OPEN ACCESS

International
Journal
Of Modern Engineering Research (IJMER)

Classification Rule Discovery Using Ant-Miner Algorithm: An Application Of Network Intrusion Detection

J. Uthayakumar[1], D. Nivetha[2], D.Vinotha[3], M.Vasanthi[4]
[1]*Apollo Computer Education, Puducherry, India*
[2,3,4] *Department of CSE, Sri Manakula Vinayagar Engineering College, Puducherry, India*

Abstract: *Enormous studies on intrusion detection have widely applied data mining techniques to finding out the useful knowledge automatically from large amount of databases, while few studies have proposed classification data mining approaches. In an actual risk assessment process, the discovery of intrusion detection prediction knowledge from experts is still regarded as an important task because experts' predictions depend on their subjectivity. Traditional statistical techniques and artificial intelligence techniques are commonly used to solve this classification decision making. This paper proposes an ant-miner based data mining method for discovering network intrusion detection rules from large dataset. The obtained result of this experiment shows that clearly the ant-miner is superior than ID3, J48, ADtree, BFtree, Simple cart. Although different classification models have been developed for network intrusion detection, each of them has its strength and weakness, including the most commonly applied Support Vector Machine(SVM)method and the clustering based on Self Organized Ant Colony Network (CSOACN).Our algorithm is implemented and evaluated using a standard bench mark KDD99 dataset. Experiments show that ant-miner algorithm out performs than other methods in terms of both classification rate and accuracy.*

Keywords: *Intrusion Detection, Ant-miner, Artificial Intelligence, Cross validation*

I. INTRODUCTION

In today's information system management, large-scale data clustering and classification have become increasingly important and challenging area. As a particular application area, Intrusion Detection Systems (IDSs) are designed to defend computer system from various cyber attacks and computer viruses. There are two primary assumptions in the research of intrusion detection: (1) user and program activities are observable by computer systems and (2) normal and intrusion activities must have distinct behaviors.

1.1 Data-mining based approaches for IDSs

Researchers have proposed an implemented various models that different measures of system behavior. As it is an energy and time consuming job for security experts to update current IDSs frequently by manual encoding, using data mining approaches to network intrusion detection provides an opportunity for IDSs to learn the behaviors of networks automatically by analyzing the data trials of their activities. Two key advantages of using a data mining approach to IDSs (1) It can be used to automatically generate the detection models for IDSs, so that new attacks can be detected automatically as well. (2) It is general, so it can be used to build IDSs for a wide variety of computing environments. The central theme of data mining approaches is to take a data-centric point of view and consider intrusion detection as a analysis process. This includes four essential steps.

(1) Capturing packets transferred on the network.
(2) Extracting an extensive set of features that can describes network connection or a host session.
(3) Learning a model that can accurately describe the behavior of abnormal and normal activities by applying data mining activities.
(4) Detecting the intrusions by using the learnt models.

We assume that Step (1) and (2) have been developed and are already available for the further training and testing phases. Step (3) in data mining, in general, is by classification, link analysis, and sequence analysis. In the rest of the paper, we will use SVM to denote either the concept or the algorithm when there is no confusion.

1.2 Motivation and Contribution:

Support Vector Machine (SVMs) have been widely accept as a powerful data classification method. On the other hand, the Self –Organized Ant Colony Network (CSOACN) has been shown to be efficient in data clustering (Section 5). Our work aims to be developed an algorithm that combines the logic of both methods to produce a high-performance IDS.One challenge of developing IDSs is to realize real-time detection in high-speed networks. The machine-learning-based SVM is a good choice for learning with a little volume of data. Clustering in intrusion detection is used to resolve the multiple classification problems. The main contribution of this paper includes the following.

(1) Modifications to the supervised learning SVM and the unsupervised learning CSOACN so they can be used interactively and efficiently.
(2) A new algorithm, CSVAC, that combines the modified SVM and CSOACN to minimize the training dataset while allowing new data points to be added to the training set dynamically. The idea of combining supervised learning and unsupervised learning was applied previously.

1.3 Related work

Issues related to intrusion detection can be categorized into two broad cases(1) network security and intrusion detection models and (2) intrusion detection methods and algorithms based on artificial intelligence(mostly machine learning) techniques. In this section we shall briefly review some related works in the second area, and leave area (1) to the next section, when we discuss the background of IDSs .Intrusion detection has been studied for decades using machine learning techniques, including traditional classification methods such as K-Nearest Neighbor (K-NN), Support Vector Machine (SVMs), Decision Trees (DTs).Bayesian, Self –Organized Maps (SOMs), Artificial Neural Networks (ANNs), Generic Algorithm (GAs). A review of using these approaches, was given, which also included in statistics of the use of techniques reported in 55 research articles during the period 2000-2007. Another more recent review provided thorough survey of intrusion detection using computational intelligence. Most recently, an IDSs was introduced by integrating On Line Analytical Processing (OLAP) tools and data mining techniques. It is shown that the association of the two fields produces a good solution to deal with defects of IDSs such as low detection accuracy and high falls alarm rate. As one of the swarm intelligence approaches, Ant Colony Optimization (ACO, has been applied in many fields to solve optimization problems, but its application to the intrusion detection domain is limited. The basic ingredient of their ACO algorithm was a heuristic for probabilistically constructing solutions. Hybrid intrusion detection approaches involving SVM have been studied in the past , that uses Dynamically Growing Self-Organizing Tree (DGSOT) algorithm for clustering to help in finding the most qualified points to train the SVM classifier. Another hybrid intrusion detection approach was recently detected that combines the hierarchical clustering and SVM .The purpose of using the hierarchical clustering algorithm is to provide the SVM classifier with fewer but higher quality training data that may reduce the training time and improve the performance of the classifier .WE use ACO to achieve the goal that is capable of updating the models without a retraining process ,as explained in the previous section about motivations.

1.4 Background

In this section, we present some background knowledge about IDSs. We begin with the introduction of basic concepts and technologies of network security.

1.4.1 Network security

There are three basic security concerns that are important to information on a network.

- Confidentiality: Loss of confidentiality results when information is read are copied by unauthorized users.
- Integrity: Loss by integrity results when information is modified in unexpected ways.
- Availability: Loss of availability results when information can be erased or become inaccessible by authorized users. On the other hand, there are three security concepts that are related to the people who use the information.
- Authentication: Proving whether a user is who he/she claims to be .
- Authorization: Determining whether a particular user has the privilege to perform a certain action.
- Non-repudiation: Providing protection against an individual falsely denying having performed a particular action.

