

Automatic Finding Trapezoidal Membership Functions in Mining Fuzzy Association Rules Based on Learning Automata

Z. Anari¹, A. Hatamlou², B. Anari³ *

¹ Department of Computer Engineering and Information Technology, Payam Noor University (PNU), Tehran (Iran)

² Department of Computer Engineering, Khoy Branch, Islamic Azad University, Khoy (Iran)

³ Department of Computer Engineering, Shabestar Branch, Islamic Azad University, Shabestar (Iran)

Received 4 January 2020 | Accepted 9 June 2021 | Early Access 17 January 2022



ABSTRACT

Association rule mining is an important data mining technique used for discovering relationships among all data items. Membership functions have a significant impact on the outcome of the mining association rules. An important challenge in fuzzy association rule mining is finding an appropriate membership functions, which is an optimization issue. In the most relevant studies of fuzzy association rule mining, only triangle membership functions are considered. This study, as the first attempt, used a team of continuous action-set learning automata (CALA) to find both the appropriate number and positions of trapezoidal membership functions (TMFs). The spreads and centers of the TMFs were taken into account as parameters for the research space and a new approach for the establishment of a CALA team to optimize these parameters was introduced. Additionally, to increase the convergence speed of the proposed approach and remove bad shapes of membership functions, a new heuristic approach has been proposed. Experiments on two real data sets showed that the proposed algorithm improves the efficiency of the extracted rules by finding optimized membership functions.

KEYWORDS

Continuous Action-set Learning Automata (CALA), Data Mining, Fuzzy Association Rules, Learning Automata, Trapezoidal Membership Function.

DOI: 10.9781/ijimai.2022.01.001

I. INTRODUCTION

By increasing the volume of data in the databases, effective techniques are required to manage the data in these databases. Data mining is the process of exploring great amounts of data from transactional databases to obtain interesting information [1]-[2]. Some important data mining techniques include clustering [3], classification [4], prediction [5], text mining [6], and association rules [7-10]. Association rule mining is used to produce meaningful relationships among data elements within the transaction databases [11, 12]. One association rule is described as $X \rightarrow Y$, where X and Y belong to itemsets and $X \cap Y = \emptyset$. This means that, if the set of items belongs to X , it most likely also belongs to Y [12]. The Apriori algorithm is an efficient algorithm in data mining [13], and uses statistical techniques to find association rules.

Fuzzy theory used in many fields, such as engineering applications, optimization algorithms, data mining, and intelligent systems [14]-[26]. Some research papers have proposed algorithms for extracting fuzzy association rules [7], [27]-[35]. However, these methods only consider triangular membership functions that are not suitable for

some applications. The methods using trapezoidal membership functions (TMFs) supposed that the shape of the trapezoidal membership functions for each linguistic term are well known. If so, some mined results may be inappropriate.

In this study, we addressed the above issue and proposed a novel algorithm that uses continuous action-set learning automata (CALA), named CALA-AFTM, to find position and the number of TMFs in fuzzy association rule mining at the same time. CALA is a mathematical approach that interacts with an environment and by using the environment response, generates a common reinforcement signal [36]. CALA has several advantages. Unlike other metaheuristic methods, to optimize a function, CALA only needs to build one team of LA which is equivalent to one chromosome in other evolutionary algorithms. Therefore, it requires fewer evaluation functions. It also needs fewer parameters to perform the optimization process, and can find the optimal local value. CALA applied in many applications and algorithms [37]-[42].

In this paper, finding positions and number of TMFs for each membership function has been regarded as parameters of the search space. To find these optimal parameters, a novel representation was suggested to build a CALA team. This team is divided into two parts. The learning automaton (LA) of the first part responsible for specifying the optimal number of membership functions, and the LA of the second part is used to optimize the positions of membership functions. Briefly, the main contributions of this research are as follows:

* Corresponding author.

E-mail addresses: zanari323@yahoo.com (Z.Anari), rezahatamlou@gmail.com (A.Hatamlou), anari@iaushab.ac.ir (B.Anari).

