International Journal of Engineering Applied Science and Management

ISSN (Online): 2582-6948
Vol. 1 Issue 1, June 2020

Anomaly Detection in KDD99 with Reduced Feature Using Entropy, Gain and KNN Classifier

Rashmi Singh[1], Vinay Singh[2], Harsh Pratap Singh[3], Narendra Sharma[4]
Assistant Professor, RITS, Bhopal (M.P.), India[1]
Assistant Professor, SISTEC, Gandhinagar, Bhopal (M.P.), India[2]
Assistant Professor, SSSUTMS, Sehore (M.P.), India[3, 4]
rashisingh85@gmail.com[1]

Abstract

Internet is a widely used technology for the data communication in present days but as per increasing the demand of internet number of attackers also increases in the same proportion. To ensure the security of the network a powerful intrusion detection system is required. The aim of IDS system to continuously monitoreach and every activity which are occurring over the network and also provide accessing to the users.This paper proposes and IDS using KNN classifier and Genetic Algorithm for network with good feature reduction technique. The simulation of proposed approach is done in MATLAB2012a toolbox using KDDCUP'99 dataset. It is finally observed that the KNN classifier gives more accurate results than the other existing techniques and for feature selection method, information gain ratio based feature selection is better.

Keywords: Feature reduction, Internet, IDS, KNN classifier, Genetic Algorithm, KDDCUP'99.

1. Introduction

As the use of internet technology grows for the data communication, the network can compromised from different attack or threats. The information or network safety is becoming significant issue for any organization to preserve data and information in their computer network beside different types of attack with the help of resourceful and robust Intrusion Detection System (IDS). IDS can be developed using various machine learning techniques. IDS act as a classifier which classifies the data as normal or attack. Classification is a process of putting different categories of data together. Classification is one of the very common applications of the data mining in which similar type of samples are grouped together in supervised manner. An intrusion detection system can be classified into two categories [1]: network based and host based. The network attack also is of two types such as anomaly and misuse attack. The network based attacks are detected from the interconnection of computer systems. Since the system communicates with each other, the attack is sent from one computer system to another computer system by the way of routers and switches. The host based attacks are detected only from a single computer system and is easy to prevent the attacks. These attacks mainly occur from some external devices which are connected. The web based attacks are possible when systems are connected over the internet and the attacks can be spread into different systems through the email, chatting, downloading the materials etc. Nowadays many computer systems are affected from web based dangerous attacks. IDS are widely used area for the research and progress. This happens because detection of attack from the computer and network instead of IT security becomes major issue now-a-days. IDS efficiently and effectively detect the malicious activities on the network but the majority of existing system faces variety of challenges such as low detection rate and high false alarm rate. These problems happen due to the superiority of attack and intended similarities to normal behavior. In this paper, for the detection intrusion use KDDCUP'99 dataset [2]. KDDCUP99 Data set is an intrusion related data with almost 50 lacks samples. Ten percent of this data is publically available in UCI repository site for the experimental purpose of the researcher's. This optimum size of data contains samples for all 22 classes. A higher sample size data will require more computational resources which are not possible with simple desktop computers. So relatively low sample size data of KDD99 (10% of KDD) is used in this research work as raw material for developing a model. This data set contains about 5 million records as TCP/IP connection with 41 features, some of which are qualitative while others are continuous. Twenty two samples are categorized into five broader categories along with normal as DoS, R2L, U2R and Prob. Figure 1 illustrate the process for intrusion detection system.

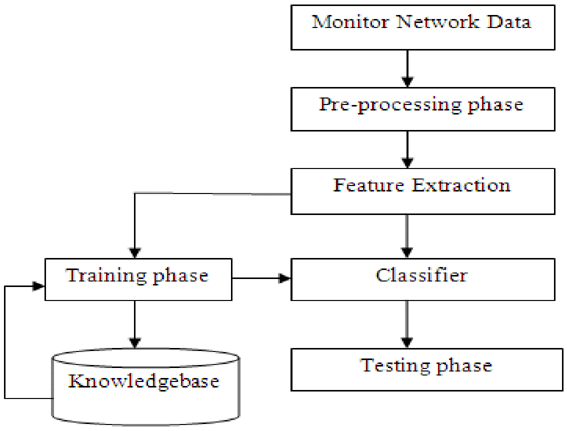


Fig.1 Process for the detection of intrusion detection system

The KDD'99 dataset may also get affected by several type of attack such as user to roots, denial of service, remote to local and probe [4].

- *Denial of Service (dos):* Attacker tries to prevent legitimate users from using a service.
- *Remote to Local (r2l):* Attacker does not have an account on the victim machine, hence tries to gain access.
- *User to Root (u2r):* Attacker has local access to the victim machine and tries to gain super user privileges.
- *Probe:* Attacker tries to gain information about the target host.