1.4.2 Network intrusion detection:

Intrusion detection is the detection of actions that attempt to compromise the integrity, confidentiality, or availability of a resource. It attempts to detect attacks by examining various data records observed through processes on the same network. Data records are split into two categories host -based data which is audit data that record all system calls in chronologically sorted order; and network- based data, which is the network traffic data. As one of the promising network security technologies, Intrusion Detection Systems (IDSs) detect a possible intrusion as soon as possible and take appropriate actions .An IDSs is a reactive rather than pro-active agent. This paper adapts an advanced AI technique, ACO based ant-miner algorithm into network intrusion detection problem.

II. ANT-MINER ALGORITHM FOR CLASSIFICATION RULE DISCOVERY (CRD)

2.1 Rule structure

In order to apply the ant-miner algorithm to the problem of classification learning, the classification rule structure can be expressed in the form of IF-THEN as follows:

IF <Predecessor > THEN < Successor >

The predecessor part of the rule consists of several conditions, usually connected by a logical conjunction operator AND. The successor part of the rule shows the classification conclusion. For example, to express a intrusion detection class, a corresponding rule can be expressed as below:

Rule 1: IF dst_host_str_diff_host_rate<=0.035 AND
hot<=25.0 AND
count<=29.5
THEN Class normal

These kinds of rules are based on practical problem-solving knowledge which provides necessary and sufficient conditions for achieving certain goals.

2.2 Data structure of rule discovery

The inspiring source of the data structure for discovering rules comes from the foraging path of real ant colonies. In Fig. 1 there is a path taken by an ant which is represented with nodes connected by blue lines: Start->$Val_{1,2}$->$Val_{2,1}$->$Val_{3,3}$-> C_3->End. It forms an IF-THEN rules give below. The attributes and selected attribute values forms the predecessor part, and the Class and the Class value forms the successor part. This rule can be expressed as: IF $Attribute_1 = Val_{1,2}$ AND $Attribute_2 = Val_{2,1}$ AND $Attribute_3 = Val_{3,3}$THEN Class = C_3. To generate a discovered rule, it should be ensured that enough ants take the same path. The method to obtain a discovered rule will be proposed in Section C.

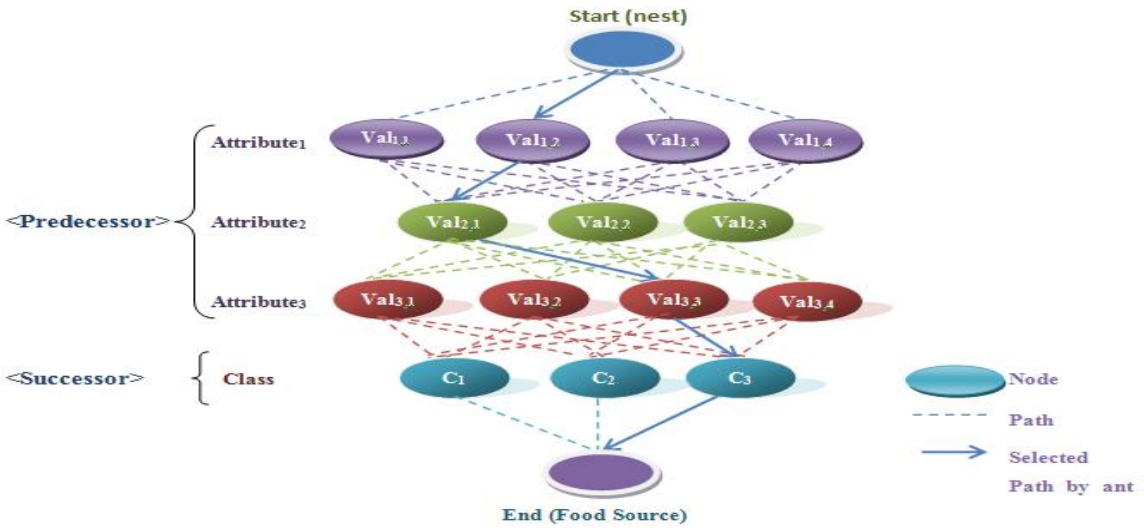

Fig. 1 Data structure for classification rule discovery

The major differences between data structure for discovering classification rules and the foraging paths of real ant colonies can be summarized as below:

(1) Real ants go to a food source from their nest and back to the nest later. But artificial ants in our data structure do not return. The movements of artificial ants are transitions from discrete states to discrete states. Unlike real ants, the artificial ants are ‘‘jumping’’ from one node to another. Thus, current study only copes with categorical attributes.
(2) For real ants, pheromone updating and path going happen simultaneously. But our artificial ants update pheromone trails only after a rule is generated.
(3) Real ants may travel in group. But in our artificial world, a single ant explores a given data structure. Only after an ant reaches the end node and generates a rule, another ant can start walking.
(4) Pheromone is the only factor to affect the movements of real ants, or else they will move on the ground randomly. To improve the algorithm efficiency, a ‘‘looking-ahead’’ characteristic has been added to the artificial ants, which will be introduced in detail below.

2.3 Description and workflow of ant-miner algorithm

The ant-miner algorithm has similar basic principle with the shortest path discovery of real ants. The design of the algorithm involves a probability function which describes the possibility of ants path choosing. The function is based on: (1) the amount of pheromone in the trail; and (2) a problem dependent heuristic function .After several ants exploring, more and more ants will take a specific path because the positive feedback mechanism, that is, the path with a larger amount of pheromone and heuristic value have a greater probability of being chosen by an ant. If a path is chosen by the ant, the pheromone on it will increase.Once enough ants take this path, it will be chosen as a candidate rule. If its quality is good enough, it will definitely become a discovered rule. The feedback mechanism is the major characteristic of ant-miner algorithms. It can be considered as the major difference between ant-miner and other CRD methods.

