International Journal of Innovative
Computing, Information and Control
Volume 8, Number 12, December 2012

ICIC International ©2012 ISSN 1349-4198
pp. 8063–8076

CELLULAR LEARNING AUTOMATA APPROACH FOR DATA CLASSIFICATION

Mansour Esmaeilpour[1], Vahideh Naderifar[1] and Zarina Shukur[2]

[1]Department of Computer Engineering
Hamedan Branch, Islamic Azad University
Hamedan, Iran
{ esmaeilpour; naderifar }@iauh.ac.ir

[2]Computer Science Department
Faculty of Information Science and Technology
National University of Malaysia
43600, Bangi, Selangor, Malaysia
zs@ftsm.ukm.my

Received August 2011; revised December 2011

Abstract. *Data classification is a process that can categorize data to achieve the relationship between attributes and extract the suitable rules for prediction process. There are different learning methods in machine learning of which each has both advantages and disadvantages. Each type provides a better and interesting position, data and special structure. These methods have differences in the manner of implementation, understandability and speed of response and each is included in a special field of the data classification. Learning process in machine learning is the most important part which causes to elevate the power of a model and can learn the trained problem more quickly and work with it. In this paper, it will present a new method for data classification by Cellular Learning Automata. This method includes three stages. In order to show the power of this model, we have tested it on several types of online dataset and study it in terms of the learning speed, accuracy and simplicity in implementation with some other models and the simulated results demonstrate that the presented method provides acceptable and better answers and that one can use the proposed method for data classification.*
Keywords: Data classification, Data mining, Accuracy, Cellular learning automata

1. **Introduction.** Regarding to increasing the size of data has been important in analyzing the knowledge and extracting the valid relationship in the datasets. So use of the suitable method in analyzing the knowledge leads to decrease of the error and increase of the prediction and performance. There are many applications for machine learning in which the most important one is the data mining. Each sample which is used in dataset with use of machine learning algorithms is expressed by a set of features. These features may be continuous, classified or binary [23]. If samples have been specified with a label, then, calls supervised learning, otherwise calls unsupervised learning. Another type of method is the reinforcement learning and this method contains an agent that studies environment which may receive either the reward or penalty. The action which is selected is then applied to the environment; the environment investigates it and reward or penalty is chargeable to it which will update the model with these values. In this paper, will be used the method which is based on the reinforcement and supervised learning. The aim of using data classification is to obtain special relationship between data. Therefore, firstly it will be needed to understand the data. This involves the attributes, its type and its range. Any faults data should be removed and this process is called cleaning process [5].

Then, the results will be reached by reduction [7] and we can do this by deleting some features in the dataset [8] and then it will be discrete dataset. For detail explanation, we can refer to [3,4]. Many algorithms have been presented for work on data classification which is referred to as follows:

1.1. **Decision tree.** Decision tree [10] is a tree which classifies samples on the basis of feature of each sample. Each node in decision tree indicates a feature in dataset which should be classified and each branch shows the value that the node can gain. The main problem of the decision tree is the high expense of its fabrication which is the NP-Complete. Different methods have been created to find more suitable features which can be referred to as [9]. One of the most evident algorithms of decision tree fabrication is C4.5. The last comparison in which the decision tree has been made with other methods is given in [11] and finally Olcay and Onur [12] studied parallelization of the C4.5 algorithm.

1.2. **Perception method.** Perceptional meaning has been studied by [13]. First, a single layer perception is studied and then will be dealt with a multilayer perception. If x_1 and x_2 are inputs and w_1 and w_2 are the weights (between $[-1, 1]$, then, the output is $\Sigma x_1 w_1$ and output acts as a threshold and if the value of output is more than threshold, output 1, otherwise is 0. There are many works for optimization of this method such as [14]. Freand and Schapir presented a new algorithm called the voted perception which saved much information about training and they were used for obtaining and prediction [15]. This method should be repeated to give/provide a better answer and was good for binary models but should be practiced for higher classes.

1.3. **Multilayer perception.** This model was presented for solving the above problem [15]. In the next works, hidden layers were used for solving complex problems and can calculate better and more precise answer by increasing and decreasing the number of neurons [16].

1.4. **Naïve Bayes (NB) classification.** NB is a simple network which uses graphs with only one unit and some children with strong hypothesis of independence of children node inside other parts. Advantage of this method is the training time of calculation [22]. If features are numerical, they should be preprocessed [17]. The major form of this method is not consistent with datasets which have many features and this needs more space and time for creation of graphs and the major form needs classifying in most cases and there are other methods [18,20].

1.5. **Learning automata.** Learning automata is another method for data analysis. It has been introduced by Narendra and Thathachar [6]. Extending it using cellular automata [23] which produce cellular automata (CA) give extra benefits to the learning process. It is a powerful mathematical model for many decentralized problems and phenomena. The main idea of CLA, which is a subclass of stochastic cellular automata, is using learning automata to adjust the state transition probability of stochastic cellular automata that it will be discussed more in next section. This paper is organized as follows. In Section 2, the training method that is CLA is explained. Then, in Section 3, the demonstration of the steps taken as well as the implementation is discussed. The experiment of this approach on several online datasets which aim at the representation of operational power of the proposed model is presented in Section 4. The result will be compared with against methods. Section 5 is the discussion and Section 6 concludes the work.