Please cite this article in press as:

Z. Anari, A. Hatamlou, B. Anari. Automatic Finding Trapezoidal Membership Functions in Mining Fuzzy Association Rules Based on Learning Automata, International Journal of Interactive Multimedia and Artificial Intelligence, (2022), <http://dx.doi.org/10.9781/ijimai.2022.01.001>

- The proposed approach dynamically determines the position and number of TMFs in mining fuzzy association rule.
- A new representation is developed to construct a CALA team.
- To reduce the domain of the search space and increase the speed of convergence, two constraints were proposed.
- A series of experiments on two real datasets conducted to represent the great effectiveness of the proposed approach.

To assess the results, the proposed CALA-AFTM algorithm compared with fuzzy web mining algorithm (FWMA) [31] and VSLA-AFTM algorithm. Two real datasets, CTI and NASA, were employed for the experiments. They were selected because they used the time that users spend on web pages, and since this parameter is a fuzzy variable, it can be considered as TMFs [43]–[47].

The rest of this document is organized as follows: In Section II, a review of related work is presented. Section III provides background information. Proposed CALA-AFTM algorithm is shown in Section IV. Section V provides the data set and experiment results. Finally, conclusion and future research is provided in Section VI.

II. RELATED WORK

Most studies on data mining, used the predefined membership functions to derive fuzzy association rules. [22] described the definition and confidence factor of fuzzy association rule and proposed a fuzzy mining method to derive fuzzy association rules in databases. [20] introduced an algorithm named F-APACS for finding association rules in fuzzy data mining. They used fuzzy linguistic terms to manage quantitative values. This had two advantages: First, the user-supplied threshold did not have to be determined; second, it could find both positive and negative association rules. [48] presented a fuzzy multiple-level approach to search meaningful fuzzy association rules from quantitative values in transaction datasets. In their approach, first each quantitative value was converted to a linguistic variable. Then, for each fuzzy variable the scalar cardinality was computed. Finally, the mining process was conducted to obtain fuzzy association rules.

Several fuzzy data mining algorithms were also suggested to extract both the proper set of membership functions. These algorithms make it possible to automatically adjust membership functions using meta-heuristic approaches to extract the best fuzzy rules. [49] utilized the genetic algorithm and presented a mining approach to find the proper set of membership functions and fuzzy rules. In their approach, each membership function was converted into a fixed-length string. Then, appropriate strings were selected to build an appropriate membership function set.

Similarly, [50] enhanced their earlier method [49] and proposed a fuzzy mining approach using genetic algorithms and k-means clustering algorithm to find the best membership functions and fuzzy rules. In their method, one chromosome was assigned to each membership function. At first, each membership functions set was converted into a fixed-length string. Then, using the k-means clustering algorithm, each chromosome was assigned in one of the clusters. Finally, by assessing the fitness value, the appropriate membership functions were determined.

Using the genetic algorithm another mining approach proposed in [51] to obtain type-2 membership functions. There was a 2-tuple linguistic layout schema that encoded chromosome membership functions. In their algorithm, they used a parameter called the diversity factor to find suitable rules. In addition, [52] utilized the genetic algorithm to obtain the best membership functions and fuzzy rules. In their algorithm, a 2-tuple linguistic approach was proposed to reduce domain search space and remove improper membership functions.

Furthermore, [53] used the genetic algorithm and proposed a fuzzy mining approach to solve the problem of intrusion detection. [54] proposed GA-based approach to find concept-drift patterns and optimal membership functions. They employed a 2-tuple demonstration method to code the membership functions in chromosomes. [55] proposed a mining method and shown a new chromosome representation to identify suitable membership functions. In their approach, each chromosome represented membership functions sets and contained two parts. The first part was shown with binary strings and specified the activation of each membership function. The second part specified the parameters corresponding to the active membership functions. They obtained the optimal membership functions by checking whether the membership functions were active or not. [56] suggested a two-step method to dynamically identify fuzzy rules and optimal membership functions. In step one, they used a GA-based approach and 3-tuple scheme to obtain the optimal membership functions. In step two, they used pattern-growth method to derive fuzzy rules.