The flowchart of the ant-miner algorithm for rule discovery is given in Fig. 2. The algorithm starts with obtaining a training set which consists of training cases. After that, the main loop will be executed to discover one rule per iteration: (1) It begins with initializing the index of ant (t), the index of converge, ConvergeIndex (j) which is used to test convergence of ants paths and pheromone on all trails. Convergence of path is an important indicator to check if a steady path chosen by ant colony has formed. It tests if ants take the same path one after another and record how many ants take this path. (2) A sub-loop is executed to discover classification rules by a number of artificial ants (No_of_ants) who explore paths in turn. And the discovery process consists of three main steps: rule generation, rule pruning and pheromone updating. The sub-loop will terminate under the condition that all ants have taken their exploration ($t \geq$No_of_ants), or the convergence state has been reached ($j \geq$No_rule_converge). No_rule_converge is the threshold of ConvergIndex (j). It means if there are No_rule_converge ants take the same path one after another, the path is qualified for a discovered rule candidate. If the current ant has constructed a rule that is exactly the same as the rule constructed by previous ants, then it is said the ants have converged to a single rule (path) and the value of ConvergeIndex (j) will increase by 1. (3) The main loop selects the best rule from the discovered rules according to their qualities. (4) The training cases which are covered by the best rule need to be removed from the training set. In other words, the number of training cases in the training set is gradually decreased with continuous matching with the best rules. The loop will end when the number of training cases is greater than the user-specified threshold that is the Max_uncovered_cases (Max_uc).Ant-miner algorithm can meet the requirements of exclusiveness and completeness: (1) It will not happen that multiple rules apply to a case. The rules apply to cases one by one in the same order with that they are discovered. The former discovered rule is used earlier. All the cases which are covered by it will be applied and these cases will not be considered when the later rules are used. This means that only the first rule which can cover the case will apply to it. Therefore, it does not have the situation that multiple rules apply to a case. (2) There will always be one rule matching the conditions of a given case. Ant-miner algorithm discovers rules according to the training cases. If a rule is discovered, all the training cases covered by this rule will be removed from the training set. After that, the algorithm will discover another rule based on the rest cases. This is a circulation process, until the number of uncovered cases is less than the parameter ‘‘Max_uncovered_cases (Max_uc)’’.The default rule has the majority class value in the set of uncovered training cases (Parpinelli et al., 2002a, 2002b).

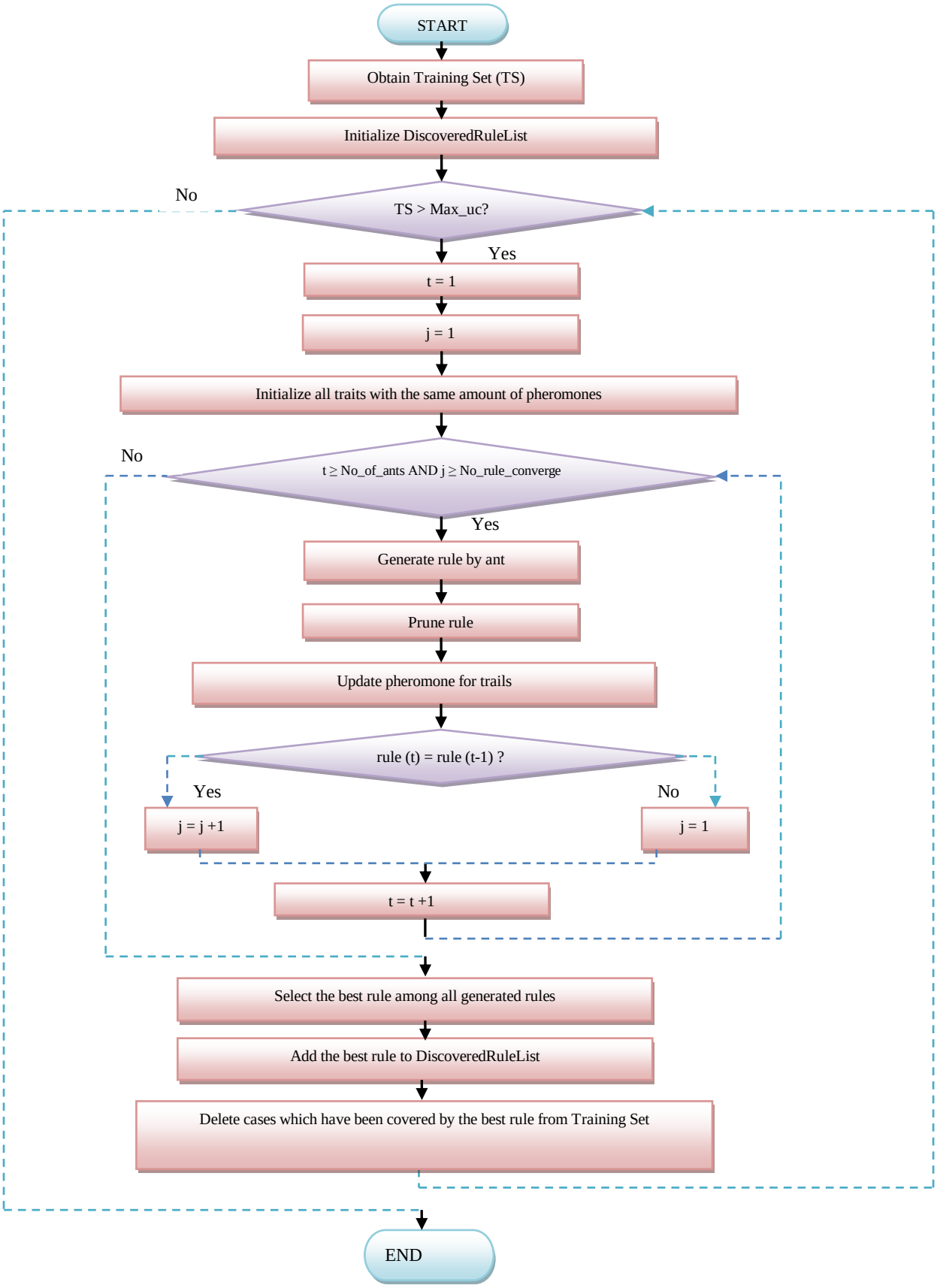

Fig. 2 Work flow of Ant-Miner Algorithm for classification rule discovery (Jia Yu et al., (2011)

2.3.1 Rule generation

The ants start from the artificial nest and choose a value for each attribute for rule generation. This process is carried out by the probability function (Eq. (1)). It gives the probability (P_{ij}) that V_{ij} is selected as the value of $Attribute_i$ ($Attribute_i = V_{ij}$), where Attributei is the i^{th} attribute and V_{ij} is the j^{th} value of $Attribute_i$ (adapted from Parpinelli et al. (2002a)).

$$P_{ij} = \frac{\eta_{ij} \cdot \tau_{ij}(t)}{\sum_{i=1}^{a}(x_i) \cdot \sum_{j=1}^{b}(\eta_{ij} \cdot \tau_{ij}(t))} \qquad (1)$$

where, η_{ij} is the value of a problem-dependent heuristic function for $term_{ij}$. The higher the value of the more relevant for classification the $term_{ij}$ is, and so the higher its probability of being chosen. The function that defines the problem-dependent heuristic value is based on information theory. τ_{ij} *(t)* is the amount of pheromone associated with $term_{ij}$ at iteration *t*, corresponding to the amount of pheromone currently available in the position *i, j* of the path being followed by the current ant. The better the quality of the rule constructed by an ant, the higher the amount of pheromone added to the trail segments visited by the ant. Therefore, as time goes by, the best trail segments to be followed – that is, the best terms (attribute-value pairs) to be added to a rule – will have greater and greater amounts of pheromone, increasing their probability of being chosen. a is the total number of attributes. x_i is set to 1 if the attribute a_i was not yet used by the current ant, or to 0 otherwise. b_i is the number of values in the domain of the *i*-th attribute. A $term_{ij}$ is chosen to be added to the current partial rule with probability proportional to the value of Equation (1), subject to two restrictions, namely:(1)The attribute A_i cannot be already contained in the current partial rule. In order to satisfy this restriction the ants must "remember" which terms (attribute-value pairs) are contained in the current partial rule.(2)A $term_{ij}$ cannot be added to the current partial rule if this makes it cover less than a predefined minimum number of cases, called the *Min_cases_per_rule* threshold. Once the rule antecedent is completed, the system chooses the rule consequent (i.e., the predicted class) that maximizes the quality of the rule. This is done by assigning to the rule consequent the majority class among the cases covered by the rule.