2. **Background of the Cellular Learning Automata.** Tsypkin shown reinforcement model to solve the problems of determination of optimal parameter and applied it to hill climbing techniques [1]. Tsetlin started working on learning automata at the same time [24]. The concept of learning automata was proposed by Tsetlin for the first time in 1973. Other researches introduced regarding the problems were finding an optimal action between permitted actions in stochastic automata [19]. However, most of attempts in learning automata were done by Tsetlin. Thereafter, an automaton updates its number of actions for which the result is the reduction of number states compression of deterministic automata. The first attempts in this case were done by fu from aspect of pattern recognition, parameter estimation, and game theory [21]. All the studies until the late 1980s have been presented by Thathachar in his book [6] and also several examples and application of learning automata has been presented by Najim and Poznyak [2].

2.1. **Learning automata.** A learning automaton is composed of two parts: 1) a stochastic automata with number of limited actions and a stochastic environment that the automata are associated with them, 2) learning algorithm in which automata will learn optimal action by using that action. Each action selected by potential environment is assessed and the answer is given to a stochastic automata. A stochastic automaton uses this answer and selects its action for the next stage. Figure 1 shows the relationship between the learning automata and environment.

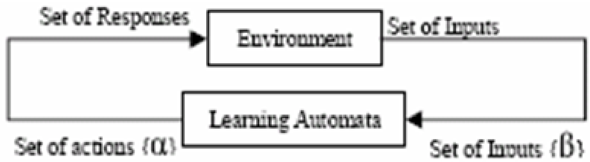


Figure 1. Stochastic learning automata

2.2. **Stochastic automata.** Stochastic automata is shown with quintuple $\{\alpha, \beta, F, G, \Phi\}$ in which $\alpha = \{\alpha_1, \ldots, \alpha_m\}$ is the set of automata actions, $\beta = \{\beta_1, \ldots, \beta_m\}$ is the set of automata inputs, F is production function and automata new state, G is an output function that maps the current state and input into the current output. Φ is a set of automata internal state. Set of α contains automata action and the automata chooses an action between its action and sends it to environment. β determined the set of automata inputs. F and G map current state of automata input into next output. Stochastic learning automata are divided into two groups: fixed-structure automata and variable-structure automata. In a fixed structure, probability of the automata action is fixes but in a variable structure, probability of automata action is updated and each repetition is based on environment response without delay for which that update is done by learning algorithm.

2.3. **Environment.** Environment can be shown with $E = \{\alpha, \beta, c\}$ in which $\alpha = \{\alpha_1, \ldots, \alpha_m\}$ is set of inputs, $\beta = \{\beta_1, \ldots, \beta_m\}$ is set of outputs and $c = \{c_1, c_2, \ldots, c_r\}$ is set of penalty probabilities. When value β_i is either 0 or 1, environment is called P-Model (Probabilistic model). $\beta_i(n) = 1$ as penalty and $\beta_i(n) = 0$ as reward. In the case of environment of Q-Model, $\beta_i(n)$ is finite output of set with more than two values between $[0, 1]$. In S-Model, $\beta_i(n)$ is a continuous random variable within the range $[0, 1]$. c_i is a set of penalty probability. In a stationary environment, values c_i remain unchanged while in non stationary environment, these values change over time.

$$c_i = \text{Prob}\{\beta(n) = 1 | \alpha(n) = \alpha_i\}, \quad i = 1, 2, \ldots, r \tag{1}$$

if action a_i is selected in stage and receives a desirable answer, probability of $P_i(n)$ increases and other probabilities decrease. For an undesirable answer, probabilities of $P_i(n)$ decrease and other probabilities increase. Anyway, changes are made in such a way that the sum of $P_i(n)$ remains equal to one. Below we can see an example of the learning algorithm in variable learning automata in the P-Model. The following algorithm is an example of linear learning algorithms in the variable automata.

Desirable answer

$$P_i(n+1) = P_i(n) + a\,[1 - P_i(n)] \tag{2.1}$$

$$P_j(n+1) = (1-a)P_j(n) \quad \forall j,\ j \neq i \tag{2.2}$$

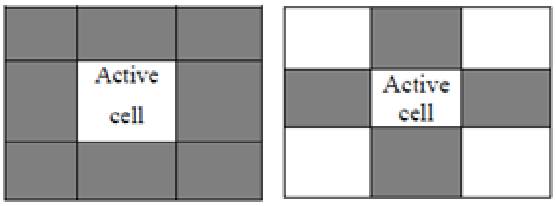
Undesirable answer

$$P_i(n+1) = (1-b)P_i(n) \tag{3.1}$$

$$P_j(n+1) = (b/r - 1) + (1-b)P_j(n) \quad \forall j,\ j \neq i \tag{3.2}$$

where a is the reward parameter and b is the penalty parameter and r is the number of actions. With regards to the values of a and b, when $a = b$, is said, $\mathrm{L_{RP}}$ (Linear Reward Penalty), when b be very low than a, is said, $\mathrm{L_{R\&P}}$ (Linear Reward epsilon Penalty) and when b equals to zero, is said, $\mathrm{L_{RI}}$ (Linear Reward Inaction).