The KDD 99 intrusion detection benchmark consistsof three components, which are detailed in Table 1. In theInternational Knowledge Discovery and Data MiningTools Competition, only "10% KDD" dataset isemployed for the purpose of training [3]. This datasetcontains 22 attack types and is a more concise version ofthe "Whole KDD" dataset. It contains more examples ofattacks than normal connections and the attack types arenot represented equally. Because of their nature, denial ofservice attacks account for the majority of the dataset. Onthe other hand the "Corrected KDD" dataset provides adataset with different statistical distributions than either"10% KDD" or "Whole KDD" and contains 14 additional attacks.

Table 1 Fundamental features of KDD'99 intrusion detection dataset [4]

Dataset	DoS	Probe	u2r	r21	Normal
"10% KDD"	391458	4107	52	1126	97277
"Corrected KDD"	229853	4166	70	16347	60593
"Whole KDD"	3883370	41102	52	1126	972780

This paper presents the literature of the proposed techniques for the detection of the some of the serious intrusion which provides destruction of the network.

The organization of the rest of the paper is as follows: next section contains literature of the previous works done after that various intrusion detection techniques are discussed with their advantages and demerits and finally conclusion of the presented paper in last section.

2. Related Work

In this section discuss related work in current scenario intrusion detection technique using soft computing and data mining approach. In recent research trend soft computing and data mining play a vital role for intrusion detection. The role of data mining such as clustering classification and rule mining apply for detection of known and unknown type of attack, Instead of that soft computing implied in form of attribute and feature selection process in intrusion section system. Some work discuss here in current trend.

Heba F. Eid, Ashraf Darwish, Aboul Ella Hassanien and Ajith Abraham **"Principle Components Analysis and Support Vector Machine based Intrusion Detection System" [5]** proposed intrusion detection system by using Principal Component Analysis (PCA) with Support Vector Machines (SVMs) as an approach to select the optimum feature subset. We verify the effectiveness and the feasibility of the proposed IDS system by several experiments on NSL-KDD dataset. A reduction process has been used to reduce the number of features in order to decrease the complexity of the system. The experimental results show that the proposed system is able to speed up the process of intrusion detection and to minimize the memory space and CPU time cost.

S. Revathi and A. Malathi**“Network Intrusion Detection Using Hybrid Simplified Swarm Optimization and Random Forest Algorithm on NSL-KDD Dataset” [6]** proposed a new technique of combining swarm intelligence (Simplified Swarm Optimization) and data mining algorithm (Random Forest) for feature selection and reduction. SSO is used to find more appropriate set of attributes for classifying network intrusions, and Random Forest is used as a classifier. In the preprocessing step, we optimize the dimension of the dataset by the proposed SSO-RF approach and find an optimal set of features. SSO is an optimization method that has a strong global search capability and is used here for dimension optimization. The experimental result shows that the proposed approach performs better than the other approaches for the detection of all kinds of attacks present in the dataset.

ShafighParsazad, EhsanSaboori and Amin Allahyar**“Fast Feature Reduction in Intrusion Detection Datasets”** [7] proposed a very simple and fast feature selection method to eliminate features with no helpful information on them. Result faster learning in process of redundant feature omission. We compared our proposed method with three most successful similarity based feature selection algorithm including Correlation Coefficient, Least Square Regression Error and Maximal Information Compression Index. After that we used recommended features by each of these algorithms in two popular classifiers including: Bayes and KNN classifier to measure the quality of the recommendations. Experimental result shows that although the proposed method can’t outperform evaluated algorithms with high differences in accuracy, but in computational cost it has huge superiority over them.

L.PremaRajeswari, A. Kannan**“An Intrusion Detection System Based on Multiple Level Hybrid Classifier using Enhanced C4.5”[8]** detection system that uses a combination of tree classifiers which uses Enhanced C4.5 which rely on labeled training data and an Enhanced Fast Heuristic Clustering Algorithm for mixed data (EFHCAM). The main advantage of this approach is that the system can be trained with unlabelled data and is capable of detecting previously “unseen” attacks. Verification tests have been carried out by using the 1999 KDD Cup data set. From this work, it is observed that significant improvement has been achieved from the viewpoint of both high intrusion detection rate and reasonably low false alarm rate.

Asim Das and S. Siva Sathya**“Association Rule Mining for KDD Intrusion Detection Data Set” [9]** focused on the association rule mining in KDD intrusion dataset. Since the datasetconstitutes different kinds of data like binary, discrete & continuous data, same technique cannot be applied to determine theassociation patterns. Hence, this paper uses varying techniques for each type of data. The proposed method is used togenerate attack rules that will detect the attacks in network audit data using anomaly detection. Rules are formed dependingupon various attack types. For binary data, A-priori approach is used to eliminate the non-frequent item set from the rules andfor discrete and continuous value the proposed techniques are used.

