Neurocomputing 111 (2013) 161–168

Contents lists available at SciVerse ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Designing of on line intrusion detection system using rough set theory and Q-learning algorithm

Nandita Sengupta [a,*], Jaydeep Sen [b], Jaya Sil [c], Moumita Saha [d]

[a] Information Technology Program, University College of Bahrain, P.O. Box 55040, Manama, Kingdom of Bahrain
[b] Department of Computer Science and Engineering, Indian Institute of Technology, Kanpur (IITK), India
[c] Department of Computer Science and Technology, Bengal Engineering and Science University Shibpur, P.O. Botanic Garden, Howrah, Pin 711103, West Bengal, India
[d] Department of Computer Science and Engineering, Indian Institute of Technology, Kharagpur, India

ARTICLE INFO

Article history:
Received 12 February 2012
Received in revised form
22 September 2012
Accepted 18 December 2012
Communicated by A. Konar
Available online 23 January 2013

Keywords:
Q-learning algorithm
Rough set theory
Discretization
Intrusion detection system

ABSTRACT

Development of an efficient real time intrusion detection system (IDS) has been proposed in the paper by integrating Q-learning algorithm and rough set theory (RST). The objective of the work is to achieve maximum classification accuracy while detecting intrusions by classifying NSL-KDD network traffic data either '*normal*' or '*anomaly*'. Since RST processes discrete data only, by applying cut operation attributes in training data are discretized. Using indiscernibility concept of RST, reduced attribute sets, called *reducts* are obtained and among the *reducts* a single *reduct* is chosen which provides highest classification accuracy. However, for the test data the same *reduct* would not provide highest classification accuracy due to change of discretized attribute values. Therefore, to overcome the problem discretization and feature selection processes are dealt in a comprehensive and systematic way in the paper using machine learning approach. The Q-learning algorithm has been modified to learn optimum cut value for different attributes so that corresponding *reduct* produces maximum classification accuracy while classifying network traffic data. Since, not all attributes but *reduct* only take part to detect intrusions, the proposed algorithm is faster than Q-learning and reduces complexity of the IDS. Classification accuracy with 98% success rate has been obtained using real time data, which demonstrates superior performance compared to other classifiers.

1. Introduction

Information exchange through computer network increases exponentially and so the potential threat to the global information infrastructure, which need to be protected from different kind of attacks. The attackers attempt to destroy confidentiality, integrity or availability of computer network or systems are cause of intrusions. Intrusion detection is one of the core activities in computer systems that classify network traffic data either '*normal*' or '*anomaly*'. However, large dimensional data set often consisting of redundant information, which dominate relevant information and affects classification accuracy negatively. In case of on line classification of data, the problem becomes more complex in order to accommodate test data dynamically.

Existing intrusion detection systems (IDS) lack systematic approach to construct a classifier that can efficiently perform misuse and anomaly detection. Most of the works either use large number of features to evaluate intrusive patterns or apply lengthy learning scheme to classify network traffic data. Supervised and unsupervised learning [1–8] have been widely used in intrusion detection system (IDS) as classification technique. Support vector machines, decision trees, *k*-nearest neighbor, artificial neural networks and clustering techniques, name a few machine learning approaches are used in developing IDS. In case of supervised learning, labeling of training data is time consuming while in unsupervised learning proper partitioning of data becomes difficult due to the absence of sufficient a priori domain knowledge. Moreover, none of the techniques able to accomodate data dynamically for on line classification.

Reinforcement learning is applied for sequential prediction and used to build both network based and host based IDS [9]. In [10], it is mentioned that for applying reinforcement learning sufficient amount of sample data is required to learn the environment. However, large sample size makes the learning procedure slow and therefore not suitable for on line applications. Q-learning, one kind of reinforcement learning technique [11–17] executes an action by comparing all available actions in a particular state without knowing the whole environment [18]. However, a single Q-learning module faces difficulty to solve a

* Corresponding author. Tel.: +973 39608665; fax: +973 1779 3828.
E-mail addresses: nanditasengupta@yahoo.com, ngupta@ucb.edu.bh (N. Sengupta), jdpsen@cse.iitk.ac.in (J. Sen).

0925-2312/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.neucom.2012.12.023

large problem. In that case, modular Q-learning is preferred to reduce system complexity that subdivide the problem manually into different modules and Q-learning is applied on each such module [19]. Distributed reinforcement learning is proposed [20,21] which works in hierarchical manner where a centralized agent is used for communication. Huge computational cost is involved to form the modules and centralized communication between the agents fail to develop a robust system.

In the paper, a comprehensive approach has been proposed in developing an IDS where RST and Q-learning algorithm are integrated to accomodate real time traffic data for detecting intrusions with highest classification accuracy. Rough set theory is applied on discrete data only and so in the work cut is applied on conditional attributes for discretization. Indiscernibility concept of RST is applied on discrete data for selecting set of most significant attributes, called *reduct* sufficient to represent the original data set. However, *reduct* is not unique and so the *reduct* which provides highest classification accuracy is chosen to build the rule base classifier to classify the network traffic data either *normal* or *anomaly*. For the test data, applying the same cut value may generate different *reduct*, resulting fall of classification accuracy. Therefore, discretization and feature selection are not to be treated as independent events to classify network data accurately. In the proposed method, Q-learning algorithm has been modified to learn different cut value for each conditional attribute and corresponding *reduct* and accuracy are evaluated to form the reward matrix. Modified Q matrix evaluates optimum cut values for each attribute to achieve highest classification accuracy in detecting intrusions using network traffic data. The procedure is terminated when two successive cut generates same accuracy or monotonically decreasing accuracy.

