Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 4, Issue. 5, May 2015, pg.1037 – 1045

RESEARCH ARTICLE

REDUCTION IN FALSE POSITIVE RATE BY COMBINING SVM AND KNN ALGO

Sushil Kumar Mishra[1], Deepak Singh Rana[2]

[1]PG Student, Computer Science Engineering, Graphic Era Hill University, Uttarakhand, India

[2]Assistant Professor, Computer Science Engineering, Graphic Era Hill University, Uttarakhand, India

***ABSTRACT:* With the emergence of internet, we have faced intrusion problem in computer system such as a file is self-replicating in nature and try to encrypt other file etc. This research work is based on intrusion detection problem of network security. Its motive is to detect to detect a network behavior as normal or abnormal. In this research work, two different machine learning algorithms have been combined together to reduce its duplicity and redundancy to enhance accuracy of new developed algorithm. Its experimental result produces better result than other algorithm in terms of performance, accuracy and false positive. This combined algorithm has been applied on KDDCUP99 dataset to find better result by enhancing its performance, accuracy and reducing its false positive rate.**

The objective of this proposed work is to predict the estimated false positive and accuracy with number of features and attributes by combining support vector machine (SVM) and K nearest neighbor (KNN) algorithm, to find the better and improved accuracy as compared to previous work done. As support vector machine (SVM) and K nearest neighbor (KNN) algorithms are combined to develop a new algorithm. In new developed algorithm, duplicity and redundancy are removed. By which, accuracy is enhanced and false positive rate is reduced.

***Key Words:* KDDCUP99 dataset, Accuracy, False positive Rate.**

--***--

1. INTRODUCTION

With the rapid growth of internet, there have increased in the number of attacks, intrusion detection system (IDS) has become a solution of information security. With the help of firewall we can facilitate some protection over a short period of time but they can not provide full protection. The main motive of intrusion detection system (IDS) is to help computer systems to deal with attacks.

An intrusion detection system (IDS) is able to monitor the behavior of all files those are coming in computer system if any file is suspicious or malicious so intrusion detection system (IDS) can detect that malicious file or attacks. Intrusion detection system (IDS) has created many models based on clustering to separate a normal and abnormal file. Intrusion detection system (IDS) provides the protection a network traffic from malicious file or virus, It basically maintains confidentiality and integrity of a computer system. Any unauthorized access of any data can not be done so secrecy of network traffic can be well maintained.

On the basis of data intrusion detection system (IDS) are categorized to Host based intrusion detection system (HIDS) and Network based intrusion detection system (NIDS). In network based intrusion detection system (NIDS), only individual packet coming through the network are analyzed. The host based intrusion detection system (HIDS) analyzes above operation on a single host.

There are basically only two kinds of intrusion detection system used to detect intrusions in computer system

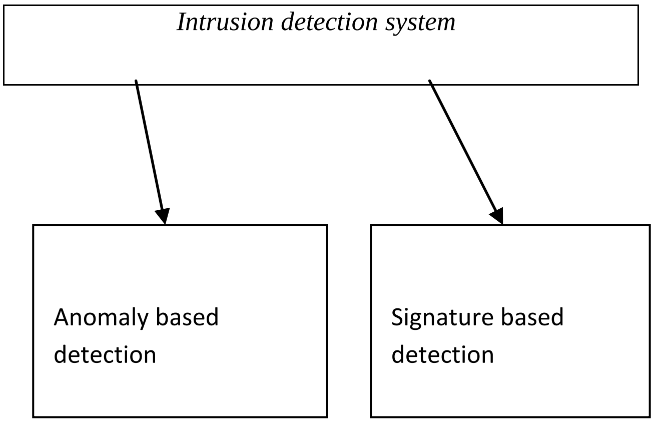


Fig-1 Types of IDS

1.1 Anomaly based detection

Anomaly based detection system is based on heuristic rule which is used to monitor a normal and abnormal behaviors in a computer system. Anomaly based detection system contains many types of machine learning algorithm, these algorithm is used to identify a normal behavior and abnormal behavior in computer system.
The main drawback of anomaly based detection system is higher false positive rate and low accuracy.

1.2 Signature based detection

Signature based detection system can detect only predefined attacks, these attack or virus patterns are stored in database if a similar pattern and similar situation occurs so it is declared as attack. In signature based detection system, previous pattern data is stored in database. The computer virus, those are discovered. Its signatures are created and stored in database. If any file comes in a computer network so its signatures are matched with database. If file matches with virus signature so it is declared a computer virus otherwise a normal file. The main drawback of signature based detection system is that it can not detect a new attack.