LI Han **“Research of K-MEANS Algorithm based on Information Entropy in Anomaly Detection” [10]** used the unsupervised K-MEANS algorithm to model and detects anomaly activities. The aim is to improve the detection rate and decrease the false alarm rate. A K-MEANS algorithm based on information entropy (KMIE) is proposed to detect anomaly activities.KMIE can filter the outliers on the dataset to reduce thenegative impact, and indentify the initial cluster centers using entropy method. Then, KMIE can use these centers to iterative calculate and classify records into different clusters. This paper uses KDD CUP 1999 dataset to test the performance of KMIE algorithm. The results show that our method has a higher detection rate and a lower false alarm rate, it achieves expectant aim.

Devendrakailashiyaand R.C. Jain **“Improve Intrusion Detection Using Decision Tree with Sampling” [11]** presented the a method to improve accuracy Rate of intrusion detection using decision tree algorithm. Intrusion detection systems aim to identify attacks with a high detection rate and a low Error rate. In this paper we have supervised learning with preprocessing step for intrusion detection. We are using the stratified weighted sampling techniques to generate the samples from original dataset. These sampled applied on the proposed algorithm. The accuracy of proposed model is compared with existing results in order to verify the validity and accuracy of the proposed model. The results showed that the proposed approach gives better and robust representation of data. The experiments and evaluations of the proposed intrusion detection system are performed with the KDD Cup 99 dataset. The experimental results clearly show that the proposed system achieved higher Accuracy and Low Error in identifying whether the records are normal or attack one.

Yang Li and Li Guo**"An active learning based TCM KNN algorithm for supervised network intrusion detection" [12]** proposed a novel supervised network intrusion detection method based on TCM-KNN (Transductive Confidence Machines for K-Nearest Neighbors) machine learning algorithm and active learning based training data selection method. It can effectively detect anomalies with high detection rate, low false positives under the circumstance of using much fewer selected data as well as selected features for training in comparison with the traditional supervised intrusion detection methods. A series of experimental results on the well-known KDD Cup 1999 data set demonstrate that the proposed method is more robust and effective than the state-of-the-art intrusion detection methods, as well as can be further optimized as discussed in this paper for real applications.

3. Proposed Methodology

In this section presents our methodology for the detection of intrusion. The feature reduction of intrusion is applied on KDD'99 dataset using KNN classifier and GA algorithm. The overview of the methods uses is described below:

3.1 K- Nearest Neighbor (KNN)

A more sophisticated approach, k-nearest neighbor (kNN) classification is to find a group of k Patterns in the training set that are adjoining to the test pattern and bases the obligation of a label on the preponderance of a meticulous class in this neighborhood. This addresses, in many data sets, it is unlikely that one pattern will exactly match another, as well as the fact that conflicting information about the class of a pattern may be provided by the patterns closest to it. There are many key elements of this model [16]:

(1) The set of labeled patterns to be used for evaluating a test pattern's class,
(2) A distance or similarity metric that can be used to compute the closeness of patterns
(3) The value of k, the number of nearest neighbors and
(4) The method used to determine the class of the target pattern based on the classes and distances of the k nearest neighbors.

In its simplest form, KNN can involve assigning a pattern of the class of its nearest neighbor or of the majority of its nearest neighbors. Generally, KNN is a special case of instance-based learning and is also an example of a lazy learning technique, that is, a technique that waits until the query arrives to generalize beyond the training data. Although kNN classification is a classification technique that is easy to understand and implement, it performs well in many situations.Also, because of its simplicity, KNN is easy to modify for more complicated classification problems. For instance, KNN is particularly well-suited for multimodal classes as well as applications in which an object can have many class labels.

$$\mathrm{d}(\mathrm{x},\mathrm{y}) = \sqrt{\sum\nolimits_{\mathrm{i}=1}^{\mathrm{n}} (\mathrm{x}_\mathrm{i} - \mathrm{y}_\mathrm{i})^2}$$

where $x_i$is the i^{th} feature of the instance and is the total number of features in the data set. When all the attributes are of nominal, the distance can be measured as:

$$d(x,y) = \sum_{i=1}^{n} \delta(x_i, y_i)$$

where $\delta(x_i, y_i)$=0 if $x_i = y_i$ and $\delta(x_i, y_i)$=1, if $x_i \neq y_i$.

DARPA dataset that contains only network data is termed as KDDCup'99 dataset. It contains seven weeks of training data and two weeks of test data. KDD dataset is widely used as a benchmark dataset for offline network traffic, which helps the researchers to test and implement their algorithms.