2.4. **Cellular automata.** A cellular automaton (CA) is a mathematical model to the system in which number of simple components cooperates to produce complex patterns. In fact, a cellular automata is discrete dynamical system and communication between its cells is limited which is based on local interaction [25]. CA composes of lattice of cells and set of rules. Each square is called cell with each cell having two states that are black and white colors. The rules of cellular automata determine how the states changes. Figure 2 shows an example of well-known neighbor of CA as follows.



(a) (b)

Figure 2. Show the Moor (a) and Von Neumann (b) neighborhood

For each cell, there is a set of cells around active cell called neighbors. In Figure 2(a) the "neighbors" of a cell are eight squares interacted and in Figure 2(b), the neighbors are declared as up, down, left and right. A cellular automaton is based on neighbor's behavior and past experience. The neighbor is defined as a set of cells around the activity of the cell such as for example with distance of two or less than two. A cellular automaton is based on some of the local rules and the rules of automata can usually be designed by the user. Individual cell is affected by its local neighbor's rules. An automaton is updated based on local rules. The value of neighbors will impact the determination of new state of each cell. Formally, a cellular automaton can be defined as follows:
A d-dimensional cellular automata is a structure of A $= (Z^d, \Phi, N, F)$ where

1. Z^d is a lattice of d-tuples of integer number of which this lattice could consist finite lattice, infinite lattice or semi-finite.
2. $\Phi = \{1, \ldots, m\}$ is a finite set of states.
3. $N = \{x_1, x_2, \ldots, x_m\}$ is a finite subset of Z^d called the neighborhood vector, ($x_i \in Z^d$).

4. F is the local rule of the cellular automata. This rule can be defined by the users.

The cellular automata rules determine how the states change and how the set of cells are neighbors with each other of which this rule can be defined by the users. Figure 3 is an example of rules.

The value of neighbors will impact on the determination of the new state of each cell.

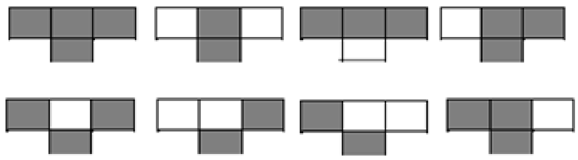


FIGURE 3. How to create neighborhood of several cells

In Figure 3, there are eight rules because there are eight possible ways to set the states of the cell's three neighbors.

2.5. **Cellular leaning automata.** A cellular automaton is unable to solve most of problems collectively [26]. Regarding the same issue and weakness as for cellular automat with combination of two recent models (learning automata and cellular automata) was created a new model called the cellular learning automat. It is a powerful mathematical model for many decentralized problems and phenomena. The main idea of cellular learning automata, which is a subclass of stochastic cellular automata, is to use the learning automata to adjust the state transition probability of stochastic cellular automata. A cellular learning automata is a cellular automata in which a learning automata or more than one is assigned to every cell. There are rules that the cellular learning automata is operating under it. Those rules and action selected by the neighbors learning automata of any learning automata determine the reinforcement signal to the learning automata residing in each cell based on that give reward or penalty to relative path. Giving penalty or reward will update the CLA structure. A cellular learning automata is formally defined below.

A d-dimensional cellular learning automata is a structure of $\{Z^d, \Phi, A, N, F\}$ where

1. Z^d is a lattice of d-tuples of integer number which this lattice could consist finite lattice, infinite lattice or semi-finite.
2. $\Phi = \{1, \ldots, m\}$ is a finite set of states.
3. A is collection of learning automata (LA) each of which is assigned to one cell of the CLA. Each cell can have one LA or more.
4. $N = \{x_1, x_2, \ldots, x_m\}$ is a finite subset of Z^d called the neighborhood vector ($x_i \in Z^d$).
5. F is the local rule of the cellular automata. This rule can be defined by users.

The operation of cellular learning automata can be described as follows. Each learning automata in cellular learning automata chooses an action. The selection action may be chosen on the basis of random or previous observation. Selected action causes movement of the learning automaton from one cell to another. The rule of cellular automata determines the reinforcement signal to each learning automaton residing in that cell like Figure 4. Learning automata actions in every active cell receive a reward or penalty based on the current learning automata actions in the neighbor learning automata and rules. Receiving reward or penalty update the internal structure of the learning automata. Rewarding or giving penalties continues until the system arrives to a sustainable state. Finally, each learning automaton updates its action probability vector on the basis of supplied

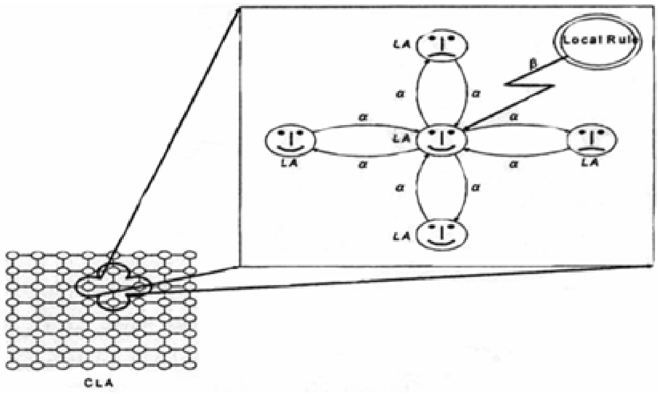


Figure 4. Cellular learning automata [26]

reinforcement signal and the selected action. This process continues until the desired result is obtained.