2. EXPERIMENTAL PARAMETERS

There are the following parameters to be evaluated during this experiment

Performance: *Performance is basically used to deal with achieving a target in more correct manner.*
Performance = (TP)/(TP)+(TN)

Accuracy: *Accuracy is the degree of closeness of measurements of a quantity to that quantity's actual (true) value.*
Accuracy = (TP+TN)/(TP+TN+FP+FN).

False Positive Rate: *False positive rate is used to falsely detect a normal file as abnormal file.*
False positive rate = (FP)/(FP+TN).

Negative Predictive Value: *Negative predictive value is used to detect a abnormal file as normal file.*
Negative predictive value = TN/(FN+TN).

3. EVALUATION DATA SOURCES

Several parameters is evaluated by the standard data set KDDCUP99 provided by MIT laboratory. In KDDCUP99 data set, there are several types of attacks those can be categorized as normal and abnormal data.

MIT Lincoln laboratory has established a computer network. About 9 days, which monitors network traffic, containing normal and abnormal data.

KDDCUP99 dataset contains normal, remote to user, buffer overflow, user to root, guess_pawd and probe type of attacks.

Denial of service: Denial of service (DOS) intrusion is an attack in which, correct information can not be made available to legitmate receiver.

Denial of service intrusion can slow down computer system.

User to Root (U2R): User to Root (U2R) intrusion ia an attack in which, an attacker can access client's password without any authentication and access secret information from computer by using stolen password.

Remote to User (R2U): In this Intrusion, attacker transmits a packet over network which is not legitmate for particular network which overcrowds traffic in the network. Remote to User (R2U) can restart a computer system repeatedly.

Probe: In this intrusion, attacker has the ability to monitor all information which are being transmitted in a network and can access that information without any authentication.

4. COMBINING SVM AND KNN ALGORITHM

Support vector machine (SVM) is a machine learning method used for classification in which, a hyperplane is created by support vector for separating normal and abnormal data from each other. Support vector machine (SVM) can be divided into two phases-

1- ***Training phase***

2- ***Testing phase***

1-Training phase*: Support vector machine (SVM) is basically used to learn a large set of pattern from dataset. In dataset, there are several types of homogeneous pattern and heterogeneous pattern which can give better classification scheme for separating normal and abnormal data.*

2-Testing phase: *Testing phase basically depends upon training phase can be done by support vector machine. In testing phase, machine learning algorithm is applied on data sets to get desired result.*

K nearest neighbor *algorithm is a kind of machine learning algorithm which can provide a mechanism to solve traveling salesman problem (TSP).*

By applying K nearest neighbor algorithm (KNN), false positive rate can be evaluated but it gives higher false positive rate which is inversely proportional to accuracy.

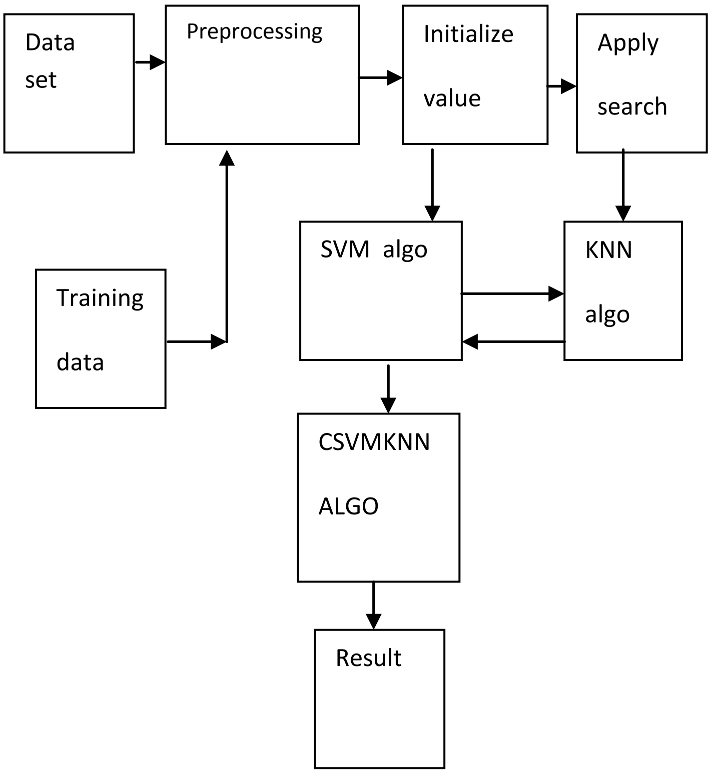


Fig- 2 IDS using CSVMKNN

Algorithm1: SVM with KNN clustering

Input*: Use training data set containing normal and abnormal data (Class type).*
Output*: classifier.*

```
1 start
2 Randomly select data  from different class;
3 Generate of support vector machine classifier;
4  While number of iteration to add data to data set
5  Make available support vector to create hyperplane;
6 Hyperplane separate normal and abnormal data;
7 Apply KNN clustering
8  Add points to the training dataset
9 Retrain support vector machine classifier using update     training dataset
10   end
```

