users/pooja/Desktop/python practice/naive bayes.py"
Performance: 0.670
Process finished with exit code 0
13:29 CRLF UTF-8
Type here to search
8:22 PM
6/18/2018

fig 4.5.

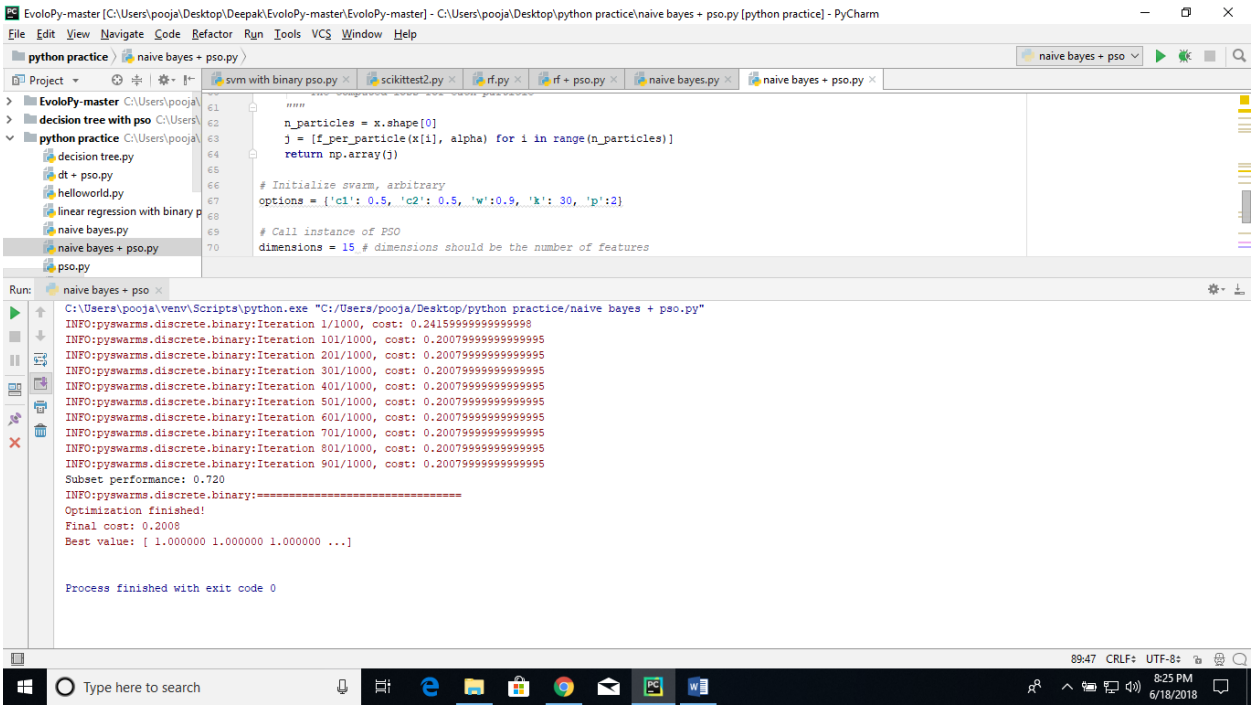

EvoloPy-master [C:\Users\pooja\Desktop\Deepak\EvoloPy-master\EvoloPy-master] - C:\Users\pooja\Desktop\python practice\naive bayes + pso.py [python practice] - PyCharm
File Edit View Navigate Code Refactor Run Tools VCS Window Help
python practice > naive bayes + pso.py
naive bayes + pso
Project
svm with binary pso.py
scikittest2.py
rf.py
rf + pso.py
naive bayes.py
naive bayes + pso.py
EvoloPy-master C:\Users\pooja\
decision tree with pso C:\Users\
python practice C:\Users\pooja\
decision tree.py
dt + pso.py
helloworld.py
linear regression with binary p
naive bayes.py
naive bayes + pso.py
pso.py
"""
n_particles = x.shape[0]
j = [f_per_particle(x[i], alpha) for i in range(n_particles)]
return np.array(j)
# Initialize swarm, arbitrary
options = {'c1': 0.5, 'c2': 0.5, 'w':0.9, 'k': 30, 'p':2}
# Call instance of PSO
dimensions = 15 # dimensions should be the number of features
Run:
naive bayes + pso
C:\Users\pooja\venv\Scripts\python.exe "C:/Users/pooja/Desktop/python practice/naive bayes + pso.py"
INFO:pyswarms.discrete.binary:Iteration 1/1000, cost: 0.24159999999999998
INFO:pyswarms.discrete.binary:Iteration 101/1000, cost: 0.20079999999999995
INFO:pyswarms.discrete.binary:Iteration 201/1000, cost: 0.20079999999999995
INFO:pyswarms.discrete.binary:Iteration 301/1000, cost: 0.20079999999999995
INFO:pyswarms.discrete.binary:Iteration 401/1000, cost: 0.20079999999999995
INFO:pyswarms.discrete.binary:Iteration 501/1000, cost: 0.20079999999999995
INFO:pyswarms.discrete.binary:Iteration 601/1000, cost: 0.20079999999999995
INFO:pyswarms.discrete.binary:Iteration 701/1000, cost: 0.20079999999999995
INFO:pyswarms.discrete.binary:Iteration 801/1000, cost: 0.20079999999999995
INFO:pyswarms.discrete.binary:Iteration 901/1000, cost: 0.20079999999999995
Subset performance: 0.720
INFO:pyswarms.discrete.binary:================================
Optimization finished!
Final cost: 0.2008
Best value: [1.000000 1.000000 1.000000 ...]
Process finished with exit code 0
89:47 CRLF UTF-8
Type here to search
8:25 PM
6/18/2018

fig 4.6.

CONCLUSION

In this project we address the problem of feature selection for anomaly based feature selection prediction model. Traditional feature selection algorithm are not as efficient as our proposed model. In this project we introduced a wrapper based feature selection algorithm in which we used Hybrid Binary PSO algorithm with SVM classifier to reduce irrelevant features from dataset and to select only those features subset which are suitable and required for our prediction model. The proposed feature selection model removes