$$\eta_{ij} = \max\left(\frac{freqT^1_{ij}, freqT^2_{ij} \ldots\ldots\ldots freqT^k_{ij}}{|T_{ij}|}\right) \qquad (2)$$

where T_{ij} is the partition containing the cases where $Attribute_i = V_{ij}$; $|T_{ij}|$ is the number of cases in T_{ij}; $freqT^k_{ij}$ is the number of cases in T_{ij} where Class = C_k. The higher the value of η_{ij}, the higher the possibility of V_{ij} to be selected as a new node into the rule. Compared with the heuristic function which was originally proposed by Parpinelli et al. (2002a), this function has lower complexity. V_{ij} is selected to be added to a rule depending on the probability from Eq. (1). But there is an exception that Vij cannot be added to the current partial rule when the rule covers less than a specified minimum number of cases, called the Min_cases_per_rule (minimum number of cases covered per rule). After the ant has explored all attributes (in other words, the predecessor of the rule has been generated), it will choose C_i, which is the i^{th} value of the Class (the successor of the rule). The ant selects the C_i which accumulates mostly in the training cases covered by the predecessor of the rule. The whole rule is not generated until the ant has selected the successor.

2.3.2 Rule pruning

Rule pruning is a common place technique in data mining. Once a rule is created, the rule pruning operation is invoked. Rule pruning will increase the quality of a rule. It makes the rule simpler and easier to be understood. The rule pruning process tries to remove each node of the rule predecessor in turn, and then computes the quality of the rule. It picks the node whose removal most significantly improves the quality of the rule, and actually removes it from the rule after each node in the rule has been tried. Once a node is removed, the rule has been shortened.

2.3.3 Pheromone Updating

Pheromone updating of the ant-miner algorithm is designed to simulate the pheromone ants left that evaporated in the real world. Pheromones on nodes guide artificial ants to find the right ''paths'' (rules). Pheromone updating improves the classification accuracy of ant-miner in this study. It is because that the positive feedback effect of the pheromone updating helps to correct some mistakes made by the short-sightedness of the heuristic measure. Pheromone updating copes better with attribute interactions than entropy measure, because it is based on the performance of a rule as a whole (Parpinelli et al., 2002b). Therefore,

pheromone updating helps getting better classification rules. Eq. (3) (Parpinelli et al., 2002a) introduces the definition of rule quality.
Where,

$$Q = \frac{TP}{TP+FN} \cdot \frac{TN}{FP+TN} \qquad (3)$$

- *TP* (true positives) is the number of cases covered by the rule that have the class predicted by the rule.
- *FP* (false positives) is the number of cases covered by the rule that have a class different from the class predicted by the rule.
- *FN* (false negatives) is the number of cases that are not covered by the rule but that have the class predicted by the rule
- *TN* (true negatives) is the number of cases that are not covered by the rule and that do not have the class predicted by the rule.

where Q is the quality of a rule, $0 \leq Q \leq 1$; TruePos (true positive) is the number of training cases in the training set whose antecedent part and consequent part are covered by the rule; FalsePos (false positive) is the number of cases whose antecedent part is covered by the rule and the consequent is not covered; FalseNeg (false negative) is the number of cases whose antecedent part is not covered by the rule but the consequent part covered; TrueNeg (true negative) is the number of cases whose antecedent part and consequent part are not covered by the rule. Q in Eq. (3) decides how much pheromone will be added to the path which has been taken by the ant. The better the quality of a rule (a path taken by ants), the more pheromone will be exposed to the path so as to attract more ants to take this path. Several equations below define how to update pheromone.

(1) Pheromone initialisation
The operation is to ''Initialize all trails with the same amount of pheromone''. It is defined as Eq. (4) (Parpinelli et al., 2002b), where a is the number of attributes; b_i is the number of values of $Attribute_i$; t is the sequence number of iteration.

$$\tau_{ij}(t=0) = \frac{1}{\sum_{I=1}^{a} b_i} \qquad (4)$$

(2) Pheromone updating for explored nodes
The amount of pheromone on the nodes which have been used by the current rule will be updated because the artificial ant deposits pheromone during path exploration. Meanwhile, the pheromone evaporation also needs to be simulated. Therefore, the integrative operation is performed according to Eq. (5) (Liu et al., 2004).

$$\tau_{ij}(t) = (1-\rho)\,\tau_{ij}(t-1) + (1-1/1+Q)\,\tau_{ij}(t-1) \qquad (5)$$

where ρ is the pheromone evaporation rate which controls how fast the pheromone evaporates from the trails; Q is the quality of the rule which is calculated from Eq. (3); t is the sequence number of iteration. This equation adopted from the pheromone updating function of Ant-Miner (Parpinelli et al., 2002) has higher classification accuracy because Ant-Miner's function does not consider pheromone evaporation for explored nodes.

(3) Pheromone updating for unexplored nodes
The nodes which have not been used by the current rule will only have pheromone evaporation. The evaporation is performed according to Eq. (6):

$$\tau_{ij}(t) = \frac{\tau_{ij}(t-1)}{\sum_{i=1}^{a} \sum_{j=1}^{bi} b_i=1\ \tau_{ij}(t-1)} \qquad (6)$$

where a is the number of attributes; b_i is the number of values of $Attribute_i$; t is the sequence number of iteration. The equation means that the amount of pheromone of unexplored nodes will be decreased as time goes by.

III. IMPLEMENTATION OF CLASSIFICATION RULE DISCOVERY FOR NETWORK INTRUSION DETECTION

There are mainly three reasons for us to use ant-miner algorithm for bankruptcy prediction purpose. (1)The ant-miner algorithm can achieve CRD problems with good performances. (2) Bankruptcy prediction is a classification based problem, ant-miner algorithm yields better results compare to other algorithm. (3) CRD can help discovering knowledge of bankruptcy prediction classification in large amount of data. ACO, a sub-field of swarm intelligence (Blum & Dorigo, 2004; Dorigo, Di Caro, & Gambardella, 1999). It is one of the most advanced techniques for approximate optimization (Blum, 2005). In this paper, we proposed the solution for network intrusion detection using ant-miner algorithm. The results of the experiment show that the ant-miner method has significantly better performance than other classifiers in terms of rules generation and predictive accuracy.

