International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:9, No:8, 2015

A New DIDS Design Based on a Combination Feature Selection Approach

Adel Sabry Eesa, Adnan Mohsin Abdulazeez Brifcani, Zeynep Orman

***Abstract*—**Feature selection has been used in many fields such as classification, data mining and object recognition and proven to be effective for removing irrelevant and redundant features from the original dataset. In this paper, a new design of distributed intrusion detection system using a combination feature selection model based on bees and decision tree. Bees algorithm is used as the search strategy to find the optimal subset of features, whereas decision tree is used as a judgment for the selected features. Both the produced features and the generated rules are used by Decision Making Mobile Agent to decide whether there is an attack or not in the networks. Decision Making Mobile Agent will migrate through the networks, moving from node to another, if it found that there is an attack on one of the nodes, it then alerts the user through User Interface Agent or takes some action through Action Mobile Agent. The KDD Cup 99 dataset is used to test the effectiveness of the proposed system. The results show that even if only four features are used, the proposed system gives a better performance when it is compared with the obtained results using all 41 features.

***Keywords*—**Distributed intrusion detection system, mobile agent, feature selection, Bees Algorithm, decision tree.

I. Introduction

WITH the development of the Internet and its wide applications in all domains of people's life, intrusion detection has become a critical process in computer network security. An Intrusion Detection System (IDS) is defined [1] as a component that analyzes system and user operations in computer and network systems in terms of activities which are considered as undesirable from security perspectives. IDSs can be categorized into two methodologies as anomaly detection and misuse detection. Anomaly detection techniques [2] identify any unacceptable deviation from the expected behavior of an individual user or an application. The expected behavior is defined in advance for developed profiles by manually or automatically. This is then compared with the current activities of the user or the application. An uncharacteristic deviation would be an indication of an intrusion. On the other hand, misuse intrusion detection [3] refers to the analysis of certain well-defined patterns of attacks that exploit weaknesses in the system and the application software. For example, packets of network traffic could be analyzed for a series of characters, which could represent a signature of an attack sequence. This mechanism requires the knowledge of unacceptable behavior to detect an intrusion as opposed to anomaly detection which is based on the identification of normal behavior.

Recently, new approaches for developing Distributed Intrusion Detection Systems (DIDSs) which is based on Mobile Agents (MAs) are presented by many researchers [4]-[8]. MAs are particular software agents that have the capability to move from one host to another. MAs may offer unique features that can be used to improve the methods that are used for the design, development and deployment of the intrusion detection systems in the network.

Feature Selection (FS) has been a fertile field of computer science research and development since 1970's, and it is also used successfully in IDSs domain [9]-[16]. FS is a process of selecting an optimal subset of features among the existing features and it does not involve any feature transformation. Given a feature set of size n, the FS problem finds a minimal feature subset of size m ($m < n$) and still retains a suitably high accuracy for representing the original features. The objective of FS is to simplify a dataset by reducing its dimensionality and identify the relevant underlying features without sacrificing from the predictive accuracy. By doing that, it also reduces redundancy in the information provided by the selected features [17].

This paper presents a new design of DIDS based on a combination feature selection approach. In this approach, BA and DT are used as a basis for the feature selection process. BA is proposed to find the optimal subset of features, whereas DT is proposed as a judgment for the selected features. The performance of the presented system is evaluated by using KDD Cup 99 dataset, the benchmark dataset commonly used by IDS researchers.

The organization of this paper is as follows: Section II presents a background of bees algorithm and decision tree. The proposed DIDS and its architecture, the rule generator and feature selection approach are discussed in Section III. Section IV details the evaluation criteria to test the performance of the proposed approach. Section V reports the experimental results of the proposed system and a brief discussion on the obtained results. Finally, the conclusion is stated in Section VI.