3. Data Classification Using Cellular Learning Automata.

The data classification process generally involves three main steps. The first and second steps are to develop the model and the third step is to test of the model. The desired data must be transformed in a discrete form prior to that process. In this work, the first step involves extracting the pattern from original source by using complete graph. In the second step CLA will be used to identify the strong patterns. Strong pattern is a pattern that has been repeated more than others. Finally the third step, in a raw data the pattern model will be tested by calculating the compliance rate based on the reward and penalty value.

At first all pair wise attributes it will be extracted with repeat number in dataset by using complete graph and these information is recorded in a dynamic array, after that these pair wise attributes will be combined in order to extract the strong pattern for prediction that this section use the CLA for combination. Dynamic arrays are used in this section where its size can change during runtime, leads to decrease the physical processing time and running time. Data classification will be done as follows.

3.1. First step: extraction of pattern using complete graph.

Complete graph is a graph that each node is connected to each others. In this work a graph has been assigned to one class of dataset, a node represents an attribute, an edge represents a pair wise attribute that is a pattern, and the weight of an edge represents the effective rate of the pattern.

Implementation wise, at this step, list of data that has certain pattern and its effective rate will be recorded. The information is recorded by using the following format:

<Data#> <atr#>.<atrValue>→<atr#>.<atrValue> <EffectiveRate> <Class>

Example 3.1. *Extracting patterns using complete graph. Given a dataset with 5 rows and 4 attributes as in Table 1. The attribute values of the data range from 0 to 3, with two classifications that are 0 and 1.*

From Table 1 for class 1, attribute #1 and #2 for data number 1 is similar to attribute #1 and #2 for data number 4 (the bold number). Therefore, we record this information as:

$$1,4 \quad 1.1 \rightarrow 2.0$$

The effective rate refers to the repetition of data that has similar pair wise of attributes values $(1,4)$, whilst the class value is copied directly from the table, so the effective rate

Table 1. Sample of the dataset

Data number	Attribute #1	Attribute #2	Attribute #3	Attribute #4	class
1	**1**	**0**	**1**	2	1
2	2	0	2	3	0
3	2	0	1	3	0
4	**1**	**0**	**1**	1	1
5	2	**0**	**1**	1	1

and the class are 2 and 1 respectively. If there is no similarity of a pair wise of attributes value between two or more data, it can be recorded as single information. For example, attribute #1 and #4 for data number 1 is not similar to any other data, and therefore, we record this as:

$$1 \quad 1.1 \rightarrow 4.2$$

The effective rate and the class are 1 and 1 respectively. This information will be recorded in the extraction table as in Table 2. Since the dataset has 4 attributes, it should have 4 pair wise of attributes value; hence each data should appear 4 times in the extraction table, as with data number 1 in Table 2. If it has n attributes, therefore the same data number should appear ${}^nC_2$ times in the extraction table.

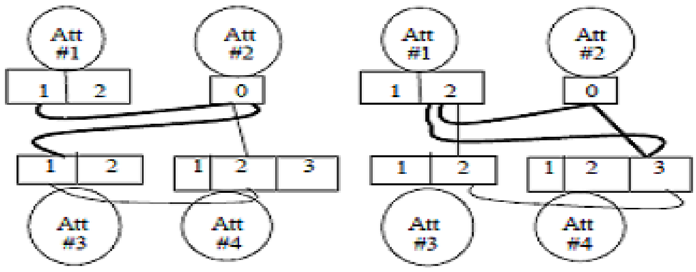
$$ {}^nC_2 = \frac{n!}{(n-2)!2!} \tag{4}$$

Table 2. Pair wise attributes

Row number	Pair wise attributes	Effective rate	Class value
1, 4	1.1→2.0	2	1
1, 4	1.1→3.1	2	1
1	1.1→4.2	1	1
1	2.0→4.2	1	1
1	3.1→4.2	1	1
1, 4, 5	2.0→3.1	3	1
2, 3	1.2→2.0	2	0
2, 3	1.2→4.3	2	0
2, 3	2.0→4.3	2	0
2	1.2→3.2	1	0
2	2.0→3.2	1	0
2	3.2→4.3	1	0
3	1.2→3.1	1	0
3	2.0→3.1	1	0
3	3.1→4.3	1	0
4	1.1→4.1	1	1
4	2.0→4.1	1	1
4	3.1→4.1	1	1
5	1.2→2.0	1	1
5	1.2→3.1	1	1
5	1.2→4.1	1	1
5	2.0→4.1	1	1
5	3.1→4.1	1	1

In order to create Table 2 from Table 1 the following processes are done: for some class, each row of dataset is run on a complete graph of properties. Node represents the attribute and edge represents pair wise attributes. Furthermore, the state of the nodes represents the value of the attributes. For example, attribute 1 has 2 values: 1 and 2. Therefore, in Figure 5, node with label "Atr#1" has 2 states (1 and 2). When the first data (in Table 1) is read, an edge will be created from atr#1.1 to atr#2.0. One graph is for one class. Therefore, this first graph is for class 1 (Figure 5(a)). The second data is for class 0, then a new graph will be created (Figure 5(b)), for example edge from atr#1.2 to atr#2.0.

