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

A R T I C L E   I N F O

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

A B S T R A C T

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.

© 2013 Elsevier B.V. All rights reserved.

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 accommodate 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
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, 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. Skowron [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 \(m_i\) is a \(|U| \times |U|\) matrix, in which the element \(m_{ij}\) for an object pair \((x_i,x_j)\) is obtained as

\[
m_{ij} = \left| \{a \in C : a(x_i) \neq a(x_j) \} \right| \quad i,j = 1,2,3,\ldots,n
\]

(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 \(m_{ii} = 0\). 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

\[
f(s) = (a \land b \land c \land d) \land (a \land b \land c \land e) \land (a \land b \land d \land e) \land (a \land b \land c \land d)
\]

\[
\land (a \land b \land c \land d \land e) \land (a \land b \land c \land e) \land (a \land b \land d \land e) \land (a \land b \land e)
\]

\[
\land (a \land b \land c \land d) \land (a \land b \land c \land e) \land (a \land c \land d) \land (a \land c \land e)
\]

\[
\land (a \land c \land d \land e) \land (a \land b \land c \land d) \land (a \land b \land c \land e)
\]

\[
\land (b \land d \land e) \land (b \land d \land c \land e) \land (b \land e)
\]

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

\[
f(s) = (a \land b \land c \land d) \land (a \land b \land c \land e) \land (a \land b \land d \land e) \land (a \land b \land c)
\]

\[
\land (a \land b \land c \land d \land e) \land (a \land c \land d \land e)
\]

\[
\land (b \land d \land e) \land (b \land d \land c \land e) \land (b \land e)
\]

\[
\land (a \land b \land c \land d) \land (a \land b \land c \land e) \land (a \land c \land d)
\]

\[
\land (a \land b \land c \land d \land e) \land (a \land c \land d \land e)
\]

\[
\land (b \land d \land e) \land (b \land d \land c \land e) \land (b \land e)
\]
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

\[ f(s) = (a \land e) \land (a \land v \lor b \land v d). \]

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 (i): \( (c \land e) \).

From (ii): \( (a \land c \land d) \).

From (iii): \( c \land (b \lor d) \).

From (iv): \( (a \land e) \land (c \land (b \lor d)) \).

So

\[ f(s) = (a \land e) \land (c \land (b \lor d)) = (a \land e) \land (c \land b) \lor (c \land d) = (a \land e \land c) \lor (a \land e \land c \land d). \]

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 online 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 \( \gamma \). Maximum value of Q is calculated considering all actions at a particular state

\[ Q(s_i, a) = R(s_i, a) + \gamma \max [Q(s_{i+1}, a)]. \]

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
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

$$r_{ij} = -1 \text{ if } \text{Accuracy}(\text{Red}_i) < 90\%$$

$$= 0 \text{ if } 90\% < \text{Accuracy}(\text{Red}_i) < 95\%$$

$$= 1 \text{ if } 95\% < \text{Accuracy}(\text{Red}_i) < 100\%$$

$$= \text{NR if attribute } j \text{ is not in Red}_i,$$

where Red$_i$ provides highest accuracy for cut $i$. \hfill (3)

The initial reward matrix $(R)$ is shown below

$$R = \begin{bmatrix}
\text{NR} & -1 & -1 & \cdots & \text{NR} \\
\text{NR} & 0 & 0 & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
\text{NR} & 1 & 1 & \cdots & 1 \\
\end{bmatrix}$$

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 as 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 $(\text{Red}_1, \text{Red}_2, \ldots, \text{Red}_q)$ where $(\text{Red}_i)$ represents $n$-th reduct obtained by applying cut $i$ and $i = 1 \ldots q$.

Step 2: Find $\text{Red}_i$ where Accuracy

$$(\text{Red}_i)$ = maximum$(\text{Accuracy}(\text{Red}_1), \text{Accuracy}(\text{Red}_2), \cdots, \text{Accuracy}(\text{Red}_q))$$

Step 3: The attributes of Red$_i$ are assigned with values using Eq. (3)

Until $(\text{Accuracy}(\text{Red}_i) = \text{Accuracy}(\text{Red}_i+1))$ OR $(\text{Accuracy}(\text{Red}_i+2) < \text{Accuracy}(\text{Red}_i+1) < \text{Accuracy}(\text{Red}_i))$.

**Algorithm final reward matrix $(RF)$**

Input : Initial Reward Matrix $R(n \times m)$

Output: Final Reward Matrix $RF(n \times p)$ where $p \leq m$.