The paper has been organized into six sections. Section 2 describes the concepts required to design the proposed IDS and the modified Q-learning algorithm is presented in Section 3. Experimental results using synthetic data set is listed in Section 4 while performance of the proposed method is evaluated in Section 5 with the help of NSL-KDD data set. Finally, conclusions and future research are summarized in Section 6.

2. Background of work

2.1. Intrusion detection system

Intrusion detection system (IDS) [1–5] protects the network from internal as well as external attacks for providing security to the network. This system differentiates the normal and the attacked traffic of the network. The basic purpose of the system is to identify the suspicious/intrusive behavior and inform the network administrator about the same by raising a flag or alarm so that the network administrator executes the appropriate action against the intruders.

In case the network size is small and database for network behavior is maintained up-to-date, intrusion detection by human analyst is to some extent feasible. But considering the growth of internet and use of computers in the network, an automatic, reliable, high speed on line intrusion detection system is becoming essential for predicting the data as '*normal*' or '*anamoly*'.

Existing works on intrusion detection [9,22,23] have been carried out to classify the network traffic as '*anomaly*' or '*normal*'. A majority of current IDS follow signature based approach, similar to virus scanners. Events are detected and matched with specific predefined patterns known as '*signatures*'. The signature based IDS fails to identify novel attacks and even minor variation of patterns are not detected accurately. In addition, sometimes IDS generate false alarm for alerting network administrator due to failure of handling imprecise data, which has high possibility to appear in network traffic. Therefore, accuracy, computation time and adaptability of the system are the key issues to be addressed properly for classifying such data in order to protect the network from intruders.

2.2. Reduct

Information system is a data table consisting of objects and attributes with values. Formally, the table is represented as a set of tuple (U,A,V_a) where U is the non-empty set of objects, A is a non-empty set of attributes and V_a is the value of attribute a such that $a: U \rightarrow V_a$. The set of attributes A is divided into two subsets, namely conditional set of attributes, C and decision set of attributes, D. Conditional set of attributes represent features of objects while decision set of attributes represent the class label of the objects. In order to eliminate the redundant and insignificant attributes from the table, concept of *reduct* is emerged in *RST*. Reduct is a minimum subset of conditional attributes sufficient to representing the whole data table. Reduct is not unique and so finding all reducts is NP hard problem. Data mining research community is engaged in developing new algorithms [24–27] for finding approximate reducts. Skrowron [28] has introduced the concept of discernibility matrix for computation of *reduct*. In the paper, discernibility matrix based reducts are generated by deriving discernibility function. As a next step, absorption and expansion laws are applied on discernibility function to remove redundant attributes while generating reducts.

2.2.1. Discernibility function

The discernibility matrix is defined as follows:

Given a decision system $DS=(U, A, C, D)$ where U is the universe of discourse and A is the total number of attributes. The system consists of two types of attribute namely conditional attributes (C) and decision attributes (D) so that $A = C \cup D$. Let the universe $U = x_1, x_2, \ldots, x_n$, then the *discernibility matrix* $M = (m_{ij})$ is a $|U| \times |U|$ matrix, in which the element m_{ij} for an object pair (x_i, x_j) is obtained as

$$m_{ij} = \{a \in C : a(x_i) \neq a(x_j) \bigwedge (d \in D, d(x_i) \neq d(x_j))\} i,j = 1,2,3,\ldots,n, \quad (1)$$

where m_{ij} is the set of attributes classifies objects x_i and x_j into different decision class labels using partition U/D. The physical meaning of the matrix element m_{ij} is the objects x_i and x_j can be distinguished by any attribute in m_{ij}. The pair (x_i, x_j) can be discerned if $m_{ij} \neq 0$. A *discernibility matrix* M is symmetric, i.e., $m_{ij} = m_{ji}$, and $m_{ii} = \emptyset$. Therefore, it is sufficient to consider only the lower triangular or the upper triangular of the matrix.

Consider an information system shown in Table 1 consisting of 10 objects, five conditional attributes, namely a,b,c,d,e and one decision attribute, say f.

Discernibility matrix of the information system (Table 1) is shown in Table 2 and corresponding discernibility function $f(s)$ has been derived

$$\begin{aligned} f(s) = {} & (a \vee b \vee e) \wedge (a \vee b \vee e) \wedge (a \vee b \vee c \vee d) \\ & \wedge (a \vee b \vee c \vee d) \wedge (a \vee c \vee d \vee e) \wedge (a \vee b \vee d) \wedge (a \vee b \vee d) \\ & \wedge (a \vee b \vee c) \wedge (a \vee c \vee d \vee e) \wedge (e) \wedge (a \vee b \vee c \vee e) \wedge (a \vee b \vee e) \\ & \wedge (a \vee b \vee e) \wedge (a \vee c \vee d \vee e) \wedge (a \vee c \vee d \vee e) \wedge (a \vee b \vee c \vee d) \\ & \wedge (a \vee b \vee c \vee d) \wedge (a \vee b \vee c \vee d \vee e) \wedge (a \vee c) \wedge (b \vee c \vee e) \\ & \wedge (a \vee b \vee c \vee d \vee e) \wedge (b \vee d \vee e) \wedge (a \vee b \vee c \vee e)) \wedge (e) \wedge (b \vee d \vee e) \\ & \wedge (a \vee c \vee e) \wedge (a \vee b \vee e) \wedge (a \vee b \vee e) \wedge (a \vee c \vee d \vee e) \\ & \wedge (a \vee c \vee d \vee e) \wedge (a \vee b \vee c \vee d \vee e) \wedge (b \vee d \vee e). \end{aligned}$$