The repetition of edge creation represents the effective rate. For example, if the edge is created twice, then the rate is 2. In Figure 5, the bold line shows the edge with rate 2 that has been gotten two times repeated.



(a) For class 1 (b) For class 0

FIGURE 5. An isolation mode of the pair attributes based on Table 2

This work will be done for all class separately, and after execution of the above algorithm, will be achieved the first and second columns of Table 2.

3.2. **Second step: extraction of strong pattern using cellular learning automata.** In this step, the strong pattern in each class will be extracted. In order to perform this work, CLA is used. CLA is able to combine two-items and extract the amount of influence on them. A cell of CLA represents row number. Neighborhoods represent a group of data that has similar pattern. Each cell of CLA has one LA that will record probability actions of each neighbor. In each LA α_i is set of row number and reward will be given ($\beta_i(n) = 1$) if "neighbor rate" > "adaptation rate" otherwise the pattern receives penalty ($\beta_i(n) = 0$). c_i is penalty rate. Adaptation rate is determined based on specific percentage of adaptation two group neighbors. Neighbor rate is equals to the occurrence of similar neighbor in any patterns. Implementation wise, the result from step 1 become an input to this second step. The neighbor rate refers to the occurrence of each row in Table 2, whilst the adaptation rate is set by the user. A neighbor of CLA will be created if the row number adaptation (neighbor rate) is greater than adaptation rate.

Proposed of the reward and penalty will increase and decreases the probability actions. (If ($\beta_i(n) = 1$)) then one pattern get rewards and use Equation (2.1), otherwise get penalty and use Equation (3.1). In this section reward rate is equal to penalty rate, $a = b = 0.5$). As regards, reinforcement is as important as weakness, has been used $\mathrm{L_{RP}}$ learning model. Initially all patterns has the following probability actions value,

$$\forall i,\ i <= n \quad \rho_i = \frac{1}{n} \tag{5}$$

where n is number of pair wise attributes.

The output from this second step is a pattern model of the respective dataset. This model will be recorded is; the row numbers that contain the pattern of the pair wise attributes (Table 2), the pair wise attributes itself, and the effective rate. Based on that, the reinforcement value can be calculated. Table 3 is a result of running CLA on Table 2. In this work, we set the adaptation rate as more than 50%. The following explains steps by steps of how CLA works based on Table 2 in Example 3.1.

Example 3.2. *Extracting the strong patterns. In Table 2, 1 and 4 is neighbors and has adaptation rate more that 50% with 1, 4 and 5, so according to this rule, these neighbors can synthesize. By reading the first row of Table 2, a neighbor of 1 and 4 will be created for the first time in the CLA. The creation of the first neighborhood is called the first reinforcement of the neighbor if get reward by another neighborhood that has adaptation greater than adaptation rate. This process will be done for each pattern that has more than one member.*

Table 3. The CLA process on Table 2

Stage number	α_i	Neighbor rate	Neighbor rate >50%	Neighborhood status	$\beta_{\alpha i}$(i)	a and b	Probability actions ρ_i
0	–	–			–	$a = b = 0.5$	All 1/6 = 0.17
1	1, 4	0%	No	Weakness	$\beta_{1,4}(1) = 0$	$a = b = 0.5$	0.085
2	1, 4	100%	Yes	Reinforcement	$\beta_{1,4}(2) = 1$	$a = b = 0.5$	0.54
3	1	ignore	No	–	–	–	–
4	1	ignore	No	–	–	–	–
5	1	ignore	No	–	–	–	–
6	1, 4, 5	67%	Yes	Reinforcement	$\beta_{1,4,5}(1) = 1$	$a = b = 0.5$	0.59
7	2, 3	0%	No	Weakness	$\beta_{2,3}(1) = 0$	$a = b = 0.5$	0.085
8	2, 3	100%	Yes	Reinforcement	$\beta_{2,3}(2) = 1$	$a = b = 0.5$	0.54
9	2, 3	100%	Yes	Reinforcement	$\beta_{2,3}(3) = 1$	$a = b = 0.5$	0.77
10	2	ignore	No	–	–	–	–

Regarding to row number 6 in Table 3, there are two group neighbors that have adaptation to each other. This adaptation is 2 members from 3 members, in other word 2/3, so neighbor rate ≈ 67%. The neighbor rate equation is as follow:

$$\text{Neighbor rate} = \frac{\text{The number of matching rows for both pair wise attributes}}{\text{The total number of row for both pair wise attributes}} \tag{6}$$

If Probability of actions is more than 0.5 then, it means is there are neighbors that has been reinforced and will be combined all pair wise attributes of them. Table 4(a) and Table 4(b) will be achieved after pair wise attributes combination.