Also, [57] suggested a clustering method to automatically tune membership functions and extract weighted fuzzy rules. In their approach, each chromosome encoded by real number values and each population using the fitness function value was evaluated to find the suitable membership functions. using ant colony approach [58] proposed an approach to derive the optimal membership functions. In their method, each membership function encoded by binary strings. First an initial graph was created, then the ants moved in the graph and formed the final membership functions. [59] suggested a mining approach based on improved ant colony method to find appropriate membership functions. They developed a code representation and found the real global optimum solution in a continuous space by introducing certain operators.

[34] proposed a genetic-based approach using master-slave parallel processing technology to find the optimal membership functions and fuzzy rules. The main processor distributes fitness tasks among the slave processors. The fitness function was evaluated by each slave processor, and then the result of each slave processor sends to the main processor. Subsequently, the main processor used all the values of the fitness function to determine the best membership function. [60] presented a memetic based algorithm to extract optimal membership functions. Their algorithm for representing chromosomes the structure type of the membership function is considered. They proposed an approach to eliminate inappropriate membership functions using the nature of the structure in their method, thereby reducing the search space. In addition, their approach used structure types and extend the local search approach to reduce domain search space and find optimal membership functions.

[33] presented a GA-based approach to find appropriate membership functions. In their method, each membership function was demonstrated by three parameters. Each parameter corresponded to one of the vertices of the triangle, and these parameters encoded as chromosomes. Then, an appropriate membership functions sets was derived. Also, two heuristics proposed to remove improper membership functions and reduce domain search space. Using the Levenberg-Marquardt method and bacterial memetic algorithm [61] proposed a memetic based approach to find best fuzzy rules. In their approach, each bacterium represented the parameters of the fuzzy rule. [62] presented a mining approach using bat algorithm to find optimal membership functions dynamically. Their algorithm considered more factors in fitness function and by improving the local and global search extracted more precise rules.

[63] proposed a temporal mining approach, which combines the bee method and the fuzzy temporal mining method. Their algorithm obtained suitable membership functions and extracted fuzzy rules. [64] used a particle swarm optimization approach and proposed a

framework to extract optimal membership functions. In addition to single-objective optimization, some algorithms have also utilized multi-objective optimization to optimize the membership functions and extract association rules. [65] proposed a multi-objective optimization method for a classifier system using genetic algorithm. They used two criteria to assess precision and interpretability. Their method determined both the structure and membership functions on the basis of fuzzy rules. Using a multi-objective genetic algorithm [66] proposed a fuzzy mining approach to find fuzzy rules. They used three criteria as the goals of multi-objective optimization methods, namely, confidence, comprehensibility, and interestingness. [67] proposed a multi-objective-based approach using genetic algorithm for mining fuzzy association rules. In their approach, minimum support and minimum confidence parameters were determined automatically. Also, the fitness function specified the position of chromosomes and did not affect the genetic operator. Therefore, their method converged with the value of any fitness function.

To derive fuzzy web browsing patterns or association rules, many mining approaches have been reported. In [43] a fuzzy mining approach proposed to obtain sequential web patterns from the log records. In their algorithm, the importance of each web page was considered as a fuzzy variable and the importance of each web page transformed into fuzzy values. Using object-oriented concepts and fuzzy sets [29] proposed a web mining approach. In their method, each web page and the visited that web page were represented as a class and instance of that class, respectively. In their method, in the first step, for each web page association rules were determined and in the second step the relationships among all instances were determined. [27] proposed a GA-based approach to mine temporal association rules in web datasets. They showed that converting the data set to a graph extracted more reliable rules.

[68]-[69] introduced a generalized fuzzy web mining algorithm for extracting interesting association rules. Their approach used a fixed number of membership functions to convert quantitative web data to fuzzy values. [28] employed fuzzy concepts and proposed an algorithm to extract fuzzy rules from web usage data. In their method, the time duration of each web page visited by users were considered as a linguistic variable. [69] used a CALA algorithm and developed an algorithm to identify optimal membership functions which used in trust and distrust fuzzy recommender systems. Their method, only adjusted the center parameter of membership functions.