II. Bees Algorithm and Decision Trees

A. Bees Algorithm

The Bees algorithm is a population-based optimization algorithm inspired by the foraging behavior of bees and is developed in 2005 [18]. Bees search for food by using scouts to explore sites deemed most likely to produce favorable

Adel Sabry Eesa is with the Department of computer science in Zakho University, Duhok City, KRG-Iraq (e-mail: adelsabryissa@gmail.com).
Adnan Mohsin Abdulazeez Brifcani is Presidency of Duhok Polytechnic University, Duhok City, KRG-Iraq (e-mail: president@dpu.ac).
Zeynep Orman is with the Department of Computer Engineering, Faculty of Engineering, Istanbul University, 34320, Avcilar, Istanbul, Turkey (e-mail: ormanz@istanbul.edu.tr).

results. At first, the scouts conduct random searches to locate the sites where food exists in the greatest abundance. Then the fitness values of the sites that are visited by the scout bees are evaluated and the bees that have the highest fitness value are chosen as "selected bees" and the corresponding sites are chosen for neighborhood search. The scout bees go back to the sites with follower bees that were waiting inside the hive and the follower bees are sent to more promising sites. This allows the colony to gather food more quickly and efficiently. Fig. 1 illustrates the pseudo-code for a simple BA.

1. *Initialize population with random solutions.*
2. *Evaluate fitness of the population.*
3. *While (stopping criterion not met) //Forming new population.*
4. *Select elite bees.*
5. *Select sites for neighborhood search.*
6. *Recruit bees for selected sites (more bees for best e sites)and evaluate fitness.*
7. *Select the fittest bee from each patch.*
8. *Assign remaining bees to search randomly and evaluate their fitness.*
9. *End While.*

Fig. 1 Pseudo code of the Bees Algorithm

B. Decision Tree

A decision tree [19] is a tree data structure in which internal nodes contain tests on attribute values, and leaves have assigned class labels. It is initially constructed from a set of pre-classified data. The main approach is to select the attributes, which best divides the data into their classes. According to the values of these attributes, the data items are partitioned. This process is recursively applied to each partitioned subset of the data. The process terminates when all the data in current subset belongs to the same class. A node of a decision tree specifies an attribute by which the data is to be partitioned. Each node has a number of edges, which are labeled according to a possible value of the attribute in the parent node. An edge connects either two nodes or a node and a leaf. Leaves are labeled with a decision value for categorization of the data. The main problem is to decide on the attribute, which will best partition the data into various classes. The most widely used algorithms include ID3, C4.5 [20] which uses the information gain approach to solve this problem. Information gain uses the concept of entropy, which measures the impurity of data.

Once the decision tree is built, its potential classification accuracy is calculated using the test data. Each test record is compared with the test data at the root of the tree, then passed down to one of the branches depending on the outcome of the evaluation. This process is repeated until the record reaches a leaf. The class label of that leaf is used as the predicted classification for that record. Based on the train data set, a tree is constructed as a set of rules. These rules can be represented as sets of if-then rules to improve human readability.

III. Proposed Design of DIDS

The architecture of the proposed DIDS is consists of six agents as shown in Fig. 2: Capture Agent (CA), Data Collector Mobile Agent (DCMA), Rule and Feature Generator (RFGA), Decision Making Mobile Agent (DMMA), Action Mobile Agent (AMA), and User Interface Agent (UIA). The responsibilities of each agent are described as follows:

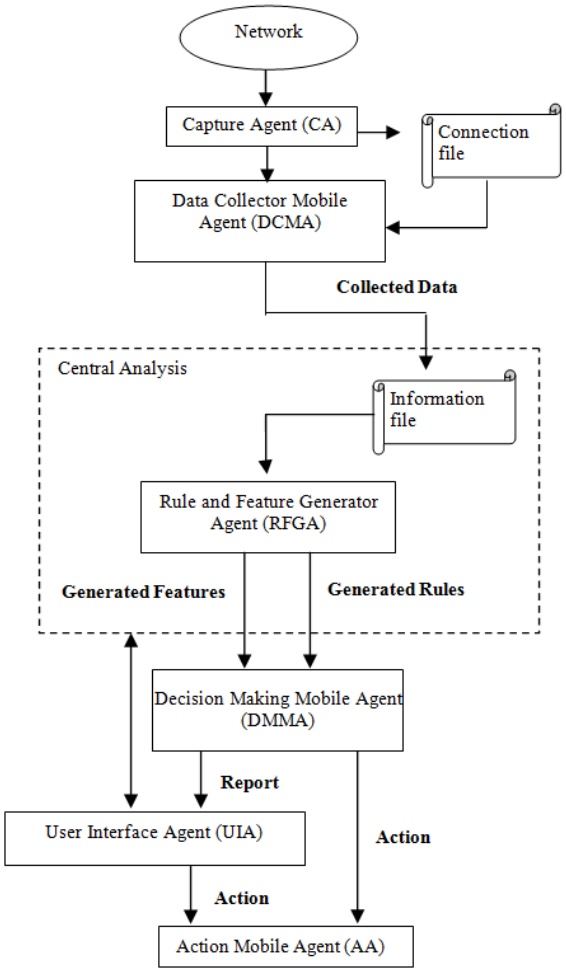

Fig. 2 Architecture of the proposed DIDS

Capture Agent (CA): The main task of this agent is to gather information about the incoming packages from the Internet and store this information on a file called as the connection-file. This task will occur in each node.