Equivalent terms (elements connected by logical "OR" operation) are removed and discernibility function becomes

$$\begin{aligned} f(s) = {} & (a \vee b \vee e) \wedge (a \vee b \vee c \vee d) \wedge (a \vee c \vee d \vee e) \\ & \wedge (a \vee b \vee d) \wedge (a \vee b \vee c) \wedge (a \vee b \vee c \vee d \vee e) \\ & \wedge (e) \wedge (a \vee b \vee c \vee e) \wedge (a \vee c) \wedge (b \vee c \vee e) \\ & \wedge (b \vee d \vee e) \wedge (a \vee c \vee e). \end{aligned}$$

For generating reducts, absorption law is applied first. The absorption law specifies that if one term is a pure subset of another term and connected with boolean "AND" operation then the term with minimum number of variables is sustained. By applying the absorption law, discernibility function is derived as

$$f(s)=(e)\wedge(a\vee c)\wedge(a\vee b\vee d).$$

Expansion law algorithm is applied to retain the conditional attributes which are more frequently appearing in partitions compared to other attributes. *AND* operation is applied on such attributes having highest frequency and selected because they play important role in classification, compared to others which appear less frequently. *OR* operation is applied on less frequently conditional attributes to select any of them because considering all will not improve classification accuracy significantly but increase computational complexity. Finally, *AND* operation is applied on each of the *OR* term so that any of them may belong to different *reducts*.

Expansion law algorithm is described below:

(i) Find the attributes appearing most frequently (at least twice).
(ii) Apply "AND" operation on the terms having such attributes and "OR" operation on the rest.
(iii) Apply the connective "AND" between the "OR" terms and the term if consisting of such attribute then eliminate.
(iv) Combine the terms, obtained from (ii) and (iii) using "AND" operation.

In this case, most frequent attribute is a, based on which we derive

From (ii): $(a\wedge e)$.
From (iii): $c\wedge(b\vee d)$.

Table 1
An information system.

Objects	Attributes (A)					
	Condition attributes					Decision attribute (f)
	a	b	c	d	e	
O1	1	2	0	1	1	CLASS1
O2	1	2	0	1	1	CLASS1
O3	2	0	0	1	0	CLASS2
O4	0	0	1	2	1	CLASS3
O5	2	1	0	2	1	CLASS2
O6	0	0	1	2	2	CLASS1
O7	2	0	0	1	0	CLASS2
O8	0	1	2	2	1	CLASS3
O9	2	1	0	2	2	CLASS1
O10	2	0	0	1	0	CLASS2

From (iv): $(a\wedge e)\wedge(c\wedge(b\vee d))$.
So

$$\begin{aligned}f(s)&=(a\wedge e)\wedge(c\wedge(b\vee d))\\&=(a\wedge e)\wedge((c\wedge b)\vee(c\wedge d))\\&=(a\wedge e\wedge c\wedge b)\vee(a\wedge e\wedge c\wedge d).\end{aligned}$$

Therefore, the reducts are $\{a,e,c,b\}$ and $\{a,e,c,d\}$.

2.3. Q-learning

Q-learning [15] is a widely used reinforcement learning technique, suitable for on line applications. The integral part of *Q matrix* is *reward matrix*, have *states* mapped as rows and *actions* as columns. The learning algorithm executes best possible *action* in a particular *state* to reach to the goal state, assigned by the agent(s). Basically, the training algorithm learns the environment by trial and error method to reach to the goal state.

There are three main components of the reinforcement learning algorithm namely environment, reinforcement function and value function. According to the environment, states (s) and actions (a) are considered and values of state-action pairs (s,a) are estimated to construct the reward matrix. The reward matrix is used for formation of the Q matrix by estimating the value of Q of state-action pair (s_i,a), described in Eq. (2). The Q value determines what possible action the agent will take at a particular state s_i so that the next state s_{i+1} approaching to the goal state. After formation of the reward matrix, *Q matrix* is obtained by finite number of iterations using a learning parameter Υ. Maximum value of Q is calculated considering all actions at a particular state

$$\mathrm{Q}(s_i,a)=\mathrm{R}(s_i,a)+\Upsilon*\mathrm{Max}[\mathrm{Q}(s_{i+1},a)]. \quad (2)$$

3. Proposed Q-learning algorithm

In the paper, modified Q-learning algorithm has been applied on NSL-KDD data set to detect intrusions automatically. The proposed algorithm classifies on line network traffic data set either as '*anomaly*' or '*normal*'.

3.1. Developing reward matrix

In the proposed Q-learning algorithm, the reward matrix is developed in two phases: (i) initial reward matrix and (ii) final reward matrix. First by applying a particular cut on all attributes of the network data set, continuous attributes are discretized. Each cut is mapped as row while each attribute is mapped as column in the initial reward matrix. Number of cuts or rows of the initial reward matrix is not known a priori and determined at the end of the initial reward matrix (R) formation procedure. A cut value is applied on the attributes to discretizing the attribute set and using RST based discernibility matrix concept, reducts are

Table 2
Discernibility matrix using Table 1.