3.2 Genetic Algorithm

Genetic algorithm is one of the components of evolutionary computation technique .A simple genetic algorithm may consist of a population generator and a selector, a fitness estimator and three genetic operators namely selection, mutation and crossover. The mutation operator inverts randomly chosen bits with a certain probability. The crossover operator combines parts of the species of two individuals, generates two new off springs, which are used to replace low fitness individuals in the population. After a certain number of generations, the search process will be terminated. A genetic algorithm (or GA for short) is a programming technique that mimics biological evolution as a problem-solving strategy. Given a specific problem to solve, the input to the GA is a set of potential solutions to that problem, encoded in some fashion, and a metric called a fitness function that allows each candidate to be quantitatively evaluated. These candidates may be solutions already known to work, with the aim of the GA being to improve them, but more often they are generated at random. The GA then evaluates each candidate according to the fitness function. In a pool of randomly generated candidates, we choose promising candidates toward solving the problem. These promising candidates are kept and allowed to reproduce. Multiple copies are made of them, but the copies may not perfect; random changes are introduced during the copying process. These digital off spring then go on to the next generation, forming a new pool of candidate solutions, and are subjected to a second round of fitness evaluation. Those candidate solutions which were worsened, or made no better, by the changes to their code are again deleted; but again, purely by chance, the random variations introduced into the population may have improved some individuals, making them into better, more complete or more efficient solutions to the problem at hand. Again these winning individuals are selected and copied over into the next generation with random changes, and the process repeats. The expectation is that the average fitness of the population will increase each round, and so by repeating this process for hundreds or thousands of rounds, very good solutions to the problem can be discovered. [13][14].

Methods of Change:

Once selection has chosen fit individuals, they must be randomly altered in hopes of improving theirfitness for the next generation. There are two basic strategies to accomplish this, they are.

(1) Mutation:By applying random changes to a single individualin the current generation to create a child.
(2) Crossover:By selecting vector entries, or genes, from a pairof individuals in the current generation and combines them toform a child.

In the classification of big data domains, sometimes hidden data possibility has been arise while the classification process. So generated features contains the false correlations, which is not up to the mark of finding the process of intrusion detection. The drawback of extra features is that it contains huge time for computing the process, and it impacts the accuracy of IDS. Here feature selection improves the more classification accuracy by searching for the best features, which best classifies the training data. So in the proposed approach probability has been calculated of the each individually attributes, then entropy has been calculated and finally information gain has been calculated for each every attributes separately. And here they applied some logical implies that if calculated gain is very less (gain<0.15) then that type of attribute will not be participated for the data preprocessing. So finally 18 attributes found whose gain was higher, that process is done in feature extraction and feature reduction.

The calculation process is done as follows:

Src_bytes	**Dst_bytes**	**Logged_in**	**Root_shell**	**count**	**Dst_host_count**	**Dst_host**	**Type**
200	3000	1	0	50	400	0	Normal
400	6000	1	0	50	400	0	Normal
200	6000	1	0	50	400	0	Normal
200	3000	1	0	0	400	0	Normal
200	6000	0	1	1	50	0	Normal
400	3000	0	1	1	50	0	Normal
200	3000	1	0	50	400	0	Normal
2000	0	0	0	600	400	0	Dos
2000	0	0	0	600	400	0	Dos

2000	3000	1	1	50	50	0	Dos
2000	6000	0	0	600	400	0	Dos
2000	0	0	0	600	400	0	Dos
50	0	0	0	50	50	0	Probe
50	0	0	0	50	50	0	Probe
50	3000	0	0	600	50	0	Probe
50	6000	1	1	1	400	0	Probe
50	0	0	0	50	50	0	Probe
3000	6000	0	1	1	50	0	R2l
3000	6000	0	1	1	50	0	R2l
3000	0	1	0	0	400	0	R2l

Now for example we have to find out attack type of tuple given below;

X = (Source_bytes= 200, dest_bytes= 3000, logged_in= 1, root_shell= 0, count= 50, dst_host_count= 400, dst_host_rerror_rate= 0).