Data Collector Mobile Agent (DCMA): The task of this agent is to collect the information from connection-file that is previously stored by CA agent. DCMA will migrate through the networks moving from node to node in order to collect the information about network traffic from each node until reaching the central analysis computer and storing its information about visited node on a file called information-file.

Rule and Feature Generator Agent (RFGA): This agent is used to generate a subset of features with the corresponding

rules by using the combination of the Bees algorithm and the decision Tree. BA is used as the feature generator whereas DT is used as a criterion on the generated features. For more detailed information, see our previous work in [21]. The main steps of this agent are described in Fig. 3.

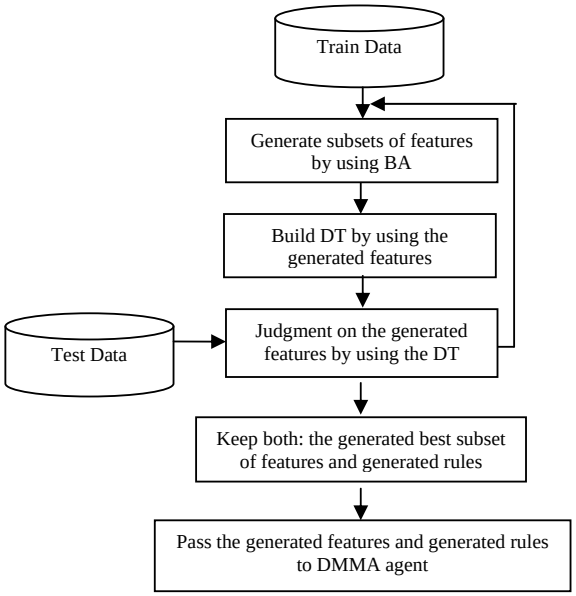

Fig. 3 Main steps of the RFGA agent

The proposed agent starts by generating several subsets of features from information-file using BA. The quality of generated subsets will be evaluated by using DT. These two steps will repeated until finding the best subset of features. The best features with their corresponding rules will be kept and passed to DMMA.

- Decision Making Mobile Agent (DMMA): DMMA takes the generated decision rules and the generated features from the RFGA and applying them on each node through its migration in order to detect attacks. When DMMA observes that there is an attack on one of the nodes, then it will alert the user through UIA. The user then can take an appropriate action through AMA.
- User Interface Agent (UIA): this agent interacts with the users to start or shutdown the system and interprets the intrusion information and alarms. When intrusion is detected by DMMA, it will report and alert the system administrator. Then the administrator can decide to take some action through the AMA.
- Action Mobile Agent AMA: this agent receives its order from the administrator or the DMMA and it generates passive or reactive responses to different attacks.

IV. Evaluation Criteria

To rank the different obtained results, a cost matrix (C) is defined as in [23]. With the given cost matrix illustrated in Table I and the confusion matrix which is obtained by a subsequent empirical testing process, a Cost per Test (CPT) value is calculated by using:

$$CPT = \frac{1}{N}\sum_{i=1}^{m}\sum_{j=1}^{m} CM(i,j) * C(i,j) \quad (1)$$

where *CM* and *C* are the confusion matrix and the cost matrix, respectively. *N* represents the total number of test instances, and *m* is the number of the classes in the classification. An entry at row *i* and column *j*, *CM* (*i*, *j*), represents the number of misclassified instances that originally belong to class *i*, although incorrectly identified as a member of class *j*. The entries of the primary diagonal, *CM* (*i*, *i*), stand for the number of properly detected instances.