III. PRELIMINARIES

This section describes the concepts associated with fuzzy association rules and CALA.

A. Fuzzy Association Rule

A fuzzy association rule is described as follows: Let $I = \{i_1, i_2, \dots, i_m\}$ represent a set of items, and let $T = \{t_1, t_2, \dots, t_n\}$ represent a set of transactions, and each transaction contains items from a set of items I . Thus, each transaction t_i includes a subset of the items in I where $T \subseteq I$. An association rule is represented as $x \rightarrow y$, where $x \subseteq y$, $y \subseteq I$ and $x \cap y = \emptyset$, x and y are a set of items known as itemsets. Intuitively, the $x \rightarrow y$ rule means that transactions that contain x tend to contain y . For example, in market basket analysis, the association rule $\{\text{bread, milk}\} \rightarrow \{\text{butter}\}$ tells those customers have purchased bread and milk and are then likely to purchase butter. The goal association rule mining is to obtain rules in the transaction data set such that the support and confidence of each rule are greater than or equal to the minimum support and minimum confidence. The support value for the rule $x \rightarrow y$ is described as follows:

$$\text{Support}(x \rightarrow y) = \frac{|x \cup y|}{|I|} \quad (1)$$

The confidence value for the rule $x \rightarrow y$ is described as follows:

$$\text{Confidence}(x \rightarrow y) = \frac{|x \cup y|}{|x|} \quad (2)$$

The fuzzy support value for region $R_{j,k}$ is described as follows:

$$\text{Fuzzy Support}(R_{j,k}) = \frac{\sum_{i=1}^n f_{j,k}^{(i)}}{|I|} \quad (3)$$

where $R_{j,k}$ represents the fuzzy region for the k -th membership function of an item I_j , $f_{j,k}^{(i)}$ demonstrates the fuzzy membership function value of region $R_{j,k}$ in the i -th transaction.

B. Continuous Action-Set Learning-Automata (CALA)

LA is a mathematical tool for solving optimization problems. Each automaton selects an action from its set of actions and applies it to the environment. Then, LA uses the response received from the environment and decides whether the selected action is rewarded or penalized [38],[70]. LA has been used in many fields such as computer networks [71], image processing [42], speech analysis [72], signal processing [73], and clustering [74]. The interaction between the LA and the environment is shown in Fig. 1.

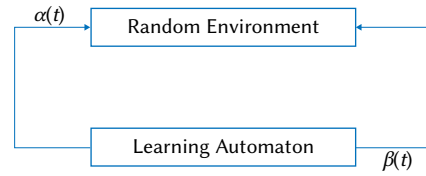


Fig. 1. The interaction between the LA and the environment.

The environment is determined by a triple $\langle \alpha, \beta, c \rangle$, where $\alpha = \{\alpha_1, \alpha_2, \dots, \alpha_s\}$ shows the input action-set, $\beta = \{\beta_1, \beta_2, \dots, \beta_m\}$ denotes the environment's response, and $c = \{c_1, c_2, \dots, c_r\}$ shows the penalty probabilities.