P (*C*$_i$), the prior probability of each class, can be computed based on the training tuples:

P (type = normal) = 7/20 = 0.35
P (type = DOS) = 5/20 = 0.25
P (type = probe) = 5/20 = 0.25
P (type = R2L) = 3/20 = 0.15

To compute *P (**X**│C*$_i$), for *i* = 1, 2, 3, 4 we compute the following conditional probabilities:

P (source_bytes = 200│ type = normal) = 5/7 = 0.7143
P (source_bytes = 200│ type =DOS) = 0
P (source_bytes = 200│ type = probe) = 0
P (source_bytes = 200│ type = r2l) = 0

P (dst_bytes = 3000│ type = normal) = 4/7 = 0.57143
P (dst_bytes = 3000│ type = DOS) = 1/5 = 0.2
P (dst_bytes = 3000│ type = probe) = 1/5 = 0.2
P (dst_bytes = 3000│ type = r2l) = 0

P (logged_in = 1│ type = normal) = 5/7 = 0.7143
P (logged_in = 1│ type = DOS) = 1/5 = 0.2
P (logged_in = 1│ type = probe) = 1/5 = 0.2
P (logged_in = 1│ type = r2l) = 1/3 = 0.33

P (root_shell = 0│ type = normal) = 5/7 = 0.7143
P (root_shell = 0│ type = DOS) = 4/5 = 0.8
P (root_shell = 0│ type = probe) = 4/5 = 0.8
P (root_shell = 0│ type = r2l) = 1/3 = 0.33

P (count = 50│ type = normal) = 4/7 = 0.57143
P (count = 50│ type = DOS) = 1/5 = 0.2
P (count = 50│ type = probe) = 3/5 =0.6
P (count = 50│ type = r2l) = 0

P (dst_host_count = 400│ type = normal) = 5/7 = 0.7143
P (dst_host_count = 400│ type = DOS) = 4/5 = 0.8
P (dst_host_count = 400│ type = probe) = 1/5 = 0.2
P (dst_host_count = 400│ type = r2l) = 1/3 = 0.33

Using the above probabilities, we obtain

P(*X* │ *type= normal*) = P (Source_bytes = 200│ type = normal) ×P (dst_bytes = 3000│ type = normal) × P (logged_in = 1│ type = normal) × P (root_shell = 0│ type = normal) × P (count = 50│ type = normal) × P (dst_host_count = 400│ type = normal)
= 0.7143×0.57143× 0.7143× 0.7143 × 0.57143× 0.7143
= 0.085

To find the class, *C*$_i$, that maximizes *P*(***X***│*C*$_i$)*P*(*C*$_i$), we compute

***P*(*X* │ *type = normal*) P (type = normal) = 0.085 × 0.35 = 0.02975.**

Therefore, the naïve Bayesian classifier predict attack type = *Normal* for tuple ***X***.

After that entropy has been calculated as follows

Entropy H(X) = $-p_1\log_2 p_1 - p_2\log_2 p_2 - p_3\log_2 p_3$ **- ………………..**$-p_n\log_2 p_n$ **………………*eq(1)***

$$= -\sum_{i=1}^{m} pilog2\ pi$$

In the equation 1, the class-wise probability has been settled then entropy has been calculated of each individual attributes.

Then gain was calculated as follows:

Gain = Entropy(X) **-** Entropy (**X|Y**) **…………………………………………..…eq (2)**

So as per the above process feature reduction has been done, where gain was higher than that attribute has been qualified for the process and less gain was reduced from dataset.

ALGORITHM STEPS:

```
Step 1: Make X1 reduced datasets from a database.
Step 2: Set a learning algorithm to individual pattern for test dataset.
Step 3: Set a learning algorithm to individual pattern training dataset.
        svmStruct = svmtrain(X1(train(:,i1),:),groups(train(:,1))
Step 4: Object with unknown found to do with each of the X1 classifiers predictions.
Step 5: Select the most repeatedly predicted samples.
```

```
KNN steps:
Step1: Initialize population = X1
Step2: Apply genetic search into selected dataset
Step3: Apply KNN classifier for testing of all five data which is classified or misclassified data.
Step4: Each attribute will organize as their ranks.
Step5: Higher ranked attribute will select.
Step6: Apply knnclassify() on the each five subset of the attributes for enhance the accuracy level.
Step7: If knnga_classifier(class_knn)>knn_classifer(class_knn)
                data_class = class_knn;
        else
                data_class = class_id3knn;
Step8: Perform the reproduction
Step9: Apply crossover operator
Step10: Perform mutation then produce new population X'1
Step11: Calculate the local maxima for each category.
Repeat the steps till iteration is not finished
Step12: For each test X'1, start all trained base models then prediction of result by combining of all trained models, and separate the misclassified by optimized knn.
Classification: Majority occurrences
```

4. Experimental Results

The KDD99cup data set used for the purpose of experimental research analysis, as they know that KDD 99 dataset has been widely used for the evaluation of signature based intrusion detection. In the novel approach they have usedKDDCup'99 intrusion detection dataset, which contains 26167 records with.50:50 training ratio.