The accuracy is based on the percentage of successful predictions on the test data set, which is given by:

$$AR = \frac{No.\ of\ correctly\ classified\ instance}{Total\ No.\ of\ instance\ in\ the\ test\ dataset} * 100\% \quad (2)$$

Higher values of *AR* and lower values of *CPT* show better classification for the intrusion detection system. Detection Rate (DR) which is given by (3) is the ratio of the number of correctly classified instances as an attack to the total number of this attack in the test dataset. In this paper, the DR, AR and CPT measures are used to rank the different results.

$$DR = \frac{number\ of\ correctly\ classified\ instances\ as\ an\ attack}{total\ number\ of\ this\ attack\ in\ test\ dataset} * 100 \quad (3)$$

TABLE I
Cost Matrix

	Normal	Probing	DoS	U2R	R2L
Normal	0	1	2	2	2
Probing	1	0	2	2	2
DoS	2	1	0	2	2
U2R	3	2	2	0	2
R2L	4	2	2	2	0

V. Results

The simulations have been carried out by using three computers. One of the computers is used as a central analysis and the other two computers are used as the networks that the mobile agents will migrate through. The KDD Cup 99 train and test data subsets which are depicted in Table II are used by the RFGA agent to produce the optimal subset of features. The produced features contain only four of the features as {f_3, f_{30}, f_{32}, f_{34}}, these features are chosen after many experiments. In Table II, the number of training data is 4947 and the number of test data is 3117, which are selected randomly from the 10%KDD Cup 99 dataset. To keep the proportion of each attack both in the extracted train and in the test datasets, each attack is divided by 100.

To test the proposed system, the complete 10%KDD Cup 99 train dataset is used by the RFGA to build and generate the decision tree rules from the four produced features; In order to simulate the distributed environment, the KDD Cup 99 test dataset is distributed into two data sets equally which simulates the instances collected from different CA. The complete 10%KDD Cup 99 train and test datasets are shown

in Table III. The classification task is to classify each connection record in test dataset to one of the five classes [22] that are considered in the KDD Cup 99 dataset as Normal, Probing, DoS, U2R and R2L.

TABLE II
DIFFERENT ATTACK TYPES AND THEIR CORRESPONDING OCCURRENCE NUMBER RESPECTIVELY IN THE EXTRACTED TRAIN AND TEST DATASET

Normal (973;606)	
Probing (41; 42)	psweep(12;3),Mscan(0;11),Nmap(2;1) Portsweep(11;4), Saint(0;7),Satan(16;16).
DoS(3915 ; 2299)	apache2(0;8),back(22;11),land(0; 0), mailbomb(0;50),Neptune(1072;580), processtable(0;8),Pod(3;1), udpstorm(0;0),Smurf(2808;1641), Teardrop(10;0),
U2R(5 ; 10)	buffer_overflow(3;1), httptunnel(0;3), loadmodule(0;0),perl(0;0), rootkit(2;2),xterm(0;2), Ps(0;2),Sqlattack(0;0),
R2L(13; 160)	ftp_write(0;0),imap(0;0), guesspasswd(2;44),named(0;0), multihop(0;0),phf(0;0), sendmail(0;0),snmpgetattack(0;77), snmpguess(0;24),spy(0;0,) warezclient(10;0),worm(0;0), warezmaster(1;15),xsnoop(0;0), xlock(0;0),

TABLE III
COMPLETE 10%KDD CUP 99 TRAIN AND TEST DATASETS

Normal(97,277; 60,593)	
Probing (4, 107; 4, 176)	ipsweep(1, 247; 306),mscan(0; 1, 053), nmap(231; 84),portsweep(1, 040; 364), saint(0; 736),satan(1, 589; 1, 633).
DoS(391, 458; 229, 853)	apache2(0; 794), back(2, 203; 1.098), land(21; 9), mailbomb(0; 5, 000), neptune(107, 201; 58, 001), pod(264; 87), processtable(0; 759), smurf(280, 790; 164, 091), teardrop(979; 12), udpstorm(0; 2).
U2R(52; 228)	buffer overflow(30, 22), httptunnel(0; 158),loadmodule(9; 2), perl(3; 2),perl(3; 2), ps(0; 16), rootkit(10; 13), sqlattack(0; 2), xterm(0; 13).
R2L(1, 126; 16, 189)	ftp write(8; 3), imap(12; 1), multihop(7; 18), named(0; 17), phf(4; 2),sendmail(0; 17), snmpgetattack(0; 7, 741), guess passwd(53; 4, 367), snmpguess(0; 2, 406), spy(2; 0), warezclient(1, 020; 0), warezmaster(20; 1, 602), worm(0; 2), xlock(0; 9), xsnoop(0; 4).
Total Train data set = 494020 Total Test data set = 311039	