1	4

Figure 6. Show the cells neighbor in CLA

If consequences data happened to be 1, 4 again (as in Table 2, row 2), then second reinforcement will be done on the same neighbor and get reward, otherwise get penalty (refer to Figure 8). Second (and above) reinforcement will generate an output as in Table 4(a). So, each data in Table 4 will have at least two pair wise attributes. Data in Table 2 with effective rate that is not more than adaptation rate will be dropped. They are also known as weak patterns. For data in row 7 of Table 2, a neighbor of 1, 4 will be reinforced for the third time, and neighbor 5 will be created, as in Figure 7. Neighbor of three, that is 1, 4, 5 will be reinforced for the first time and get reward. This reward lead to reinforcement the pair wise attributes and also, if adaptation is less than and equal to adaptation rate then it will penalty and this pair wise will be receded from this neighbors.

Figure 7. Show the cells neighbor in CLA

Table 4. (a) Sequence pattern compilation for class 1, (b) sequence pattern compilation for class 0

(a)

Row number	Pair wise attributes	Effective rate	Reinforcement value
1, 4	$1.1 \to 2.0$, $1.1 \to 3.1$, $2.0 \to 3.1$	2	$3 * 2 = 6$
1, 4, 5	$1.1 \to 2.0$, $1.1 \to 3.1$, $2.0 \to 3.1$	2	$3 * 2 = 6 \to 6 + 1 = 7$
	and $2.0 \to 3.1$	1	$1 * 1 = 1$

(b)

Row number	Pair wise attributes	Effective rate	Reinforcement value
2, 3	$1.2 \to 2.0$, $1.2 \to 4.3$, $2.0 \to 4.3$	2	$3 * 2 = 6$

In general, if n subsequent row number combination repeats the previous combination or subset of them, then neighbors will be strengthened and each time of repetition they will get reward of leaning automata in each cell. Regarding to Figure 8, if a new neighbor has "neighbor rate" > "adaptation rate" then it will get reward and they will be categorized in the same group with the previous data. In other word, they make strong pattern. Otherwise, it will get penalty and lead to decrease its relationship between previous neighbors. When this happen, it will be sent to another group and lead to decrease its relationship between previous neighbors.

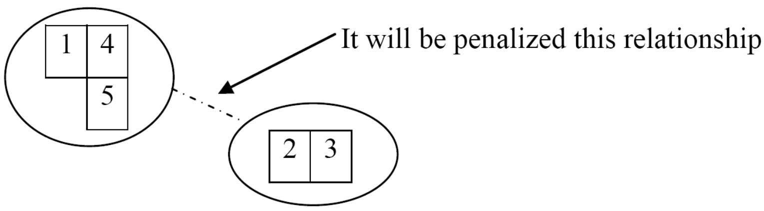


Figure 8. Get penalty between two groups of neighbors

The obtained Tables 4(a) and 4(b) show pattern found in dataset with amount of effective pattern shown in the last column. Reinforcement value demonstrates the combination power and prediction about the result and how much that number is bigger and more number of combination members will be straighter for prediction. Those tables are also known as a pattern model. In order to ensure its accuracy, the model will be tested against some test data. The testing process of the model is explained in the next stage. This section run base on follow algorithm.

3.3. **Third step: testing pattern model.** In this part, the test step or prediction of raw data will be demonstrated.

Example 3.3. *Test step of the propose model. Assume that given an unknown class data with the attributes value as in Table 5.*

Table 5. The sample of test data

Data number	Attribute #1	Attribute #2	Attribute #3	Attribute #4	Class
1	**1**	**0**	**1**	3	Unknown

The data have the following six combinations of pair wise attributes:

$$\mathbf{1.1} \rightarrow \mathbf{2.0}$$
$$\mathbf{1.1} \rightarrow \mathbf{3.1}$$
$$\mathbf{2.0} \rightarrow \mathbf{3.1}$$
$$1.1 \rightarrow 4.3$$
$$3.1 \rightarrow 4.3$$
$$\mathbf{2.0} \rightarrow \mathbf{4.3}$$

Extraction of pattern from Tables 4(a) and 4(b) will be done based on the above pair wise attributes. Hence, the pattern in Tables 6(a) and 6(b) are obtained.

Table 6. (a) Sequence pattern compilation for class 1, (b) sequence pattern compilation for class 0

(a)

Pair wise attributes	Effective rate	Reinforcement value
$1.1 \rightarrow 2.0$, $1.1 \rightarrow 3.1$, $2.0 \rightarrow 3.1$	2	$3{\times}2 = 6$

(b)

Pair wise attributes	Effective rate	Reinforcement value
$2.0 \rightarrow 4.3$	2	$1 \times 2 = 2$

In this case, the desired test data complies with pattern in Tables 6(a) and 6(b), with the reinforcement value 6 and 2. The higher reinforcement value means the higher is the compliance. If compliance rate is the same in both tables then the class that has maximum length of combined sequence pattern will be chosen. Hence, in this example the data is determined as class 1.

The algorithm for this CLA is presented in Figure 9.