Attack types are four categories:

1. Denial of Service (DoS)
2. Remote to Local (R2l)
3. User to Root (U2R)
4. Probe

The proposed IDS has been implemented in MATLAB2012A tool and the machine configuration is Intel I3 core 2.20Ghz processor, with 4GB RAM, windows 7 home basis. The proposed methodology have first used the partially ID3 algorithm for the feature reduction from the KDD, the svm train function is use for training purpose of the trained sample, then knn is use for the clustering and classification process for the classify or misclassify.

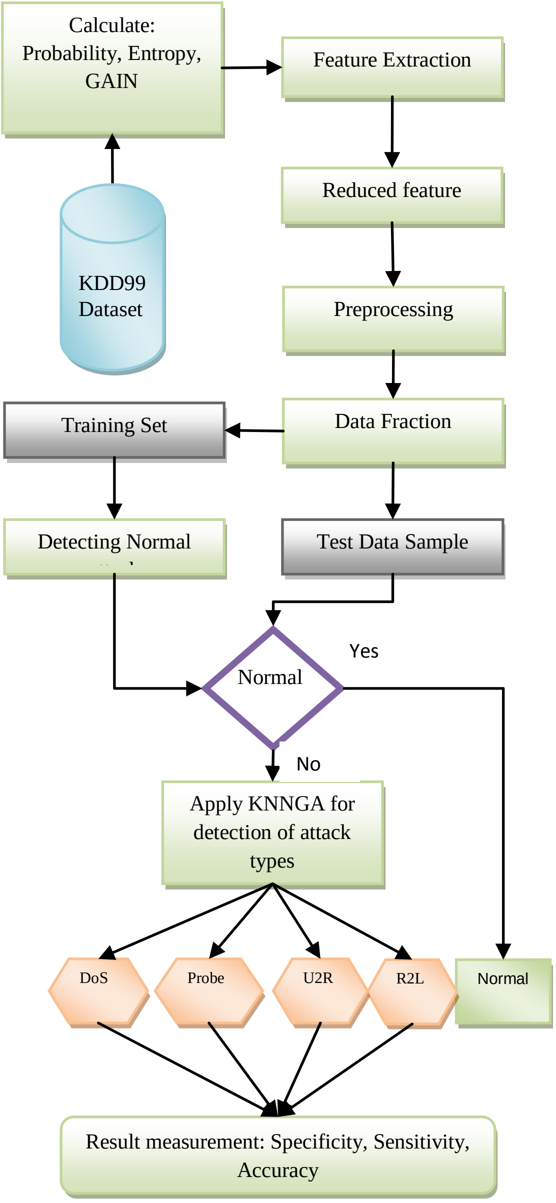


Fig. 2 Block diagram of proposed work

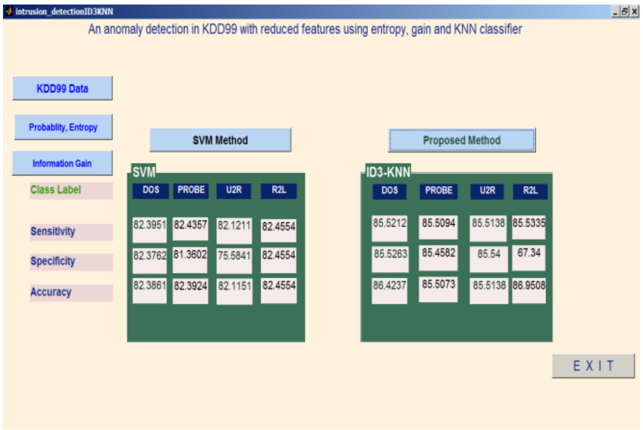
of the data, where GA is ensemble with KNN to enhance the best classification rate and optimized the result in very efficient manner. Here classification has five classes' data which is (normal, dos, u2r,r2l, and probe). This did classified or classified.

KNNGA classified data which were misclassified by alone SVM and KNN then applying KNNGA on multiple classifiers. This approach is focused on misclassified classifiers. Where GA is putted extra efforts to optimize best classified of the category until they are not accurately classified.

Then method has been tested on full (41 attributes) dataset as well as in reduced dataset (18 attributes), and used measurement parameters are:

Sensitivity, specificity and accuracy, and method is compared with SVM, KNN and found that proposed method produced most accurate result into maximum cases.