3.1 Data and experiment design

In order to make a reliable comparison we used three benchmark datasets including network intrusion detection dataset. In this research, Network Intrusion Detection dataset is collected by our self from the experts and we donated this to UCI repository. Table 1 shows the variable name, instances.

TABLE 1
Description of datasets

Dataset	Feature	Instances	Normal/Anomaly
Network Intrusion Detection	41	783	408/375

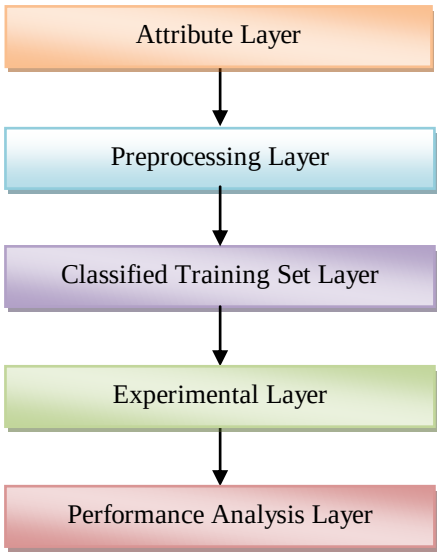

Fig. 3 Framework of Bankruptcy Prediction System

3.2 Results

The ant- miner finally extracts 43 rules, 22 of which are anomaly and the others are normal. The simple cart finally extracts 3 rules out of which 2 are anomaly and others are normal. The rules and the corresponding descriptions are illustrated in table 1and 2 for ant- miner and AD tree. The AD tree extracts 10 rules out of which 3 are anomaly and others are normal. J48 finally extracts 4 rules out of which 3 are normal and other is anomaly. BF tree finally extracts 4 rules out of which 2 normal and others are anomaly. The rules and descriptions of BF tree are illustrated in table 5. Overall classification means the accuracy level when the rules are applied to the cases according to the application steps generated from 5 data mining techniques. The results show that the rules of the ant-miner methods are significantly better than those of other data mining techniques. While the rules generated from the AD tree, BF tree, J48. The below results shows that the ant-miner is best than the other data sets.

TABLE 1
The descriptions of the rules generated from Ant-miner

Rule	Description
Rule 1	IF logged_in=0 AND src_bytes<=28.5 THEN anomaly
Rule 2	IF dst_host_same_src_port_rate<=0.995 AND src_bytes<=1009.0 THEN normal
Rule 3	IF dst_bytes>0.5 AND hot<=1.5 THEN normal
Rule 4	IF dst_host_same_srv_rate>0.395 AND src_bytes>290.0 THEN anomaly
Rule 5	IF dst_host_srv_diff_host_rate<=0.035AND hot<=25.0 AND count<=29.5THENnormal
Rule 6	IF src_bytes>28.5 AND dst_host_same_src_port_rate<=0.995 THEN normal
Rule 7	IF dst_host_srv_count<=84.0 AND count>1.5 THEN anomaly
Rule 8	IFdst_host_same_src_port_rate>0.635ANDdst_host_rerror_rate<=0.845AND dst_host_srv_diff_host_rate>0.095THEN anomaly
Rule 9	IF count<=4.5 AND dst_host_srv_count>2.0 THEN normal
Rule 10	IF dst_host_count>23.0 AND dst_bytes<=2.0 THEN anomaly
Rule 11	IF dst_host_srv_count<=82.5 AND count>1.5 THEN anomaly
Rule 12	IF protocol_type=tcp AND dst_host_diff_srv_rate<=0.125 AND src_bytes <=285.5 THEN Normal
Rule 13	IF dst_host_srv_diff_host_rate>0.09 THEN Anomaly
Rule14	IF dst_host_count>251.0 AND dst_bytes<=2 THEN anomaly
Rule 15	IF flag=SF THEN normal
Rule 16	IF src_bytes> 28.5 AND flag = SF AND hot<= 1.5 THEN normal
Rule 17	IF dst_bytes<= 2.0 AND dst_host_count > 223.5 THEN anomaly
Rule 18	IF srv_count > 2.5 THEN anomaly
Rule 19	IF dst_host_same_srv_rate > 0.025 AND src_byte<= 17.5 AND dst_host_diff_srv_rate <= 0.6899 THEN normal
Rule 20	IF dst_host_same_src_port_rate> 0.03 THEN anomaly
Rule 21	IF count <= 1.5 AND flag = SF THEN normal
Rule 22	IF src_bytes > 28.5 AND hot <= 0.5 THEN normal
Rule 23	IF srv_count > 3.5 THEN anomaly
Rule 24	IF dst_bytes <= 1044.0 AND protocol_ type = tcp AND dst_host_srv_count > 4.5 THEN normal
Rule 25	IF dst_host_rerror_rate > 0.005 THEN anomaly
Rule 26	IF dst_host_same_src_port_rate <= 0.995 AND src_bytes <= 1009.0 AND is_guest_login = 0 THEN normal
Rule 27	IF count <= 1.5 AND dst_host_rerror_rate <= 0.005 AND dst_host_srv_diff_host_rate <= 0.095 THEN Normal
Rule 28	IF duration <= 2.5 AND dst_host_diff_srv_rate < 0. 025 AND dst_host_rerror_rate <= 0.375 THEN anomaly
Rule 29	IF land = 0 THEN normal
Rule 30	IF src_bytes > 28.5 AND hot <= 1.5 THEN normal
Rule 31	IF dst_bytes <= 2.0 AND dst_host_count > 223.5 THEN Anomaly
Rule 32	IF srv_count > 3.5 THEN anomaly
Rule 33	IF src_bytes <= 17.5 AND dst_host_diff_srv_rate <= 0.06 THEN normal
Rule 34	IFdst_host_rerror_rate>0.005AND dst_host_same_src_port_rate <= 0.3 THEN anomaly
Rule 35	IF protocol _type = tcp AND flag = SF THEN normal
Rule 36	IF dst_host_same_src_port_rate > 0.5649 AND dst_host_rerror_rate <= 0.845 AND dst_

	bytes <= 0.5THEN anomaly
Rule 37	IF dst_bytes <= 0.5 THEN anomaly
Rule 38	IF src_bytes <= 24.0 AND dst_host_same_srv_rate <= 0.34 THEN anomaly
Rule 39	IF src_bytes <= 399.0 AND dst_host_srv_serror_rate <= 0.135 THEN normal
Rule 40	IF flag = SF AND dst_host_same_src_port_rate <= 0.995 THEN normal
Rule 41	IF dst_host_srv_count <= 82.5 AND src_bytes <= 13.0 AND dst_host_diff_srv_rate > 0.025 THEN anomaly
Rule 42	IF src_bytes > 4.0 AND dst_bytes <= 0.5 THEN anomaly
Rule 43	IF dst_host_srv_serror_rate<=0.035 THEN normal