CALA [36] is an optimization tool which is used to minimize a multivariate function. To minimize the multivariate function $f: R^m \rightarrow R$, a team with m LA is required. Each LA $LA^i (1 \leq i \leq m)$ in this team at instant n uses two internal parameters, mean $\mu^i(n)$ and variance $\sigma^i(n)$. Each LA^i chooses an action $\alpha^i(n) \in N(\mu^i(n), \sigma^i(n))$ where $\alpha^i(n)$ is a normal distribution with the mean $\mu^i(n)$ and standard deviation $\sigma^i(n)$. Using the generated actions and the mean value of each LA, two inputs $\alpha(n) = (\alpha^1(n), \alpha^2(n), \dots, \alpha^m(n) \in R^m)$ and $\mu(n) = (\mu^1(n), \mu^2(n), \dots, \mu^m(n) \in R^m)$ for computing noisy functions $\beta_\alpha(n)$ and $\beta_\mu(n)$ are described as $\beta_\alpha(n) = f(\alpha(n))$ and $\beta_\mu(n) = f(\mu(n))$. Then, each LA^i updates $\mu^i(n)$ and $\sigma^i(n)$ according (4) to (8), respectively. By repeating the learning process, each learning automaton finds its optimal action and the final value for β_μ is considered as the minimum value of function f or cost function [36]. The CALA algorithm is represented in Algorithm 1, where $C > 0$ shows the penalty parameter; and $0 < \lambda < 1$ shows the learning parameter and $\delta_l \in R$ represents the lower bound for the variance parameter.

$$\mu^i(n+1) = \mu^i(n) - \lambda F_1^i(\mu(n), \sigma(n), \alpha(n), \beta_\alpha(n), \beta_\mu(n)) \quad (4)$$

$$\sigma^i(n+1) = \lambda F_2^i(\mu(n), \sigma(n), \alpha(n), \beta_\alpha(n), \beta_\mu(n)) + C[\delta_l - \sigma^i(n)] \quad (5)$$

$$F_1^i(\mu(n), \sigma(n), \alpha(n), \beta_\alpha(n), \beta_\mu(n)) = \left(\frac{\beta_\alpha(n) - \beta_\mu(n)}{\sigma[\sigma(n)]} \right) \left(\frac{\alpha^i(n) - \mu^i(n)}{\sigma[\sigma^i(n)]} \right) \quad (6)$$

$$F_2^i(\mu(n), \sigma(n), \alpha(n), \beta_\alpha(n), \beta_\mu(n)) = \left(\frac{\beta_\alpha(n) - \beta_\mu(n)}{\sigma[\sigma(n)]} \right) [(\frac{\alpha^i(n) - \mu^i(n)}{\sigma[\sigma^i(n)]})^2 - 1] \quad (7)$$

$$\sigma[\delta] = \begin{cases} \delta_l & \text{for } \delta \leq \delta_l \\ \delta & \text{for } \delta > \delta_l \end{cases}, \text{ For } \delta, \delta_l \in R \quad (8)$$

The pseudo-code of the CALA algorithm is shown below.

Algorithm 1: CALA

Input: m // m is the number of learning automata in CALA.
Output: minimum value of f // $f: R^m \rightarrow R$ is cost function.
 // Initialize
 1 Initialize the value of C , λ and σ_L /* C is a penalizing constant, λ is a learning rate and CTL is a lower bound. */
 2 $n \leftarrow 0$ // n is time step.
 3 **foreach** $LA^i, 1 \leq i \leq m$ **do**
 4 Initialize the mean and variance of the Gaussian distribution as $\mu^i(n)$ and $\sigma^i(n)$
 5 **end foreach**
 6 **repeat**
 7 **foreach** $LA^i, 1 \leq i \leq m$ **do**
 8 Choose an action $\alpha^i(n) \in R$ using Gaussian distribution $N(\mu^i(n), \sigma^i(n))$. /* $\mu^i(n) \in R$ is mean and $\sigma^i(n) \in R$ is the standard deviation. */
 9 **end foreach**
 // using the generating actions build $\alpha(n)$
 10 $\alpha^i(n) = (\alpha^1(n), \alpha^2(n), \dots, \alpha^m(n)) \in R^m$
 // using the the mean value of each LA build $\mu(n)$
 11 $\mu(n) = (\mu^1(n), \mu^2(n), \dots, \mu^m(n)) \in R^m$
 // compute two noisy functions $\beta_\alpha(n)$ and $\beta_\mu(n)$
 12 $\beta_\alpha(n) \leftarrow f(\alpha(n))$
 13 $\beta_\mu(n) \leftarrow f(\mu(n))$
 // update parameters of each LA
 14 **foreach** $LA^i, 1 \leq i \leq m$ **do**
 15 update $\mu^i(n)$ and $\sigma^i(n)$ according to Eqs. (4) to (8).
 16 **end foreach**
 // increase time step.
 17 $n \leftarrow n + 1$
 18 **until** for all LA, $\mu^i(n)$ does not change noticeably and $\sigma^i(n)$ converges to σ_i ;
 19 minimum value of $f \leftarrow \beta_\mu(n)$

IV. THE PROPOSED CALA-AFTM ALGORITHM

In this section, the cost function and the problem formulation of the proposed algorithm (CALA-AFTM) are defined.