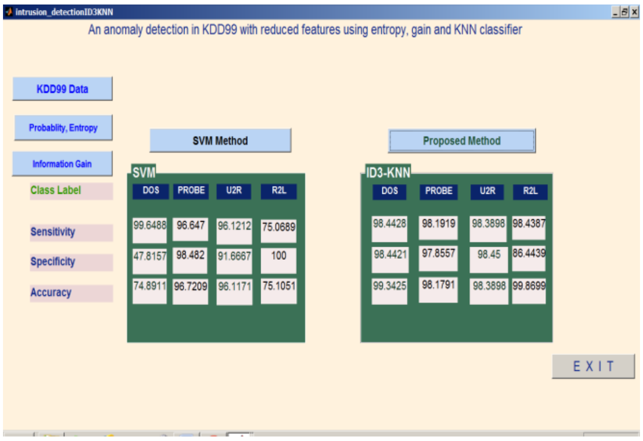
Here figure 4.1 shows that the main GUI environment of the implemented all methods along with proposed approach, here we clearly observe that the proposed approach yield more accurate output compared to the other previous developed methods and figure 4.2 shows the GUI environment for reduced feature set of 18 dataset.


intrusion_detectionID3KNN

An anomaly detection in KDD99 with reduced features using entropy, gain and KNN classifier

Class Label	SVM DOS	SVM PROBE	SVM U2R	SVM R2L	ID3-KNN DOS	ID3-KNN PROBE	ID3-KNN U2R	ID3-KNN R2L
Sensitivity	82.3951	82.4357	82.1211	82.4554	85.5212	85.5094	85.5138	85.5335
Specificity	82.3762	81.3602	75.5841	82.4554	85.5263	85.4582	85.54	67.34
Accuracy	82.3861	82.3924	82.1151	82.4554	86.4237	85.5073	85.5138	86.9508

Figure 4.1: Main GUI of Full dataset with 41 attributes


intrusion_detectionID3KNN

An anomaly detection in KDD99 with reduced features using entropy, gain and KNN classifier

Class Label	SVM DOS	SVM PROBE	SVM U2R	SVM R2L	ID3-KNN DOS	ID3-KNN PROBE	ID3-KNN U2R	ID3-KNN R2L
Sensitivity	99.6488	96.647	96.1212	75.0689	98.4428	98.1919	98.3898	98.4387
Specificity	47.8157	98.482	91.6667	100	98.4421	97.8557	98.45	86.4439
Accuracy	74.8911	96.7209	96.1171	75.1051	99.3425	98.1791	98.3898	99.8699

Figure 4.2: GUI of reduced feature set with 18 attributes

Table 4.2, 4.3, 4.4 illustrated that the sensitivity, specificity and accuracy comparison table of SVM and proposed KNNGA approaches for reduced 18 attributes, we have also examine the same scenario for the 41 full attributes, and there we found that the all methods gave the less accuracy level and taking much time as compare to reduced attribute.

Table 4.2: Sensitivity (18attribute)

Sensitivity (18attribute)			
	SVM	Proposed	
DOS	99.6488	98.4428	
PROBE	96.647	98.1919	
U2R	96.1212	98.3898	
R2L	75.06889	98.4387	

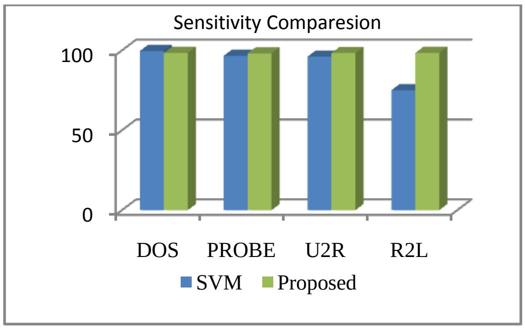


Figure 4.3: Sensitivity (18attribute)

Table 4.3: Specificity (18attribute)

Specificity(18attribute)			
	SVM	Proposed	
DOS	47.8157	98.4421	
PROBE	98.482	97.8557	
U2R	91.6667	98.45	
R2L	100	86.4439	

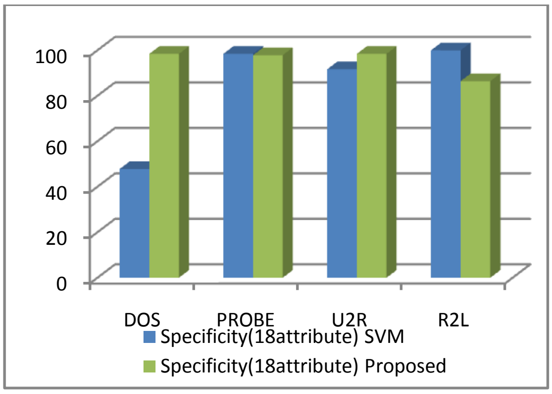


Figure 4.4: Specificity (18attribute)

Table 4.4: Accuracy (18attributes)

Accuracy(18attribute)			
	SVM	Proposed	
DOS	74.8911	99.3425	
PROBE	96.7209	98.1791	
U2R	96.1171	98.3898	
R2L	75.1051	99.8699	

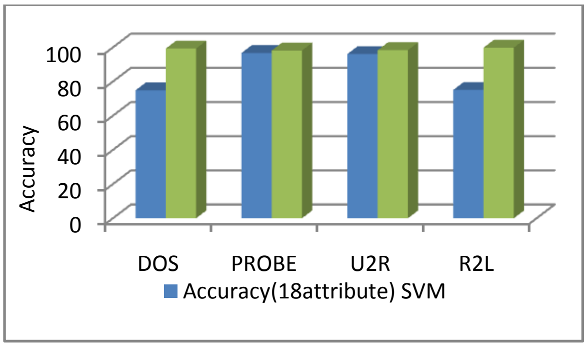


Figure 4.5: Accuracy (18attribute)

And in the figure 4.3, 4.4 and 4.5 they shows the bar graph analysis of all methods of the given table, as they clearly showing that the accuracy level of novel approach is generating more improved result.