4. **Experimental Result.**

In this section, the proposed method is tested and is compared with some other methods on some types of online dataset which were obtained from the UCI Machine Learning dataset [22]. We assess the model in terms of the speed of answering and accuracy. In this comparison, 10-fold cross validation method on the WEKA software for extracting the results of another model was used. In this work, cellular learning automata with the learning model of P and learning algorithm of L_{RP} is used and initialize p_i in each stage equal to $1/r$ where r is the number of pair wise attributes.

Systemic characteristics which are used for simulation include the CPU 2.0 GHz Core2-Duo with RAM 2GB, windows XP/SP2 operating system and visual basic.net express edition 2008 programming language and the following results is obtained.

Table 7 shows the experimental result of the proposed method and three well known methods that are MLPN, C4.5 and NB. The observation is on the training time and the accuracy of classification. It can be seen that out of 4 methods, the best training time is dominated by NB method, whilst CLA is in the second place and is better than MLPN and C4.5. For classification and other data mining technique accuracy is very important and all researchers try to improve the accuracy of the models but in suitable training

```
CLA-Data classification algorithm
Input: Item Є transaction dataset, G (V, E);
Output: integer Є class of the dataset
Begin
x_i : array of [1.. n]; i: integer;
    // n is number of dataset row and i is number of pair wise attributes
For each row in dataset do
   For i=1 to pair wise attributes do
      x_i [n] ← row number of merge(att#j,att#k);    ∀j,k  j≠k
   EndFor
EndFor
Construct an irregular CLA isomorphic;
For each x_i do
    // x_i is consist of all row number of pair wise attributes
    Each cell of CLA chooses one member of x_i;
    Calculate the neighbor rate;
    If neighbor rate > adaptation rate then
        Reward action has been chosen by automata A_i;
    Else
        Penalty action has been chosen by automata A_i;
    EndIf
        Update neighbors;
EndFor
      Combine the pair wise attributes of reinforcement neighbors
// end of training section
For each testing row in dataset do
    Compare each pair wise of testing row to Table 4(a) and 4(b);
    For each match patterns do
     Calculate match rate;
      //match rate= effect rate * number of pair wise attributes, according to Table 6(a) and
6(b)
    EndFor
EndFor
Return class number;
End
End. CLA-Data classification algorithm
```

FIGURE 9. Show the data classification algorithm using CLA

time. Training time very depends to programming language and programming technique, but accuracy depends to proposed algorithms. In this work according to the results, training time is very suitable and very near to other models whilst CLA dominated the best method for accuracy.

In this table, the efficiency of the proposed model can be observed and acceptable answer is presented in study and data classification.

5. **Discussion.** In order to evaluate the proposed model, several datasets with different number of attributes and data has been used. The model obtains all relations between all attributes and then relationships that are stronger and have been reinforced will be isolated. Wisconsin Breast Cancer dataset comprises of 10 attributes and 699 data, Solar_Flar comprises of 13 attributes and 323 data and Tic-Tac-Toe comprises of 10 attributes and 958 data. Due to the strong relationship between of the attributes, the classification accuracy of all models including the proposed model is high. However, in the Liver disorders dataset that comprises of 7 attributes and 345 data and Hoberman

TABLE 7. Experimental result of the proposed method

Dataset	Number of Attributes	Training time by MLPNN (second)	Training time by C4.5 (second)	Training time by NB (second)	Training time by CLA (second)	Accuracy by MLPNN (percentage)	Accuracy by C4.5 (percentage)	Accuracy by NB (percentage)	Accuracy by CLA (percentage)	Adaptation rate
Wisconsin breast cancer	10	2.3	0.09	0.02	0.03	96.28	94.57	95.99	99.86	100%
Liver disorders	7	0.88	0.03	0.03	0.03	71.59	68.70	55.36	69.04	75%
Tic-tac-toe	10	10.25	0.02	0.02	0.02	97.18	84.56	69.70	97.94	100%
Hoberman	4	0.63	0.01	0.01	0.01	72.55	72.88	76.14	80.49	75%
Solar_flar	13	6.94	0.01	0.01	0.01	96.91	97.94	90.72	97.84	75%

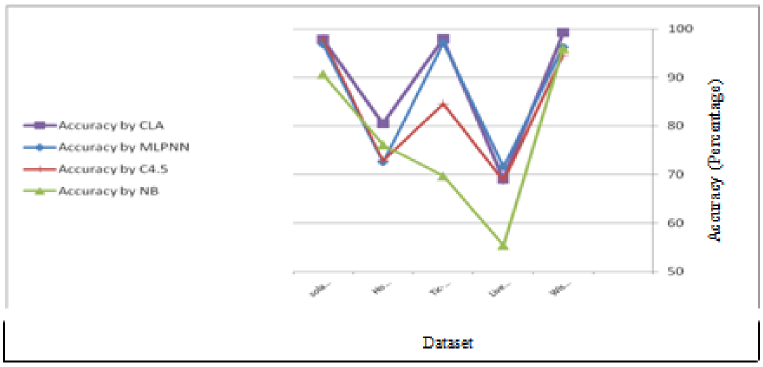


FIGURE 10. Experimental result of the proposed method

dataset that comprises of 4 attributes and 306 data, the accuracy of the all models is low with proposed model is better that another. This is due to the conflict in relationships between attributes.