Objects	O1	O2	O3	O4	O5	O6	O7	O8	O9	O10
O1	–	–	–	–	–	–	–	–	–	–
O2	–	–	–	–	–	–	–	–	–	–
O3	*a,b,e*	*a,b,e*	–	–	–	–	–	–	–	–
O4	*a,b,c,d*	*a,b,c,d*	*a,c,d,e*	–	–	–	–	–	–	–
O5	*a,b,d*	*a,b,d*	–	*a,b,c*	–	–	–	–	–	–
O6	–	–	*a,c,d,e*	*e*	*a,b,c,e*	–	–	–	–	–
O7	*a,b,e*	*a,b,e*	–	*a,c,d,e*	–	*a,c,d,e*	–	–	–	–
O8	*a,b,c,d*	*a,b,c,d*	*a,b,c,d,e*	–	*a,c*	*b,c,e*	*a,b,c,d,e*	–	–	–
O9	–	–	*b,d,e*	*a,b,c,e*	*e*	–	*b,d,e*	*a,c,e*	–	–
O10	*a,b,e*	*a,b,e*	–	*a,c,d,e*	–	*a,c,d,e*	–	*a,b,c,d,e*	*b,d,e*	–

generated. Classification rules are derived for individual reducts and accuracy corresponding to each *reduct* is calculated by designing a rule base classifier. The *reduct*, providing highest accuracy is selected, representing the best action at that particular state or cut. The procedures of applying different cut and successively generating reducts and accuracy evaluation are continued until two successive cut provides same accuracy or monotonically decreasing accuracy. Accuracy is thresholded to frame the initial reward matrix with three discrete values [−1, 0, 1], representing different kind of actions, as given

$$
\begin{aligned}
r_{ij} &= -1 \text{ if Accuracy}(Red_i) < 90\% \\
&= 0 \text{ if } 90\% < \text{ Accuracy}(Red_i) < 95\% \\
&= 1 \text{ if } 95\% < \text{ Accuracy}(Red_i) < 100\% \\
&= \text{NR if attribute } j \notin Red_i, \\
&\text{where } Red_i \text{ provides highest accuracy for cut } i. \qquad (3)
\end{aligned}
$$

The initial reward matrix (R) is shown below

$$
R = \begin{bmatrix} NR & -1 & -1 & \dots & NR \\ NR & 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ NR & 1 & 1 & \dots & 1 \end{bmatrix}.
$$

Therefore, from the initial reward matrix, it has been observed that actions are taken based on the most significant attributes (*reduct*) only while other attributes remain inactive denoted by NR in R matrix.

From R, the final reward matrix (RF) is obtained by eliminating the columns having NR values in all rows. Dimension of RF matrix is now determined, which is less compared to R. If a particular attribute (column) does not belong to the *reduct*, selected for a particular cut (row) then respective elements of RF are set to −1, indicating insignificant attributes.

Algorithm initial reward matrix (R)

Repeat

Step 1: Apply cut i on continuous attributes and generate a set of reducts, say (red_1^i, red_2^i, …, red_q^i) where (red_n^i) represents n-th *reduct* obtained by applying cut i and $n = 1 \dots q$.

Step 2: Find red_b^i where Accuracy (red_b^i)=maximum(Accuracy (red_1^i), Accuracy (red_2^i) ,..........., Accuracy (red_q^i)).

Step 3: The attributes of red_b^i are assigned with values using Eq. (3)

Until (Accuracy (red_b^i)=Accuracy (red_b^{i+1})) OR (Accuracy(red_b^{i+2}) < Accuracy(red_b^{i+1}) < Accuracy(red_b^i)).

Algorithm final reward matrix (RF)

```
Input : Initial Reward Matrix R(n × m)
Output: Final Reward Matrix RF(n × p) where p ≤ m.

Step 1: For each m, ascertain whether all r_ij are NR.
a=0;
  Begin for (j=1 to m)
      Counter=0;
      Begin for (i=1 to n)
        if (r_ij==NR)
          Counter++;
      End for

      if (counter==number-of-cuts)
          Set Deleted-column[a++]=j;          /* a is the index
of Deleted Column
                              which keeps track of column to
be deleted */

  End for

Step 2: Remove the selected c ∈ Deleted-column[ ]
          //     c is the column to be deleted
    having all r_ij as NR.
p=0;
  Begin for (j=1 to m)
    flag=0;
    Begin for (k=1 to a)
        Check if (i==Deleted-column[k])
            Set flag=1;
    End for

    if (flag==0)
      Begin for (i=1 to n)
        r_ip=r_ij;
      End for
      p +=1;
  End for

Step 3: Replace the element NR with −1.
    Begin for (i=1 to n)
      Begin for (j=1 to p)
        check if (r_ip==NR)
          r_ip=−1;
        End for
    End for.
```