5. Conclusion

The detection and prevention of the intrusion from the network is an important issues and it is not possible to violate security completely on using the existing approaches. The intrusion detection helps security organization accordingly by enhancing the efficiency and it is easy to use. In this paper we present an efficient method for intrusion detection using KNN (K-nearest neighbor) and genetic algorithm (GA). The KNN is used for the feature reduction among the large set of data then apply KNNGA together which improve the data categorization due this the performance increases. The analysis of the methodology is done using the KDDCUP'99 dataset. The comparison of our method and existing method SVM is evaluated using the performance metrics such as, sensitivity, specificity and accuracy. Among all obtained results our proposed methods is better than the other methods.

Reference

[1] S. Devaraju and S. Ramakrishnan "Performance Comparison For Intrusion Detection System Using Neural Network With KDD Dataset" ICTACT Journal On Soft Computing, April 2014, Volume: 04, Issue: 03743 ISSN: 2229-6956(Online).

[2] PratibhaSoni, Prabhakar Sharma "An Intrusion Detection System Based on KDD-99 Data using Data Mining Techniques and Feature Selection", International Journal of Soft Computing and Engineering (IJSCE), ISSN: 2231-2307, Volume-4 Issue-3, July 2014.

[3] S. Hettich, S.D. Bay, "The UCI KDD Archive" Irvine, CA: University of California, Department of Information and Computer Science, http://kdd.ics.uci.edu, 1999.

[4] H. GünesKayacık, A. NurZincir-Heywood, Malcolm I. Heywood "Selecting Features for Intrusion Detection: A Feature Relevance Analysis on KDD 99 Intrusion Detection Datasets".

[5] Heba F. Eid, Ashraf Darwish, Aboul Ella Hassanien and Ajith Abraham "Principle Components Analysis and Support Vector Machine based Intrusion Detection System", in proceeding of IEEE.

[6] S. Revathi and A. Malathi "Network Intrusion Detection Using Hybrid Simplified Swarm Optimization and Random Forest Algorithm on NSL-KDD Dataset" International Journal Of Engineering And Computer Science ISSN: 2319-7242 Volume 3 Issue 2 February, 2014 Page No. 3873-3876.

[7] ShafighParsazad, EhsanSaboori and Amin Allahyar "Fast Feature Reduction in Intrusion Detection Datasets" MIPRO 2012, May 21-25, 2012, Opatija, Croatia.

[8] L.PremaRajeswari, A. Kannan "An Intrusion Detection System Based on Multiple Level Hybrid Classifier using Enhanced C4.5" IEEE-International Conference on Signal processing, Communications and Networking Madras Institute of Technology, Anna University Chennai India, Jan 4-6, 2008. pp75-79.

[9] Asim Das and S. Siva Sathya "Association Rule Mining for KDD Intrusion Detection Data Set" International Journal of Computer Science and Informatics ISSN (PRINT): 2231 –5292, Volume-2, Issue-3, 2012.

[10] LI Han "Research of K-MEANS Algorithm based on Information Entropy in Anomaly Detection" 2012 Fourth International Conference on Multimedia Information Networking and Security.

[11] Devendrakailashiya and R.C. Jain "Improve Intrusion Detection Using Decision Tree with Sampling" International Journal of Computer Technology &Applications,Vol 3 (3), 1209-1216, ISSN:2229-6093

[12] Yang Li and Li Guo "An active learning based TCM-KNN algorithm for supervised network intrusion detection" computers & security 2 6 (2007) 4 5 9 – 4 6 7 in proceeding of Elsevier.

[13] Mark Crosbie and Gene Spafford. Applying genetic programming to intrusion detection. In Working Notes for the AAAI Symposium on Genetic Programming, pages 1–8. MIT, Cambridge, MA, USA: AAAI, 1995.

[14] DewanMdFarid, Mohammad ZahidurRahman, and ChowdhuryMofizurRahman. Adaptive intrusion detection based on boosting and naıve bayesian classifier. International Journal of Computer Applications, 24(3):12–19, 2011.