The active learning *support vector machine process is introduced at this stage; support vector machine (SVM) is used to perform support vector machine training rely on various training data sets.The main goal of active learning support vector machine is how to choose the training data set for each training step. The separate hyperplane is used to tackle such issue; hyperplanes are created by support vectors. The hyperplane is tailored gradually by adding data points between maximum margins after each support vector machine training phase. Hence it produces more efficient selection strategy of data points.*

Algorithm2: Training in CSVMKNN

Input: *Training data set.*
Input: *T-Number of iteration.*
Input: *D-Maximum detection rate.*
Output: *Support vector machine (SVM) and K nearest neighbor (KNN) Classifier.*

```
1 Start
2 Normalize data;
```

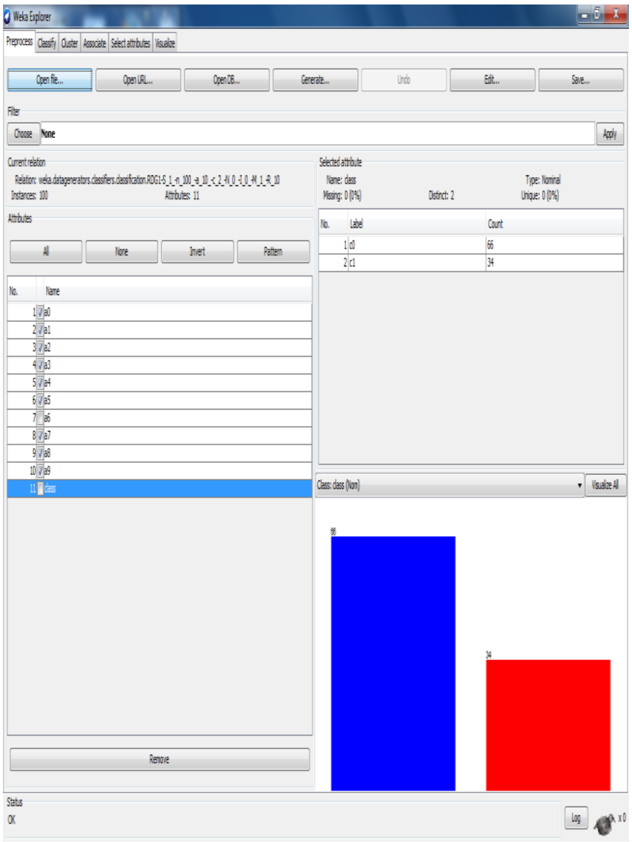
```
Let DR is detection rate , initially 0;
While   DR < D  do
    for i =1,…….. , T do
        SVM training phase;
        KNN Clustering Phase;
    end
    Construct classifiers;
    Do testing to updated  DR;
end
end
```

5. EXPERIMENTAL SETUP AND RESULT

Support Vector Machine (SVM) Classifier:

Support vector machine (SVM) is a group of supervised method used to apply on dataset (KDDCUP99) for classification and regression through which, false positive rate can be evaluated.

Support vector machine classifier is a method used to create a hyper plane by using support vector between data points. The main function of hyper planes is to separate normal and abnormal data from each other. Support vector machine (SVM) is also used for pattern identification. These methods have some specific application in speech analysis, image analysis etc.



Filter: None

Relation: weka.datagenerators.classifiers.classification.RDG1-S_1_-n_100_-a_10_-c_2_-N_0_-I_0_-M_1_-R_10
Instances: 100
Attributes: 11

Name: class
Type: Nominal
Missing: 0 (0%)
Distinct: 2
Unique: 0 (0%)

No.	Label	Count
1	c0	66
2	c1	34

No.		Name
1	✓	a0
2	✓	a1
3	✓	a2
4	✓	a3
5	✓	a4
6	✓	a5
7	□	a6
8	✓	a7
9	✓	a8
10	✓	a9
11	□	class

Class: class (Nom)

OK

Fig-3 SVM classifier

The optimal separation hyper plane is a linear classifier used with a maximum margin for finite pattern.
Suppose classification of different classes of same pattern is linear separable.
Linear separable hyper plane is w . x + b=0.