3.2. Modified Q-learning algorithm

From the final reward matrix, Q matrix has been developed where start state corresponds to a particular cut and the goal state is defined as the state at which maximum accuracy is achieved. All rows excluding the last row (goal state) of the Q matrix comprising of all zeroes. The last row contains all ones representing maximum classification accuracy would be achieved at the goal state. Learning procedure consisting of several episodes and continues till all the elements in the modified Q matrix become greater than '0' representing accuracy, which is acceptable (90–95%).

```
BEGIN


Input : Final Reward Matrix RF (n × p).

Step 1: Initialize all the elements of the Q Matrix, QM (n × p) to
zero.

Step 2: Assign values to the n-th state (i.e., goal state) of the
QM.

    For (j=1 to p)
    QM [n ] [j ]=RF [n ] [j ];
    End


Step 3: Derive a sparse matrix SM (r m × 3) from RF (n × p)
which keeps
track of i (where i = 1,2,...,n), j (where j = 1,2,...,p) and
assign 0/1 in correspondence to r_ij of RF.
```

Step 4: Noting down the i (where $i=1,\ldots,n$) having no action.

```
    no-action-size=0;
    Begin for (i=1 to n)
      Flag2=0;
        Begin for (j=1 to p)
          if (RF [i ] [j ] ≥0)
            Flag2=1;
        End for
          if (Flag2==0)
            no-action [no-action-size ++]=i;
    End for
```

Step 5: Initialize the Flag[] to 0.

Step 6: Starting running the episodes.

```
    Begin do while
      Count=0;
    /* Starting operation from i=0 (i.e., start state) and
    continuing operation until i=n (i.e., goal state) is attained */

      Begin while (SM [count ] [0 ] != (n−1))
        State=SM [count ] [0 ];
          Begin if ((SM [count ] [2 ]==0 )and(Flag [state
]==0))
              action-number=SM [count ] [1 ];
      Calculate MAX [QM (next-state, all-actions)]

      Update the Q-Matrix
          QM[state ][action-number ]=(RF[state ][action-
number ] +
                                              (ϒ*Max));

      Update sparse matrix SM (r × 3)

      Reinitialize the Flag[] array to 0.

      /* Checking whether all values of QM[][] has been
updated */
          Flag-end=0;
      Begin for (k=1 to a)
    if (SM [k ] [2 ]==0)
           Flag-end=1;
      End for

End do while (Flag-end==1);
```

Step 7: Output the Q Matrix, QM $(n \times p)$.

END

Table 3
Part of information system table of NSL-KDD data set.

Objects	CA_1	CA_2	CA_3	CA_4	CA_5	CA_6	Decision class
OB_1	13	118	2425	1	1	26	1
OB_2	0	44	0	4	3	255	1
OB_3	0	0	44	1	1	255	1
OB_4	0	53	55	511	511	255	2
OB_5	0	0	0	1	1	16	1
OB_6	0	54,540	8314	2	9	255	1
OB_7	0	0	0	228	9	255	1
OB_8	7570	0	44	1	1	255	1
OB_9	0	56	52	294	294	255	2
OB_{10}	0	192	0	2	2	93	2

Finally the Q matrix is formed where for each j, the highest q_{ij} value is marked representing i is the optimum cut value for attribute j. Initial reward matrix (R), final reward matrix (RF) and Q matrix (QM) are shown below, considering the information system given in Table 3. After applying the modified Q-learning algorithm, Q matrix is updated to final Q matrix (Q_{final}), shown below. In the Q_{final}, 1.64 is the highest value at the first column corresponds to attribute CA_2 and it occurs at the second row that corresponds to cut 5. Therefore, highest accuracy is achieved by applying cut 5 on attribute CA_2 while detecting intrusions considering Table 3.

Actions(Attributes)

$$R = \begin{array}{c} \\ State\ 0-Cut\ 4 \\ State\ 1-Cut\ 5 \\ State\ 2-Cut\ 6 \\ State\ 3-Cut\ 7 \\ State\ 4-Cut\ 8 \end{array} \begin{array}{c} \begin{array}{cccccc} CA_1 & CA_2 & CA_3 & CA_4 & CA_5 & CA_6 \end{array} \\ \begin{pmatrix} NR & -1 & -1 & NR & NR & NR \\ NR & NR & 0 & 0 & NR & NR \\ NR & 0 & 0 & 0 & NR & NR \\ NR & NR & 1 & 1 & NR & NR \\ NR & 1 & 1 & 1 & NR & NR \end{pmatrix} \end{array}$$

Initial reward matrix, R

Actions(Attributes)

$$RF = \begin{array}{c} \\ State\ 0-Cut\ 4 \\ State\ 1-Cut\ 5 \\ State\ 2-Cut\ 6 \\ State\ 3-Cut\ 7 \\ State\ 4-Cut\ 8 \end{array} \begin{array}{c} \begin{array}{ccc} CA_2 & CA_3 & CA_4 \end{array} \\ \begin{pmatrix} -1 & -1 & -1 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \\ -1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \end{array}$$

Final reward matrix, RF

Actions(Attributes)

$$Q = \begin{array}{c} \\ State\ 0-Cut\ 4 \\ State\ 1-Cut\ 5 \\ State\ 2-Cut\ 6 \\ State\ 3-Cut\ 7 \\ State\ 4-Cut\ 8 \end{array} \begin{array}{c} \begin{array}{ccc} CA_2 & CA_3 & CA_4 \end{array} \\ \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix} \end{array}$$

Initial Q matrix

Actions(Attributes)

$$Q_{final} = \begin{array}{c} \\ State\ 0-Cut\ 4 \\ State\ 1-Cut\ 5 \\ State\ 2-Cut\ 6 \\ State\ 3-Cut\ 7 \\ State\ 4-Cut\ 8 \end{array} \begin{array}{c} \begin{array}{ccc} CA_2 & CA_3 & CA_4 \end{array} \\ \begin{pmatrix} 1 & 0.8 & 1 \\ 1.64 & 1 & 1.44 \\ 0.8 & 1.8 & 1.64 \\ 1.44 & 1.8 & 1.8 \\ 1 & 1 & 1 \end{pmatrix} \end{array}$$

Final Q matrix, Q_{final}

4. Experimental results applied on synthetic data set

Data sets are generated synthetically and used as test data to detect intrusions either as '*anamoly*' or '*normal*' using the proposed method. Correlation between data sets is studied by evaluating

Table 4
Correlation and accuracy for different synthetic data set.