Generally in the proposed method, if relations between attributes are stronger, then due to the relationship reinforcement, the accuracy will be high.

6. **Conclusion.** This section proposed a model based on cellular learning automata for data classification. The proposed model works based on the amount of relationship reinforcement which extract all relationships between attributes, then separate the stronger one and lastly make a decision according to the strength of the relationship. The result demonstrates that the proposed model can works on all of the kind of datasets and it is suitable and acceptable from the aspect of training time and excellent in accuracy. In term of algorithm, this method is very low in calculation complexity and the implementation is simple.

REFERENCES

[1] M. L. Tsetlin, *Automata Theory and Modeling of Biological Systems*, Academic Press, New York, 1973.
[2] K. Najim and A. S. Poznyak, *Learning Automata: Theory and Application*, Elsevier Science Publishing Ltd, Tarrytown, New York, 1994.
[3] A. G. Barto and R. Sutton, *Introduction to Reinforcement Learning*, MIT Press, 1997.
[4] R. L. de Mantaras and E. Armengol, Machine learning from examples: Inductive and lazy methods, *Data & Knowledge Engineering*, vol.25, pp.99-123, 1998.
[5] G. Batista and M. C. Monard, An analysis of four missing data treatment methods for supervised learning, *Applied Artificial Intelligence*, vol.17, pp.519-533, 2003.
[6] K. S. Narendra and M. A. L. Thathachar, *Learning Automata: An Introduction*, Prentice Hall, 1989.

[7] H. Peng, F. Long and C. Ding, Feature selection base on mutual information: Criteria of max dependency, max-relevance, and min redundancy, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.27, pp.1226-1238, 2005.
[8] S. B. Kotsiantis, I. D. Zaharakis and P. E. Pintelas, Machine learning: A review of classification and combining techniques, *Artificial Intelligence Review*, vol.26, no.3, pp.159-190, 2006.
[9] I. Guyon and A. Elisseeff, An introduction to variable and feature selection, *The Journal of Machine Learning Research*, vol.3, pp.1157-1182, 2003.
[10] S. K. Murthy, Automatic construction of decision trees from aata: A multi-disciplinary survey, *Data Mining and Knowledge Discovery*, vol.2, pp.345-389, 1998.
[11] J. R. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufmann, San Francisco, 1993.
[12] O. T. Yıldız and O. Dikmen, Parallel univariate decision trees, *Pattern Recognition Letters*, vol.28, no.7, pp.825-832, 2007.
[13] F. Rosenblatt, *Principles of Neuro Dynamics*, Spartan, New York, 1962.
[14] M. W. Gardner and S. R. Dorling, Artificial neural networks (the multilayer perceptron) – A review of applications in the atmospheric sciences, *Atmospheric Environ.*, vol.32, pp.2627-2636, 1998.
[15] Y. Freund and R. Schapire, Large margin classification using the perceptron algorithm, *Machine Learning*, vol.37, pp.277-296, 1999.
[16] M. Kayri and O. Cokluk, Data optimization with multilayer perceptron neural network and using new pattern in decision tree comparatively, *Journal of Computer Science*, vol.6, no.5, pp.606-612, 2010.
[17] S. Kotsiantis, D. Kanellopoulos and P. Pintelas, Data preprocessing for supervised leaning, *International Journal of Computer Science*, vol.1, no.2, pp.111-117, 2006.
[18] S. B. Kotsiantis, Supervised machine learning: A review of classification techniques, *Informatica*, vol.31, pp.249-268, 2007.
[19] K. S. Narendra and R. Viswanathan, Learning models using stochastic automata, *International Conference of Cybernetics and Society*, Washington DC, 1972.
[20] J. M. Valls, I. M. Galva and P. Isasi, Learning radial basis neural networks in a lazy way: A comparative study, *Neurocomputing*, vol.71, pp.2529-2537, 2008.
[21] K. S. Fu and G. J. McMurtry, A study of stochastic automata as models of adaptive and learning controllers, *IEEE Transactions on Automatic Control*, vol.11, pp.379-387, 1966.
[22] *UCI Repository of Machine Learning Databases*, http: //www.ics.uci.edu/mlearn/MLRepository.html, Department of Information and Computer Sciences, University of California, Irvine, USA.
[23] G. J. Mun, B. N. Noh and Y. M. Kim, Enhanced stochastic learning for feature selection in intrusion classification, *International Journal of Innovative Computing, Information and Control*, vol.5, no.11(A), pp.3625-3635, 2009.
[24] Y. Z. Tsypkin, *Adaptation and Learning in Automatic Systems*, Academic Press, New York, 1971.
[25] J. von Neumann and A. W. Burks, *Theory of Self-reproducing Automata*, University of Illinois Press, 1966.
[26] M. Esnaashari and M. R. Meybodi, Dynamic point coverage problem in wireless sensor networks: A cellular learning automata approach, *Journal of Ad Hoc and Sensors Wireless Networks*, vol.10, no.2-3, pp.193-234, 2010.