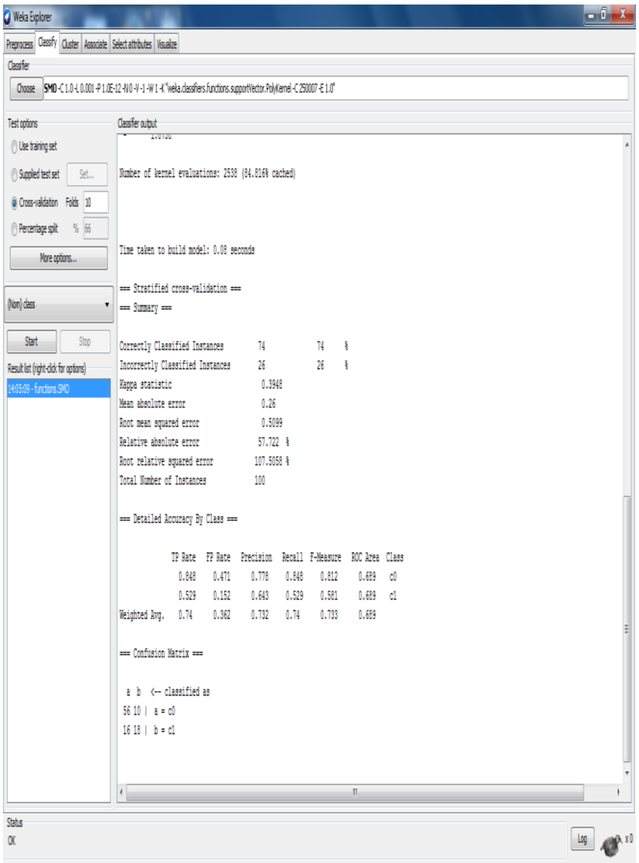
$$\begin{cases} \mathbf{w} \cdot \mathbf{x}_i + b > +1 & \text{if} \quad y_i = +1 \\ \mathbf{w} \cdot \mathbf{x}_i + b < -1 & \text{if} \quad y_i = -1 \end{cases}$$

These equations may be expressed in short form as

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq +1$$

Alternative

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1 \geq 0$$



SMO -C 1.0 -L 0.001 -P 1.0E-12 -N 0 -V -1 -W 1 -K "weka.classifiers.functions.supportVector.PolyKernel -C 250007 -E 1.0"

○ Use training set
○ Supplied test set
● Cross-validation Folds 10
○ Percentage split % 66

(Nom) class

14:05:09 - functions.SMO

```
Number of kernel evaluations: 2538 (84.816% cached)



Time taken to build model: 0.08 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        74               74      %
Incorrectly Classified Instances      26               26      %
Kappa statistic                        0.3948
Mean absolute error                    0.26
Root mean squared error                0.5099
Relative absolute error               57.722  %
Root relative squared error          107.5058 %
Total Number of Instances            100

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.848     0.471      0.778     0.848     0.812      0.689    c0
                 0.529     0.152      0.643     0.529     0.581      0.689    c1
Weighted Avg.    0.74      0.362      0.732     0.74      0.733      0.689

=== Confusion Matrix ===

  a  b   <-- classified as
 56 10 |  a = c0
 16 18 |  b = c1
```