TABLE 2
The descriptions of the rules generated from ADTree

Rule	Description
Rule 1	IF dst_bytes<0.5:0.938 AND dst_bytes>=0.5:-1.688
Rule 2	IF count>20.5:-0.771 AND count>=20.5:2.391
Rule 3	IF hot>0.5:-2.06 AND hot>0.5:1.663
Rule 4	IF src_bytes<28.5:0.511 AND src_bytes>=28.5:-0.744
Rule 5	IFdst_host_same_src_port_rate<0.985:-0.629AND dst_host_same_src_port_rate>=0.985:0.866
Rule 6	IF service=http:-1.185 AND service!=http:0.83
Rule 7	IF dst_host_srv_count<42.5:0.371 AND dst_host_srv_count>=42.5:-0.889
Rule 8	IF service=ftp_data:-1.006 AND service!=ftp_data: 0.987
Rule 9	IF dst_host_serror_rate<0.02:-0.949 AND dst_host_serror_rate>=0.02:0.851
Rule 10	IF dst_host_rerror_rate<0.005:-0.745 AND dst_host_rerror_rate<=0.005:0.958

TABLE 3
The descriptions of the rules generated from Simple cart

Rule	Description
Rule 1	IF service=(ecr_i)|(ftp)|(ftp_data) AND service!=(ecr_i)|(ftp)|(ftp_data)
Rule 2	IFdst_host_serror_rate<0.045:THEN normalAND dst_host_serror_rate>=0.045:THEN anomalySrc_bytes>=28.5ANDdst_host_same_srv_rate<0.455ANDdst_host_same_srv_rate >=0.45:THENanomalyANDsrc_byte<<34167.0:THENnormalANDsrc_bytes>=34167.0: THEN anomaly
Rule 3	IF protocol_type=(icmp) THEN anomaly AND if protocol_type=(icmp) THEN normal

TABLE 4
The descriptions of the rules generated from J48

Rule	Description
Rule 1	IFdst_host_count<=235ANDdst_host_count>235ANDdst_host_same_src_port_rate<=0.99AND dst_host_same_src_port_rate > 0.99
Rule 2	IF srv_count <= 3 AND srv_count >3: THEN anomaly AND dst_host_same_srv_rate <=0.2: THEN anomaly AND dst_host_same_srv_rate > 0.2 AND dst_host_rerror_rate<= 0 AND dst_host_rerror_rate > 0 AND src_bytes <= 241: THEN normal AND src_bytes > 241
Rule 3	IF protocol_ type = tcp AND protocol_type = udp : THEN normal AND protocol_ type = icmp : THEN anomaly AND dst_bytes <= 4 :THEN anomaly AND dst_bytes > 4 : THEN normal AND protocol_type = tcp : THEN normal AND protocol_type = udp : THEN normal AND protocol_type =icmp AND hot <= 0 : THEN normal AND hot >0 : THEN anomaly AND dst_bytes <= 1 : THEN anomaly AND dst_bytes >1 : THEN normal
Rule 4	IF dst_host_srv_count <= 2 : THEN anomaly AND dst_host_srv_count >2 : THEN normal AND src_bytes <= 570 THEN normal AND src_bytes > 570 : THEN anomaly

TABLE 5
The descriptions of the rules generated from BFTree

Rule	Description
Rule 1	IF service = (ecr_i)|(ftp)|(ftp_data) AND service != (ecr_i)|(ftp)|(ftp_data)
Rule 2	IFdst_host_serror_rate<0.045:THENnormalANDdst_host_serror_rate>=0.045:THENanomaly src_bytes >= 28.5 AND dst_host_same_srv_rate < 0.455ANDdst_host_same_srv_rate >= 0.455 : THEN anomaly AND src_bytes < 34167.0 : THEN normal AND src_bytes >= 34167.0 : THEN anomaly
Rule 3	IF protocol_type = (icmp): THEN anomaly AND protocol_type!=(icmp)
Rule4	IF hot < 26.0 : THEN normal AND hot >=26.0 THEN normal

TABLE 6
Comparison of Ant-miner with ID3, Genetic algorithm, neural networks, and Inductive learning methods

DATA-MINING CLASSIFICATION ALGORITHM	CONFUSION MATRIX		NUMBER OF RULES	ACCURACY
ANT- MINER	**410**	**1**	**43**	**99.61**
	2	**370**		
ADTREE	**394**	**14**	**10**	**97.19**
	8	**367**		
SIMPLE CART	**396**	**12**	**3**	**97.57**
	7	**368**		
J48	**398**	**10**	**4**	**97.06**
	13	**362**		
BFTREE	**396**	**12**	**4**	**97.31**
	9	**366**		