A. Cost Function

To assess the quality of trapezoidal membership functions (TMFs), we modified the cost function proposed in [7] as follows:

$$\text{Cost function} = \frac{1}{\text{Objective function}} \quad (9)$$

$$\text{Objective function} = \frac{\sum_{X \in L_1} \text{fuzzy support}(R)}{\text{Suitability}} \quad (10)$$

where fuzzy support(R) shows the large 1-itemset R in L_1 . Moreover, L_1 shows the large 1-itemset. The fuzzy support(R) is computed according to (3). The suitability parameter is defined as [7]:

$$\text{Suitability} = \text{Overlap factor} + \text{Coverage factor} \quad (11)$$

The overlap factor for two regions R_i and R_j ($i < j$) is defined as the region covered by two regions R_i and R_j to the minimum of $(wr_{i,4} - c_i)$ and $(c_j - wl_{j,1})$. The coverage factor is defined as [7]:

$$\text{Coverage factor} = \frac{\max(I_i)}{\text{range}(R_1, R_2, \dots, R_m)} \quad (12)$$

$\text{range}(R_1, R_2, \dots, R_m)$ denotes the length of the horizontal axis, $\max(I_i)$ represents the maximum quantity of item I_i , and m represents the number of membership functions of item I_i . We changed the overlap factor shown in [7] to TMFs as follows:

$$\text{Overlap_factor} = \sum_{i < j} \left(\max \left(\frac{\text{Overlap}(R_i, R_j)}{\min((wr_{i,4} - c_i), (c_j - wl_{j,1}))}, 1 \right) - 1 \right) \quad (13)$$

Let MF_i , ($1 \leq i \leq m$) denotes fuzzy regions, where $MF_i = \{R_1, \dots, R_j\}$. Each R_i shows a trapezoidal membership function. Each R_i is defined as $(wl_{i,1}, cl_{i,2}, cr_{i,3}, wr_{i,4})$, where $wl_{i,1}$ is the left spread, $cl_{i,2}$ is the left center, $cr_{i,3}$ is the right center, and $wr_{i,4}$ represents the right spread of the fuzzy region R_i . c_i shows the middle center and described in (14). In Fig. 2 the TMFs parameters are shown. An Example: Suppose that membership functions for item I_j are presented in Fig. 3.

$$c_i = \frac{cl_{i,2} + cr_{i,3}}{2} \quad (14)$$

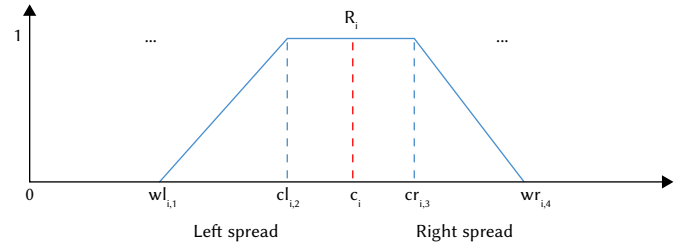


Fig. 2. Parameters of trapezoidal membership functions.

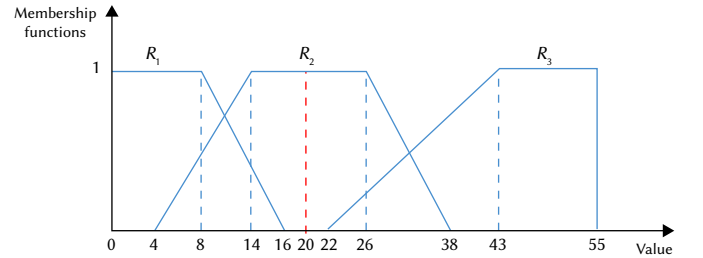


Fig. 3. Membership functions for item I_j .