Synthetic dataset	Correlation	Accuracy (%)
Dataset1	−0.1376	96.86
Dataset2	−0.0072	95.44
Dataset3	0.0058	87.98
Dataset4	0.0079	88.98
Dataset5	1.0000	95.13

Table 5
Reduct vs. accuracy.

Reduct sets	Attributes	Classification accuracy (%)
R0	34,32,2,31	90.7
R1	34,32,2,33	90.7
R2	9,32,2,31	90.7
R3	9,32,2,33	90.7

Pearsons correlation coefficient, which is defined below

$$r = \frac{n\sum xy - (\sum x)(\sum y)}{\sqrt{(n(\sum x^2) - (\sum x)^2)}\sqrt{(n(\sum y^2) - (\sum y)^2)}},$$

where x and y are two variables and n represents the number of samples. Value of r can vary from -1 to $+1$. If x and y have a strong positive linear correlation, r is close to $+1$. Similarly, x and y have a strong negative correlation, r is close to -1. If there is no correlation, r is 0. We have considered a few sets of synthetic data which are highly, moderately or uncorrelated with the training data set. Considering the synthetic data set as on line instances, the proposed method calculates accuracy for detecting intrusions. Correlation and accuracy of the data sets are listed in Table 4. It has been observed that accuracy decreases as the value of correlation tends to 0 for obvious reason. Therefore, the proposed method able to classify different on line data sets generated in real time environment.

5. Performance verification

In the paper, NSL-KDD data set [29] is used for learning the environment with 42 attributes, out of which, 41 attributes are conditional and one is decision attribute. Among conditional attributes, 34 are continuous and seven are discrete attributes. Initially uniform cut value is applied to all continuous conditional attributes and in every cases reducts are generated. For instance, consider 200 objects as training data set on which cut 2 is applied and four reducts are generated. Accuracy for reducts is calculated taking 100 objects as test data, as shown in Table 5. Since all reducts are showing same classification accuracy so *reduct* R_0, with attributes 2, 31, 32 and 34 is taken to construct the initial reward matrix.

Same steps are repeated by applying cut 3–9 on all continuous attributes and reducts with highest classification accuracy are selected for constructing the initial reward matrix. Number of cuts or rows of the initial reward matrix is determined from the termination condition of the cut generation procedure. Cut-accuracy graph in Fig. 1 shows that accuracy corresponding to cut 9, cut 10 and cut 11 are decreasing monotonically (Table 6) and so the number of rows of the initial reward matrix is determined as 8. The goal state is defined corresponding to cut 9. Say, applying cut 3, *reduct* R2 = {2,3,9,21,22,29,35} is selected for providing highest classification accuracy, which is 96.4%.

Finally, attributes 4, 5, 9, 22, 28, 29, 31, 32, 33, 34 and 35 are used to forming the columns of the final reward matrix, as shown in Table 7. Therefore, the final reward matrix has eight rows and 11 columns. By applying modified Q-learning algorithm, final Q matrix is obtained, shown in Table 8. From this final Q matrix, cuts for different attributes are derived which corresponds to highest accuracy. Say, at attribute 4, cut 9, at attribute 5, cut 6 and so on are selected as optimum cut and applied on the new datasets that provides classification accuracy 98.2% in detecting intrusions.

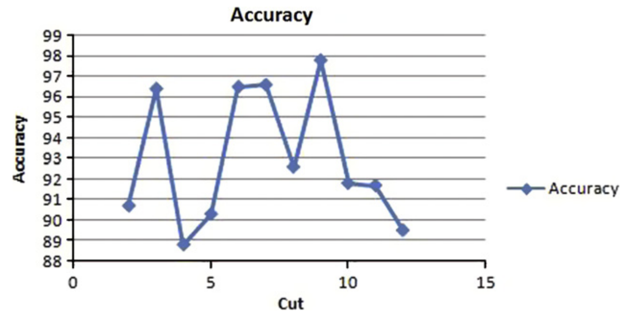

Fig. 1. Cut vs. accuracy.

Table 6
Cut and accuracy.

Cut	Accuracy (%)
2	90.7
3	96.4
4	88.8
5	90.3
6	96.5
7	96.6
8	92.6
9	97.8
10	91.8
11	91.7
12	89.5

Table 7
Final reward matrix.

Cut	Attribute numbers (*A*)										
	4	5	9	22	28	29	31	32	33	34	35
2	−1	−1	−1	−1	−1	−1	0	0	−1	0	−1
3	−1	−1	1	1	−1	1	−1	−1	−1	−1	1
4	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1
5	−1	−1	0	−1	−1	−1	−1	0	0	−1	−1
6	−1	1	−1	1	1	−1	1	−1	−1	−1	−1
7	−1	−1	−1	1	−1	1	1	−1	−1	−1	−1
8	−1	−1	−1	−1	−1	−1	0	−1	−1	−1	−1
9	1	−1	−1	1	1	−1	1	−1	−1	−1	−1

Table 8
Final Q matrix: optimum cut generation.