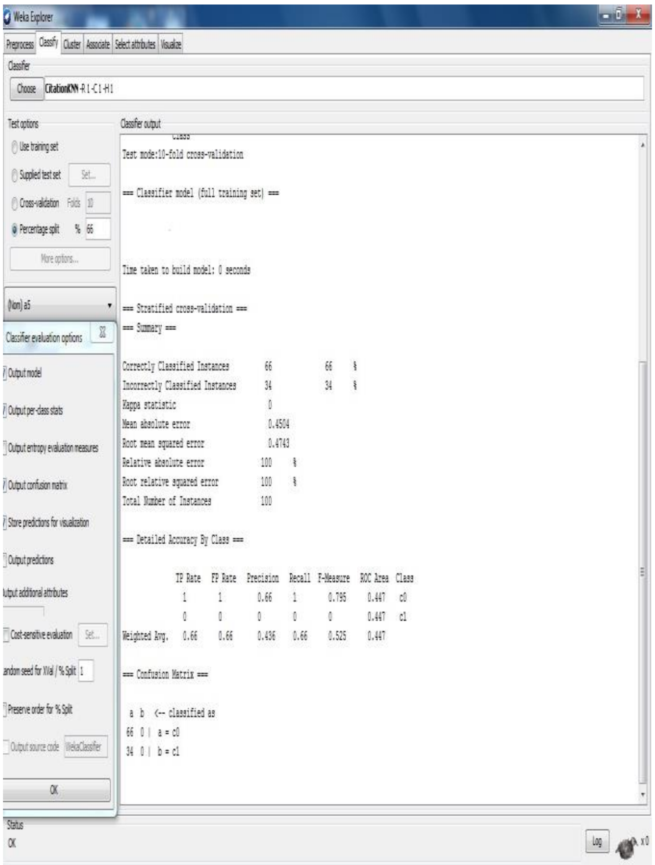
OK

Fig-4 SVM FPR

Table 1 with Experiment 1 (SVM)

False Positive Rate (FPR)
0.362

K Nearest Neighbor (KNN) Classifier is used to summarize training dataset which can be used to find out new observation by reducing number of comparison.
K nearest neighbor (KNN) algorithm is applied on dataset to find out its false positive rate to evaluate higher accuracy.



CitationKNN -R 1 -C 1 -H 1

○ Use training set
○ Supplied test set
○ Cross-validation Folds 10
● Percentage split % 66

(Nom) a5

☑ Output model
☑ Output per-class stats
□ Output entropy evaluation measures
☑ Output confusion matrix
☑ Store predictions for visualization
□ Output predictions
utput additional attributes
□ Cost-sensitive evaluation
andom seed for XVal / % Split 1
□ Preserve order for % Split
□ Output source code WekaClassifier

Classifier output

```
Test mode:10-fold cross-validation

=== Classifier model (full training set) ===



Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        66          66    %
Incorrectly Classified Instances      34          34    %
Kappa statistic                        0
Mean absolute error                    0.4504
Root mean squared error                0.4743
Relative absolute error              100     %
Root relative squared error          100     %
Total Number of Instances            100

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 1         1         0.66      1         0.795     0.447    c0
                 0         0         0         0         0         0.447    c1
Weighted Avg.    0.66      0.66      0.436     0.66      0.525     0.447

=== Confusion Matrix ===

  a  b   <-- classified as
 66  0 |  a = c0
 34  0 |  b = c1
```

Status
OK

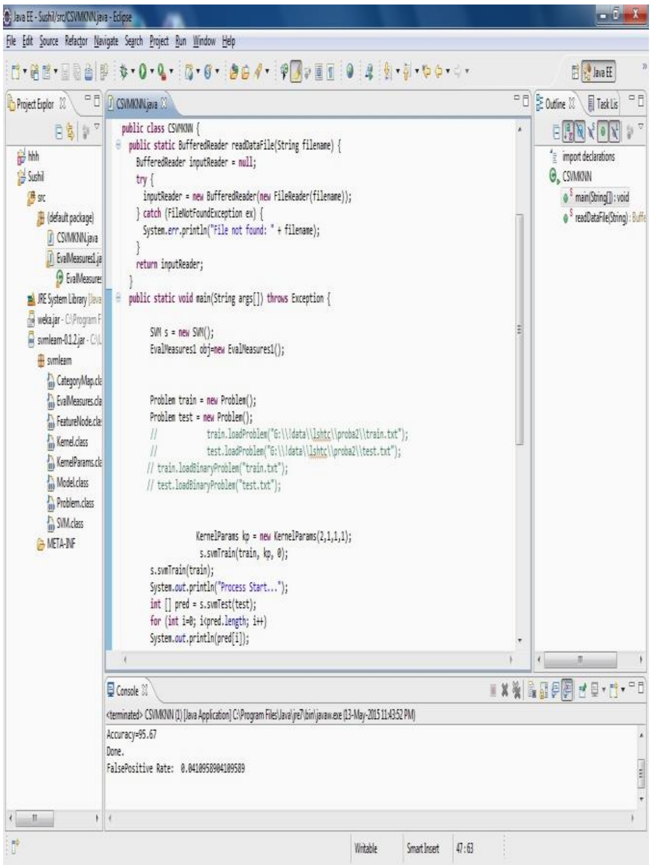
Fig-5 KNN classifier

Table 2 with Experiment 2 (KNN)

False Positive Rate
0.66

CSVMKNN classifier contains all attributes of support vector machine (SVM) and K nearest neighbor (KNN) algorithms. CSVMKNN algorithm is applied on dataset to produce accuracy, false positive rate.