Tables IV and V present the confusion matrix related with the DR, AR, and CPT which are obtained by using the produced four features and the complete 41 features, respectively. The obtained results show that by using only four of the features, the proposed system gives better results in terms of AR and CPT when compared with the obtained results by using the complete 41 features. By the evaluation of DR, there is no significant difference between the two experiments for the DoS attack. For the Normal, Probing attacks, the use of 41 features gives better results with small differences than the use of four features whereas for the R2L attacks, the use of four features gives better results. For U2R, the worst result is obtained when using the four features. This is because the number of records of U2R attacks in train dataset is very little. Also there is some attack in test dataset is not included in train dataset which makes train process is very difficult to learn these attacks.

TABLE IV
CONFUSION MATRIX RELATED TO THE DR, AR, AND CPT USING THE COMPLETE 41 FEATURES

Predicted Actual	Normal	Probing	DoS	U2R	R2L	%DR
Normal(60591)	60223	243	109	9	5	99.4
Probing (4166)	601	2862	700	0	3	68.7
DoS (229853)	7124	300	222431	0	0	96.77
U2R (228)	191	0	0	36	1	15.8
R2L (16189)	15646	13	514	11	5	0.03
41 features	*AR* = 91.811%			*CPT* = 0.2613		

TABLE V
CONFUSION MATRIX RELATED TO THE DR, AR, AND CPT USING THE FOUR FEATURES

Predicted Actual	Normal	Probing	DoS	U2R	R2L	%DR
Normal(60591)	60060	95	290	0	146	99.12
Probing(4166)	479	2824	748	0	115	67.78
DoS(229853)	7305	0	222482	0	66	96.79
U2R(228)	165	20	24	0	19	0
R2L(16189)	10626	5	4104	0	1454	8.98
4 features	*AR*= 92.2171%			*CPT* = 0.22267		

VI. CONCLUSION

In this study, we have investigated the new design of Intrusion Detection Systems based on the combination of BA and DT and evaluated its performance based on the benchmark KDD Cup 99 intrusion data. First, we have designed RFGA agent which is uses BA to generate features while it uses DT as measurement on generated features. Then we investigate the migration of DMMA agent in a physical environment to detect attacks. Empirical results reveal that the used only the four produced feature is performed better when compared with the results using the whole 41 features in term of AR and CPT.

For DR, the worst result is obtained with U2R attack by using the four produced feature. This is because there are some instances of attacks in the test dataset that are never appeared in the train dataset.

Using BA as a rule generator can be suggested as a futures work. Moreover the use of other techniques with BA instead of using DT remains an open issue.

REFERENCES

[1] Revision by Tzeyoung Max Wu, *Information Assurance Technology Analysis Center (IATAC)*, Information Assurance Tools Report – Intrusion Detection Systems,6th ed. 2009.

[2] V. Jyothsna, V. V. Ramaprasad, K. M. Prasad, *A Review of Anomaly based Intrusion Detection Systems*, International Journal of Computer Applications, vol. 28, no.7, pp. 26-35, 2011.