the redundant and unnecessary features and improve the performance and accuracy of our intrusion detection model and also reduces the false alarm rate. The time and resources required to build intrusion detection model is also less compare to the model build with the complete dataset. The analysis is performed on the NSL KDD datasets. In this project we also compare the performance of different supervised machine learning algorithms which are trained using reduced feature subset and to get the best one.

In the future work, we plan to optimize our current model to get better performance and accuracy. In future we will work on boosting methods with nature inspired algorithm. We will also intended our work towards unsupervised learning algorithms.

REFRENCE

[1] A. Gul and E. Adali, "A feature selection algorithm for IDS," in *2nd international conference on Computer science and engineering*, 2017.

[2] K. Lahre, T. d. Diwan, S. K. Kashyap and P. Agrawal, "Analyze Different approaches for IDS using KDD 99 Data Set," *International Journal on Recent and Innovation Trends in Computing and Communication,* vol. 1, no. 8, pp. 645-651, 2013.

[3] P. Bhoria and K. Garg, "An Imperial learning of Data Mining Classification Algorithms in Intrusion Detection Dataset," *International Journal of Scientific & Engineering Research*, vol. 4, no. 6, pp. 2394-2399, 2013.

[4] S. Duhan and P. Khandnor, "Intrusion detection system in wireless sensor networks: A comprehensive review," in *: Electrical, Electronics, and Optimization Techniques (ICEEOT), International Conference on*, Chennai, 2016.

[5] Ullah, Imtiaz, and Qusay H. Mahmoud. "A filter-based feature selection model for anomaly-based intrusion detection systems." In *Big Data (Big Data), 2017 IEEE International Conference on*, pp. 2151-2159. IEEE, 2017.

[6] Enache, A.C. and Sgârciu, V., 2015, July. A feature selection approach implemented with the Binary Bat Algorithm applied for intrusion detection. In *Telecommunications and Signal Processing (TSP), 2015 38th International Conference on* (pp. 11-15). IEEE.

[6] Kumar, N.V. and Guru, D.S., 2017, November. A novel feature ranking criterion for supervised interval valued feature selection for classification. In *Document Analysis and Recognition (ICDAR), 2017 14th IAPR International Conference on* (Vol. 5, pp. 71-76). IEEE.

[7] Sharma, A., Zaidi, A., Singh, R., Jain, S. and Sahoo, A., 2013, December. Optimization of SVM classifier using Firefly algorithm. In *Image Information Processing (ICIIP), 2013 IEEE Second International Conference on* (pp. 198-202). IEEE.

[8] Harb, H.M. and Desuky, A.S., 2014. Feature selection on classification of medical datasets based on particle swarm optimization. *International Journal of Computer Applications*, *104*(5).