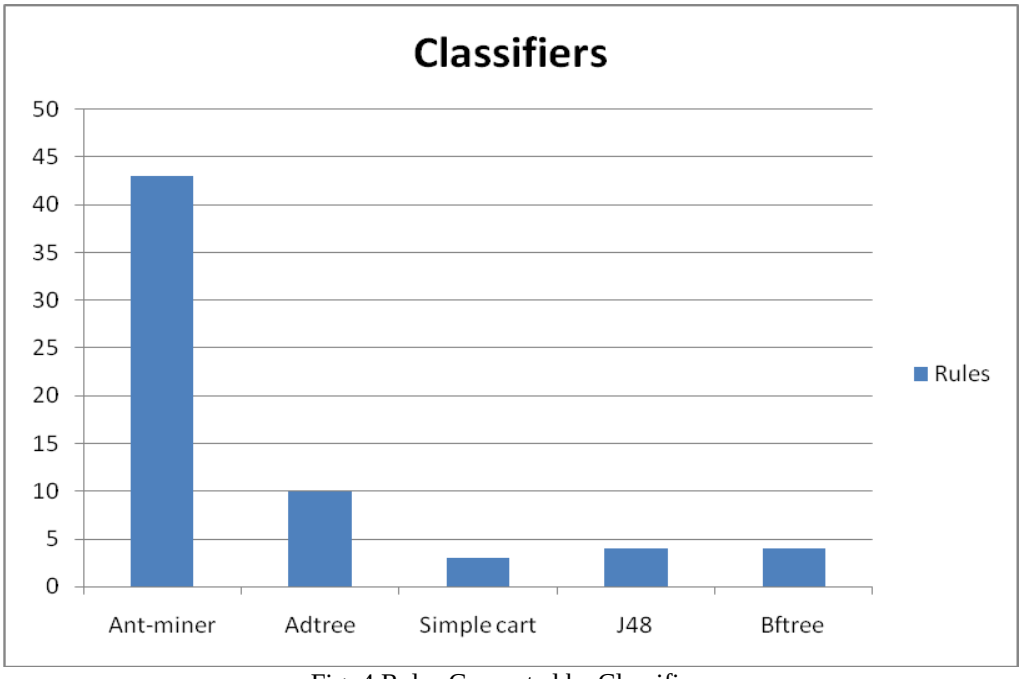

Fig. 4 Rules Generated by Classifiers

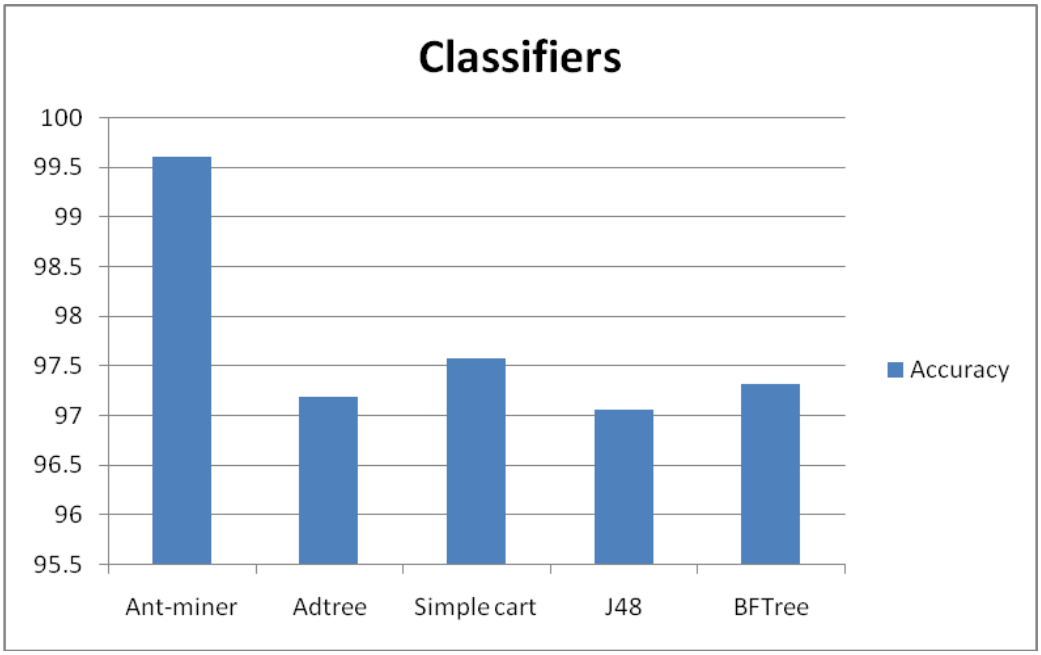

Fig. 5 Accuracy by Classifiers

This paper demonstrated the ant-miner based data mining approach to discover decision rules from experts' decision process. This study is the first work on ant-miner for the purpose of discovering experts' qualitative knowledge on bankruptcy. Four data mining techniques ID3, GA, Neural networks, Inductive learning methods are applied to compare their performance with ant-miner method.

IV. CONCLUSIONS

Data mining has been widely applied to discovering Network Intrusion Detection databases. However, few studies have reported the potential of data mining that can investigate the Network Intrusion Detection from experts' decisions. This work proposes a new method of classification rule discovery for Network Intrusion Detection by using ant-miner algorithm. This paper demonstrates ant-miner based data mining approach to discover decision rules from experts' decision process in networking way to predict Intrusion Detection. In performance terms, ant-miner provides more rules but give better predictive accuracy when compare to other techniques. This study has conducted a case study using the dataset Network Intrusion Detection dataset is retrieved from UCI repository in July 2014. Finally this paper suggests that Ant-miner could be a more suitable method than the other classifiers like J48, Simple Cart, ADTree, and BFTree. In future research, additional artificial techniques could also be applied. And certainly researchers could expand the system with more dataset.

REFERENCES

[1] Han, J., & Kamber, M. Data mining: Concepts and techniques. San Francisco, CA, USA: Morgan Kaufmann. (2001).
[2] Brachman, R. J., Khabaza, T., Kloesgen, W., Piatesky-Shpiro, G., & Simoudis, E. Mining business databases. Communication of the ACM, 39(11) (1996) 42-48.
[3] Hayes-Roth, F. Rule-based systems. Communications of the ACM, 28 (1985) 921-932.
[4] Alupoaei, S., & Katkoori, S. Ant colony system application to marcocell overlap removal. IEEE Transactions Very Large Scale Integration (VLSI) Systems, 12(10) (2004) 1118-1122.
[5] Bianchi, L., Gambardella, L.M., & Dorigo, M. An ant colony optimization approach to the probabilistic travelling salesman problem. In J. J. Merelo, P. Adamidis, H. G.Beyer, J. L. Fernandez-Villacanas, and H. P. Schwefel (Eds.), Proceedings of PPSN-VII, seventh international conference on parallel problem solving from nature. Lecture Notes in Comput Science (2002) 883-892. Berlin: Springer.
[6] Demirel, N. C., & Toksari, M. D. Optimization of the quadratic assignment problem using an ant colony algorithm. Applied Mathematics and Computation, 183(1), (2006) 427-435.
[7] Lina, B. M. T. Lub, C. Y., Shyuc, S. J., & Tsaic, C. Y. Development of new features of ant colony optimization for flowshop scheduling. International Journal of Productions Economics, 112(2),(2008)742-755.
[8] Ramos, G. N., Hatakeyama, Y., Dong, F., & Hirota, K. Hyperbox clustering with ant colony optimization (HACO) method and its application to medical risk profile recognition. Appied Soft Computing, 9(2), (2009) 632-640.

[9] Parpinelli, R. S., Lopes, H. S., & Freitas , A. A. An ant colony algorithm for classification rule discovery. In H. A. Abbass, R. A. Sarker, & C. S. Newton (Eds.), Data mining: A heuristic approach (2002a) 191-208. London: Idea Group Publishing.

[10] Parpinelli, R. S., Lopes,H.S.,&Freitas,A.A.An ant colony algorithm for classification rule discovery.In H.A.Abbass,R.A.Sarker,&C.S.Newton(Eds.),Data mining:A heuristic approach (2002a) 191-208.London Idea Group Publishing.