[3] S. R. Sriram, K. C,Vijaya, *An Overview of Intrusion Detection Systems*, IDT Workshop on Interesting Results in Computer Science and Engineering (IRCSE 9), Malardalen University, Sweden,2009.
[4] R. Sasikumar, D. Manjula,*A Distribulated Intrusion Detection System Based on Mobile Agents with Fault Tolerance*, European Journal of Scientific Research, vol. 62 no.1, pp. 48-55, 2011.
[5] S.Manmeet, S. S. Sodhi, *Distributed Intrusion Detection using Aglet Mobile Agent Technology*, Proceedings of National Conference on Challenges & Opportunities in Information Technology (COIT-2007), RIMT-IET, Mandi Gobindgarh. March 23, 2007.
[6] B Imen, B. Y. Sadok, P. Pascal, *MAD-IDS: Novel Intrusion Detection System Using Mobile Agents and Data Mining Approaches*, Intelligence and Security Informatics, Lecture Notes in Computer Science, Springer, vol. 6122/2010, pp. 73–76, 2010.
[7] G. Donald, Marks, M. Peter, S. Michael,*Optimizing the Scalability of Network Intrusion Detection Systems Using Mobile Agents*, Journal of Network and Systems Management, Springer, vol. 12, no. 1, pp. 95-110, 2004.
[8] E.Mohamad, *A New Mobile Agent-Based Intrusion Detection System Using Distributed Sensors*, In proceeding of FEASC, 2004.
[9] V. Bolon-Canedo, N. Sanchez-Marono, A. Alonso-Betanzos,*Feature selection and classification in multiple class datasets: An application to KDD Cup 99 dataset*, Expert Systems with Applications, Elsevier, vol. 38, no. 5, pp. 5947-5957, 2011.
[10] L.Shih-Wei, Y.Kuo-Ching, L.Chou-Yuan, L.Zne-Jung,*An intelligent algorithm with feature selection and decision rules applied to anomaly intrusion detection*, Applied Soft Computing, Elsevier, vol. 12, no. 10, pp. 3285-3290, 2012.
[11] T.Chi-Ho, K.Sam, W.Hanli,*Genetic-fuzzy rule mining approach and evaluation of feature selection techniques for anomaly intrusion detection*, Pattern Recognition, Elsevier, vol. 40, no. 9, pp. 2373-2391, 2007.
[12] L.Jean-Louis, R.Ryan, S.Stephen, M.Srinivas, *Signature Based Intrusion Detection using Latent Semantic Analysis*, IEEE World Congress on Computational Intelligence, Neural Networks, 2008. IJCNN, pp 1068-1074, 2008.
[13] T. N.Hai, F.Katrin, P.Slobodan,*Towards a Generic Feature-Selection Measure for Intrusion Detection*, International Conference on Pattern Recognition (ICPR), IEEE, pp. 1529-1532, 2010.
[14] N. P.Neelakantan, C.Nagesh, M.Tech,*Role of Feature Selection in Intrusion Detection Systems for 802.11 Networks*, International Journal of Smart Sensors and Ad Hoc Networks (IJSSAN), vol. 1, no. 1, pp. 98-101, 2011.
[15] R.Mohanabharathi, Mr T. Kalaikumaran, Dr.S.Karthi,*Feature Selection for Wireless Intrusion Detection System Using Filter and Wrapper Model*, International Journal of Modern Engineering Research (IJMER), vol.2, no. 4, pp. 1552-1556, 2012.
[16] D.Rupali, L.Shilpa,*Performance Comparison of Features Reduction Techniques for Intrusion Detection System*, International Journal of Computer Science and Technology (IJCST), vol. 3, no. 1, 2012.
[17] E. B.Mohammad, G-A Nasser, H. A.Mehdi,*Using Ant Colony Optimization-Based Selected Features for Predicting Post-synaptic Activity in Proteins*, EvoBIO 2008. LNCS, Springer, vol. 4973, pp. 12-23, 2008.
[18] D. T.Pham, A.Ghanbarzadeh, E.Koc, S.Otri, S.Rahim, M.Zaidi,*The Bees Algorithm.Technical Note*, Manufacturing Engineering Centre, Cardiff University, UK.
[19] L.Steven, Salzberg,*Book Review: C4.5: Programs for Machine Learning* by Ross Quinlan. Morgan Kaufmann Publishers, Inc., 1993, Machine Learning, Springer vol. 16, no. 3, pp. 235-240, 1993.
[20] R. Quinlan,*C4.5: Programs for Machine Learning*, Morgan Kaufmann Publishers Inc, San Francisco, CA, USA, 1993.
[21] Adel Sabry Eesa, Zyenep Orman, Adnan Mohsin Abdulazeez,*A New Feature Selection Model Based on ID3 and Bees Algorithm for Intrusion Detection System*, Turkish Journal of Electrical Engineering and Computer Sciences, olv. 23, no. 2, pp. 615-622, 2015.
[22] P.Sandhya, A.Ajith, G.Crina, T.Johnson,*Modeling intrusion detection system using hybrid intelligent systems*, Journal of Network and Computer Applications, Elsevier, vol. 30, no. 1, pp 114-132, 2007.
[23] E.Charles,*Results of the KDD'99 Classifier Learning*, SIGKDD Explorations, ACM SIGKDD Explorations Newsletter, vol. 1, no. 2, pp. 63-64, 2000.