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 $(r_{ij}=\text{NR})$

Counter++;

End for

if (counter = = number-of-cuts)

Step 2: The attributes of reduct $i$ are assigned with values using Eq. (3)

Until $(\text{Accuracy}(\text{Red}_i) = \text{Accuracy}(\text{Red}_i+1))$ OR $(\text{Accuracy}(\text{Red}_i+2) < \text{Accuracy}(\text{Red}_i+1) < \text{Accuracy}(\text{Red}_i))$.

Set Deleted-column[$a++$] = $j$; \hfill /* $a$ is the index of Deleted Column which keeps track of column to be deleted */

End for

Step 2: Remove the selected $c \in \text{Deleted-column[ ]}$

\begin{verbatim}
// 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 $(i=k \in \text{Deleted-column[k]})$

Set flag=1;

End for

if (flag=0)

Begin for $(i=1$ to $n)$

$r_{ip}=r_{ij}$;

End for

$p++;=1$;

End for

End for

Step 3: Replace the element NR with $-1$.

Begin for $(i=1$ to $n)$

Begin for $(j=1$ to $p)$

check if $(r_{ip}=\text{NR})$

$r_{ip}=-1$;

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 $RF(n \times p)$.

Step 1: Initialize all the elements of the Q Matrix, $QM(n \times p)$ 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 $SM(r \times 3)$ from $RF(n \times p)$ which keeps track of $i$ (where $i=1,2,\ldots, n$), $j$ (where $j=1,2,\ldots, p$) and assign 0/1 in correspondence to $r_{ij}$ of $RF$. 

END
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 ($RF[i, j] \geq 0$)
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 ($SM[0] = (n - 1)$)
State = $SM[0];$
Begin if ($SM[2] = 0$ and $Flag[0] = 0)$
action-number = $SM[1];$
Calculate MAX $QM$ (next-state, all-actions)
Update the $Q$-Matrix $QM[state][action-number] = (RF[state][action-number]+($\gamma$ * Max));
Update sparse matrix $SM (r \times 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

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. 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 $CA_4$ and it occurs at the second row that corresponds to cut 5. Therefore, highest accuracy is achieved by applying cut 5 on attribute $CA_4$ while detecting intrusions considering Table 3.

### Actions (Attributes)

<table>
<thead>
<tr>
<th>State 0—Cut 4</th>
<th>$CA_1$</th>
<th>$CA_2$</th>
<th>$CA_3$</th>
<th>$CA_4$</th>
<th>$CA_5$</th>
<th>$CA_6$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Q_{final}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| Initial reward matrix, $R$ |

<table>
<thead>
<tr>
<th>State 0—Cut 4</th>
<th>$CA_1$</th>
<th>$CA_2$</th>
<th>$CA_3$</th>
<th>$CA_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Q_{final}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Final reward matrix, $RF$

<table>
<thead>
<tr>
<th>State 0—Cut 4</th>
<th>$CA_1$</th>
<th>$CA_2$</th>
<th>$CA_3$</th>
<th>$CA_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Q_{final}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**4. Experimental results applied on synthetic data set**

Data sets are generated synthetically and used as test data to detect intrusions either as ‘anomaly’ or ‘normal’ using the proposed method. Correlation between data sets is studied by evaluating
and four reducts are generated. Accuracy for reducts is calculated considering 200 objects as training data set on which cut 2 is applied attributes and in every cases reducts are generated. For instance, Initially uniform cut value is applied to all continuous conditional attributes, 34 are continuous and seven are discrete attributes. conditional and one is decision attribute. Among conditional environment with 42 attributes, out of which, 41 attributes are.

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 R0, 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.
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 attribute, if same cut is applied to all attributes, 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.

Table 10 Classification accuracy (%).

<table>
<thead>
<tr>
<th>Name of classifier</th>
<th>Correctly classified instances (%)</th>
<th>Incorrectly classified instances (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Naive Bayes</td>
<td>70.3</td>
<td>29.7</td>
</tr>
<tr>
<td>RBF network</td>
<td>82.7</td>
<td>17.3</td>
</tr>
<tr>
<td>Lazy IB1</td>
<td>91.4</td>
<td>8.6</td>
</tr>
<tr>
<td>PART</td>
<td>94.2</td>
<td>5.8</td>
</tr>
<tr>
<td>NB Tree</td>
<td>94.2</td>
<td>5.8</td>
</tr>
<tr>
<td>Proposed method</td>
<td>98.2</td>
<td>1.8</td>
</tr>
</tbody>
</table>

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.

### References


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, Shibpur in 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 Heidelberg 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 bioinformatics.

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.