[11] Parpinelli, R.S., Lopes, H.S.,& Freitas, A.A. Data mining with an ant mining optimization Algorithm IEEE Transaction on Evolutionary Computation,6(4),(2002b) 321-332.

[12] Jiang, W., Xu, Y., & Xu, Y. A novel data method based on ant colony algorithm. Lecture Notes in Computer Science, (2005) 3584, 284-291.

[13] Jin, P., Zhu, Y., Hu, K., & Li, S. classification rule mining based on ant colony optimization algorithm. In D. S. Huang, K. Li, & G. W. Irwin (Eds.), Intelligent control and automation, lecture notes in control and information and sciences (2006) 654-663. Berlin: Springer.

[14] Liu, B., Abbass, H, & Mckay, B. Classification rule discovery with ant colony optimization. IEEE Computational Intelligence Bulletin, 3(1),(2004) 31-35.

[15] Roozmand, O., & Zamanifar, K. Parallel ant miner 2. In L. Rutkowski et al.(Eds.), Artifical Intelligence and soft computing – ICAISC (2008) 681-692 Berlin: springer.

[16] Liu, B., Abbass, H.A., & Mckay, B. Density-based heuristic for rule discovery for Ant-Miner .In: The 6th Australia-Japan joint workshop on intelligent and evolutionary system (2002) 180-184 Canberra, Australia.

[17] Wang, Z. & Feng, B. Classification rule mining with an improved ant colony algorithm. In G. I. Webb & X. Yu (Eds.), AI 2004: Advances in artificial intelligence, lecture notes in computer science (2005) 357-367 Berlin: Springer.

[18] Blum, C., & Dorigo, M. Deception in ant colony optimization. In M. Dorigo, M. Birattari, C. Blum, L. M. Gambardella, F. Mondada, & T. Stutzle (Eds.), Proc. ANTS 2004, Fourth Internet Workshop on Ant Colony Optimization and Swarm Inteligence, Lecture Notes in Computer Science (2004). Berlin:Springer.

[19] Nejad, N. Z., Bakhtiary, A. H., & Analoui, M. Classification using unstructured rules and ant colony optimization. In: Proceedings of the International multi conference of engineers and computer scientists Vol. I, (2008) 506-510. Hong Kong.

[20] Dorigo, M., Di Caro, G., & Gambardella, L. M. Ant algorithms for discrete optimization. Artificial Life, 5(2), (1999) 137-172.

[21] Blum,C.Ant colony optimization: introductionnand recent trends.Physics of Life Reviews,2(2005)353-373.

[22] Merkle, D. Middebdorf, M., & Schmerck, H. Ant colony optimization for resource-constrained project scheduling ,IEEE Transcations on Evolutionary Computation, 6(4) (2002) 333-346.

[23] Fuellerer, G., Doerner, K. F., Hartl, R. F., & Iori, M .Ant Colony Optimization for two-dimensional loading vehicle routing problem. Computers and Operations Research, 36(3), (2009) 655-673.

[24] Reimann, M., Doerner, K., & Hartl, R.F. D-ants: Savings based ants divide and conquer the vehicle routing problems. Computers & Operations Research, 31(4), (2004) 563-591.

[25] Colorni, A., Dorigo, M., Maniezzo, V., & Trubian, M. An system for job-shop scheduling. Belgian Journal of Operations Research. Statistics and Computer Science (JORBEL), 34, (1994). 39-52.

[26] Stutzle, T. MAX-MIN Ant system for the quadratic assignment problem. Technical report AIDA-97-4, FG Intellektik, FB Innformatik, TU Darmstadt, Germany (1997b).

[27] Leguizamo n, g., & Michalewicz, Z., A new version of Ant System for subset problems. In P. J. Angeline, Z. Michalewicz, M. Schoenauer, X. Yao, & A. Zalzala (Eds,), Proceedings of the 1999 Congresson Evolutionary Computation (CEC'99) (pp. 1459-1464). Piscataway, NJ, IEEE Press.

[28] De Campos, L. M., Ga'mez, J. A., & Puerta, J. M. Learning Bayesian networks by ant colony optimization: Searching in the space of orderings. Mathware and Soft computing, 9(2-3) (2002b).251-268.

[29] Corry, P., & Kozan, E.. Ant Colony Optimization for machine layout problems. Computational Optimization and Applications,28(3),(2004) 287-310.

[30] Jia Yu, Yun Chen, Jianping Wu, Modeling and implementations of classification rule discovery by ant colony optimization for spatial land-use suitability assessment, Computers, Environment and Urban Systems 35 (2011) 308-319.

[31] W. Lee, S.J. Stolfo, K.W. Mok, A data mining framework for building intrusion detection models, in: Proceedings of IEEE Symposium on Security and Privacy, 1999, pp, 120-132.

[32] W.Lee, S.J Stolfo, K. W. Mok, Mining audit data to build intrusion detection models, in: Proceedings of the 4th International conference on Knowledge Discovery and Data Mining, AAAI Press, 1998,pp, 66-72.

[33] L. Khan, M. Awad, B. Thuraisingham, A new intrusion detection systems using support vector machines and hierarchial clustering. The VLDB Journal 16(2007) 507-521.

[34] X. Xu, Adaptive intrusion detection based on machine learning: feature extraction, classifier construction and sequential pattern prediction, Information Assurance and Security 4(2006) 237-246.

[35] Y. Feng, J. Zhong, C. Ye, Z. Xiong, Z. Wu Intrusion detection classifier base on self-organizing ant colony networks clustering, Information Assurance Security 4(2006) 247-256.

[36] C.-F. Tsai, Y. F. Hsu, C.-Y. Lin, W.-Y. Lin, Intrusion detection by machine learning: a review, Expert Systems with Applications 36(2009) 11994-12000.

[37] S. Kumar, Classification and detection of computer intrusions. Ph.D. Thesis, Purdue University, August 1995.
[38] B. Mukherjee, L. T. Heberlein, K. N. Levitt, Network intrusion detection, IEEE, Network (1994) 26-41.
[39] E. H. Spafford D. Zamboni, Intrusion detection using autonomous agents, Computer Networks 34(4) (2000) 547-570.
[40] F. Klawonn, F. Hoppner What is fuzzy about fuzzy clustering? Understanding and improving the concept of the fuzzifier, in: Cryptographic Hardware Embedded Systems. Vol. 2779, Springer, 2003, pp.254-264.
[41] J. Han, M. Kamber, J. Pei, Data Mining Concepts Models Methods and Algorithms John Wiley, 2003.
[42] UCI KDD Archive, KDD Cup 1999 data, 1999.http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.