Cut	Attribute numbers (A)										
	4	5	9	22	28	29	31	32	33	34	35
2	0	0	0	0	0	0	0.8	0.8	0	0.8	0
3	0	0	1	1	0	1	0	0	0	0	1
4	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0.8	0	0	0	0	1.44	1.85	0	0
6	0	1	0	1.8	2.312	0	2.312	0	0	0	0
7	0	0	0	1	0	1.64	1.64	0	0	0	0
8	0	0	0	0	0	0	0.8	0	0	0	0
9	1	−1	−1	1	1	−1	1	−1	−1	−1	−1

Table 9
Classification accuracy (%) comparison.

Used cuts for discretization	Accuracy (%)
Random cut generated set 1	98
Random cut generated set 2	95
Random cut generated set 3	98
Random cut generated set 4	98
Random cut generated set 5	96
Random cut generated set 6	98
Proposed method	98

Table 10
Classification accuracy (%).

Name of classifier	Correctly classified instances (%)	Incorrectly classified instances (%)
Naive Beyes	70.3	29.7
RBF network	82.7	17.3
Lazy IB1	91.4	8.6
PART	94.2	5.8
NB Tree	94.2	5.8
Proposed method	98.2	1.8

We generate six different sets of cuts using random function and applied to 34 continuous attributes considering 200 training set of objects and 100 test set of objects. For each set of random cut classification accuracy is calculated and compared with the proposed method (Table 9) which generates optimum cut values.

Comparison of classification accuracy with the proposed classifier and different classifiers available in *WEKA* tool considering 10-fold cross validation model has been presented in Table 10.

6. Conclusions and future research

Development of on line IDS using modified Q-learning algorithm and *RST* is described here that detects intrusions with 98% accuracy. Reducing dimensionality of the system and with the selected feature set the environment is learnt to reach to the goal state [18]. In the paper discretization, feature selection and accuracy calculation are handled simultaneously, which reduces computational cost and build the classifier in a comprehensive way. It has been observed that for discretization of continuous attribute, if same cut is applied to all attributes, classification accuracy varies widely even for two consecutive values of cut. But combination of different cuts for different attributes yields best result of classification accuracy. The proposed method is tested with differently correlated data sets, showing effectiveness of the system in real time intrusion detection environment. It has been observed that the proposed method achieves higher classification accuracy with respect to other existing classifiers applied on the same domain.

Detection of class label is important to classify the objects accurately where class label of the objects is unknown. Depending on the attribute values either the object is classified in any of the class labels already exist in the data set or a novel class label likely to be assigned. Novel class detection is an important research area in data mining domain and has immense importance in intrusion detection. Characteristics of data changing with time and space and so handling of such data by utilization of existing knowledge opens new avenue of research. Designing of classifiers using different approaches and then fusing those classifiers surely improve classification accuracy in intrusion detection. However, its implementation in real life environment is challenging.