False positive rate (FPR) can be reduced in CSVMKNN algorithm, by removing duplicate and redundant data from CSVMKNN algorithm.



```
public class CSVMKNN {
  public static BufferedReader readDataFile(String filename) {
    BufferedReader inputReader = null;
    try {
      inputReader = new BufferedReader(new FileReader(filename));
    } catch (FileNotFoundException ex) {
      System.err.println("File not found: " + filename);
    }
    return inputReader;
  }
  public static void main(String args[]) throws Exception {


        SVM s = new SVM();
        EvalMeasures1 obj=new EvalMeasures1();



        Problem train = new Problem();
        Problem test = new Problem();
        //              train.loadProblem("G:\\!data\\lshtc\\proba2\\train.txt");
        //              test.loadProblem("G:\\!data\\lshtc\\proba2\\test.txt");
       // train.loadBinaryProblem("train.txt");
       // test.loadBinaryProblem("test.txt");



                    KernelParams kp = new KernelParams(2,1,1,1);
                     s.svmTrain(train, kp, 0);
        s.svmTrain(train);
        System.out.println("Process Start...");
        int [] pred = s.svmTest(test);
        for (int i=0; i<pred.length; i++)
        System.out.println(pred[i]);
```

```
<terminated> CSVMKNN (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (13-May-2015 11:43:52 PM)
Accuracy=95.67
Done.
FalsePositive Rate:  0.0410958904109589
```

Fig-6 CSVMKNN Classifier

Table 3 with Experiment 3 (CSVMKNN)

False Positive Rate
0.041

Table 4 with Comparison of False Positive Rate

Evaluation Measure	*SVM*	*KNN*	*CSVMKNN*
False positive Rate	*0.362*	*0.66*	*0.041*

CSVMKNN algorithm generates lesser false positive rate than Support vector machine (SVM) and K nearest neighbor (KNN) algorithms..

6. CONCLUSIONS

In this research work, a new machine learning algorithm based on classification is introduced by combining support vector machine (SVM) and K nearest neighbor (KNN) algorithms for solving intrusion detection problem.

In order to achieve higher accuracy, duplicity and redundancy are removed by reducing false positive rate.

REFERENCES

[1]. pgale, Robert, Sheodoor schote, rengin and Christopher kruegel."A Literature analysis on automated malware analysis technique ."

[2]. Pargas, Rob Jonathan jarcy, Eleazar Aguirre Anaya , Samon Galeana Huerta and Alba Felix Moreno Hernandez,"Security controls for Android" In Computational Aspects of Social Networks (CASoN), 2012 Fourth International Conference on, pp.212-216,IEEE,2012

[3]. Blasing, Thomas, Leonid Batyuk, A-D.Schmidt, Seyit Ahmet Camtepe, and Sahin Albayrak." An android application sandbox system for suspicious software detection" In Malicious and Unwanted Software (MALWARE), 2010 5th International Conference on ,pp. 55-62 IEEE, 2010.

[4]. Johnson Ryan, Zhaohui Wang , Corey Gagnon and Angelos Stavrou." Analysis of Android Applications' Permissions. " In Software Security and Reliability Companion(SERE-C),2012 IEEE Sixth International Conference on, pp. 45 - 46.IEEE,2012.

[5]. Susan M. B. and Rayford B.V. (2000). Intrusion detection via fuzzy data mining, Proceedings of the 12th Annual Canadian Information Technology,Ottawa, Canada, June 19-23, 2000, PP.109-122.

[6]. A Detailed Analysis of the KDD CUP 99 Data Set, Mahbod Tavallaee, Ebrahim Bagheri, Wei Lu, and Ali A.

[7]. Susan M. B. and Rayford B.V. (2000). Intrusion detection via fuzzy data mining, Proceedings of the 12th Annual Canadian Information Technology,Ottawa, Canada, June 19-23, 2000, PP.109-122.

[8]. A Detailed Analysis of the KDD CUP 99 Data Set, Mahbod Tavallaee, Ebrahim Bagheri, Wei Lu, and Ali A.

[9]. Blasing, Thomas, Leonid Batyuk, A-D.Schmidt, Seyit Ahmet Camtepe, and Sahin Albayrak." An android application sandbox system for suspicious software detection" In Suspicious and Unwanted Software (MALWARE), 2013 5th International Conference on ,pp. 55-62 IEEE, 2013.