References

[1] S. Zanero, G. Serazzi, Unsupervised learning algorithms for intrusion detection, in: Network Operations and Management Symposium, 2008, NOMS 2008, IEEE.
[2] D. Dasgupta, F. Gonzalez, An immunity-based technique to characterize intrusions in computer networks, IEEE Trans. Evol. Comput. 6 (2002) 1081–1088.
[3] M. Mahoney, P. Chan, Learning rules for anomaly detection of hostile network traffic, in: Proceedings of the 3rd IEEE International Conference on Data Mining (ICDM-2003), Melbourne, FL, IEEE Computer Society, 2003, pp. 601–604.
[4] Shekhar R. Gaddam, Vir V. Phoha, Kiran S. Balagani, KMeans+ID3: a novel method for supervised anomaly detection by cascading K-means clustering and ID3 decision tree learning methods, IEEE Trans. Knowl. Data Eng. 19 (March (3)) (2007) 345–354.
[5] W. Fangfei, J. Qingshan, S. Liang, W. Nannan, An intrusion detection system based on the clustering ensemble, in: IEEE International Workshop on 16–18 April 2007, p. 12.
[6] L.J. Mark, K. Michael, C. Yiu-ming, J.Z. Huang, Agglomerative fuzzy K-means clustering algorithm with selection of number of clusters, IEEE Trans. Knowl. Data Eng. 20 (November (11)) (2008) 1519–1534.
[7] C. Zhang, S. Xia, K-means Clustering algorithm with improved initial center, in: Second International Workshop on Knowledge Discovery and Data Mining, 2009 IEEE, pp. 790–793.
[8] M. Jianliang, S. Haikun, B. Ling, The application on intrusion detection based on K-means cluster algorithm, in: International Forum on Information Technology and Application, IEEE, 15–17 May 2009, pp. 150–152.
[9] X. Xu, Adaptive intrusion detection based on machine learning: feature extraction, classifier construction and sequential pattern prediction, Int. J. Web Serv. Pract. 2 (1–2) (2006) 49–58.
[10] A.L. Strehl, L. Li, E. Wiewiora, J. Langford, M.L. Littman, PAC model-free reinforcement learning, in: ICML '06 Proceedings of the 23rd International Conference on Machine Learning.
[11] J. Asmuth, M.L. Littman, R. Zinkov, Potential-based shaping in model-based reinforcement learning, in: Proceedings of the 23rd AAAI Conference on Artificial Intelligence, AAAI Press, 2008, pp. 604–609.
[12] J. Asmuth, L. Li, M.L. Littman, A. Nouri, D. Wingate, A Bayesian sampling approach to exploration in reinforcement learning, in: Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence, 2009.
[13] P. Auer, T. Jaksch, R. Ortner, Near-optimal regret bounds for reinforcement learning, Adv. Neural Inf. Process. Syst. 21 (2009) 89–96.
[14] E. Brunskill, B.R. Leffler, L. Li, M.L. Littman, N. Roy, CORL: a continuous-state offset-dynamics reinforcement learner, in: Proceedings of the 24th Conference on Uncertainty in Artificial Intelligence, 2008, pp. 53–61.
[15] E. Even-Dar, Y. Mansour, Learning rates for Q-learning, J. Mach. Learn. Res. 5 (2003) 1–25.
[16] S.M. Kakade, On the sample complexity of reinforcement learning, Ph.D. Thesis, Gatsby Computational Neuroscience Unit, University College London, 2003.
[17] B.R. Leffler, M.L. Littman, T. Edmunds, Efficient reinforcement learning with relocatable action models, in: Proceedings of the 22nd Conference on Artificial Intelligence, 2007, pp. 572–577.
[18] S. Manju, M. Punithavalli, An analysis of Q-learning algorithms with strategies of reward function, Int. J. Comput. Sci. Eng. (IJCSE) 3 (2) (2011).
[19] T. Kohri, K. Matsubayashi, M. Tokoro, An adaptive architecture for modular Q-learning, in: IJCAI'97 Proceedings of the 15th International Joint Conference on Artifical Intelligence, vol. 2.
[20] A. Servin, D. Kudenko, Multi-agent reinforcement learning for intrusion detection, in: MATES '08 Proceedings of the 6th German Conference on Multiagent System Technologies, Springer-Verlag Berlin, Heidelberg, 2008.
[21] L. Panait, S. Luke, Cooperative Multi-Agent Learning: The State of the Art, Autonomous Agents and Multi-Agent Systems, vol. 11(3), Springer, 2005, pp. 387–434.
[22] T.S. Chou, K.K. Yen, J. Luo, Network intrusion detection design using feature selection of soft computing paradigms, Int. J. Computat. Intell. 4 (3) (2008) 196–208.
[23] M. Dash, H. Liu, Consistency-based Search in Feature Selection, vol. 151(1–2), Elsevier, December 2003, pp. 155–176.
[24] A.K. Das, J. Sil, An efficient classifier design integrating rough set and set oriented database operations, Appl. Soft Comput. J. 11 (2011) 2279–2285, Elsevier, Science Direct.
[25] R.W. Swiniarski, Rough sets methods in feature reduction and classification, Int. J. Appl. Math. Comput. Sci. 11 (3) (2001) 565–582.
[26] Y. Huang, X. Huang, N. Cercone, Feature selection with rough sets for web page classification, Transactions on Rough Sets, vol. 2, SpringerLink Publishers, 2004, pp. 1–13.
[27] W. Siedlecki, J. Sklansky, An automatic feature selection, Int. J. Pattern Recognition Artif. Intell. 2 (2) (1988) 197–220.

[28] A. Skowron, C. Rauszer, The discernibility matrices and functions in information systems, in: X. Slowinski (Ed.), Intelligent Decision Support-Handbook of Applications and Advances of the Rough Sets Theory, 1991, pp. 331–362.
[29] 〈http://iscx.ca/NSL-KDD/〉.

Nandita Sengupta has done her Bachelor and Master of Engineering from Bengal Engineering and Science University, Shibpur, India. She is pursuing Ph.D. in the same institute. She has 22 yr of working experience and last 11 yr in teaching. Presently she is working as Senior Lecturer in University College of Bahrain, Bahrain. Her area of interests are machine learning, soft computing, data mining, and network computing.

Jaydeep Sen obtained his Bachelor of Engineering (B.E.) degree from Bengal Engineering and Science University, Shibpurin the year of 2011 with First Class Honors in the field of Computer Science and Technology. He is currently pursuing Master of Technology (M.Tech) at Indian Institute of Technology, Kanpur (IIT K) in the Department of Computer Science and Engineering. His primary area of interests are applied soft computing in artificial intelligence, data analysis and machine learning.

Jaya Sil an alumnus of BESUS (Bengal Engineering and Science University, Shibpur) and JU (Jadavpur University), completed her Ph.D. in Engineering from JU, Kolkata, India. She has been working in Academics for last 25 yr and presently working as Professor in the Department of Computer Science and Technology Department in BESUS. She has more than 80 publications in International Conferences and Journals. Dr. Sil worked as Post-Doc Fellow in Nanyang Technological University, Singapore, on 2002–2003 and visited Heidelburg University, Germany, on 2007. Dr. Sil contributed several Book Chapters on different areas like data mining, image processing. Her areas of research include image processing, soft computing techniques, multiagent systems and bio-informatics.

Moumita Saha has completed her Bachelor of Technology from Meghnad Saha Institute of Technology, WBUT. She has completed her Master of Engineering from Bengal Engineering And Science University (BESUS), Shibpur. She is pursuing Ph.D. at Indian Institute of Technology, Kharagpur. She is performing her research work in the field of Data Mining and Machine Learning.