In Fig. 3, item I_j has three membership functions: R_1 , R_2 and R_3 . It can be observed from Fig. 3 that the overlap (R_1, R_2) = 12, the overlap (R_2, R_3) = 16, the overlap (R_1, R_3) = 0.

the minimum spread (R_1, R_2) = $\min((16-0), (20-4)) = 16$, the minimum spread (R_2, R_3) = $\min((38-20), (55-22)) = 18$, and the minimum spread (R_1, R_3) = $\min((16-0), (55-22)) = 16$. So, the overlap factor of item I_j is computed as follows: overlap factor = $(\max(\frac{12}{16}, 1) - 1) + \max(\frac{16}{18}, 1) - 1 + \max(\frac{0}{16}, 1) - 1 = 0 + 0 + 0 = 0$. In this example, the range (R_1, R_2, R_3) = 55. Assume that the maximum number of items I_j in a transaction is 55, $\max(I_j) = 55$; then, the coverage factor of item I_j is computed as: coverage factor = $\frac{55}{55} = 1$. Therefore, the suitability value for the item I_j is determined as follows: suitability = $0 + 1 = 1$. Suppose that the number of transactions in the data set is six, and item I_j exists in five transaction data set with quantity 5, 12, 24, 30, and 46. Consequently, the fuzzy support (R_1, R_2, R_3) is computed as follows:

$$\begin{aligned} \text{Fuzzy Support}(\alpha_1) &= \frac{1}{6} \times \left(1 + \frac{16-5}{16-8} + 0 + 0 + 0 \right) = 0.39 \\ \text{Fuzzy Support}(\alpha_2) &= \frac{1}{6} \times \left(\frac{5-4}{14-4} + \frac{12-4}{14-4} + 1 + \frac{38-30}{38-26} + 0 \right) = 0.42 \\ \text{Fuzzy Support}(\alpha_3) &= \frac{1}{6} \times \left(0 + 0 + \frac{24-22}{43-22} + \frac{30-22}{43-22} + 1 \right) = 0.24 \end{aligned}$$

By considering the value 0.31 for the minimum support, the large 1-itemsets is specified as $L_1 = \{\alpha_1, \alpha_2\}$. So, the cost function computed as $\frac{1}{0.39+0.42} = 1.23$.

B. Appropriate Trapezoidal Membership Functions

Each trapezoidal membership function must have two conditions, which are shown in (15) and (16), respectively.

$$cr_{i,3} \leq cl_{i+1,2} \leq cr_{i+1,3} \leq \dots \leq cl_{m-1,2} \leq cr_{m-1,3} \leq cl_{m,2} \quad (15)$$

$$wl_{i,1} \leq cl_{i,2} \leq cr_{i,3} \leq wr_{i,4} \quad (16)$$

TABLE I. THE NUMBER OF SUITABLE MEMBERSHIP FUNCTIONS FOR THE THREE TRAPEZOIDAL MEMBERSHIP FUNCTIONS

$V_1 = (CR_{1,3}, WL_{2,1}, WR_{1,4}, CL_{2,2}, CR_{2,3}, WL_{3,1}, WR_{2,4}, CL_{3,3})$	$V_{13} = (WL_{2,1}, CR_{1,3}, WR_{1,4}, CL_{2,2}, CR_{2,3}, WL_{3,1}, WR_{2,4}, CL_{3,3})$
$V_2 = (CR_{1,3}, WL_{2,1}, WR_{1,4}, CL_{2,2}, CR_{2,3}, WL_{3,1}, CL_{3,3}, WR_{2,4})$	$V_{14} = (WL_{2,1}, CR_{1,3}, WR_{1,4}, CL_{2,2}, CR_{2,3}, WL_{3,1}, CL_{3,3}, WR_{2,4})$
$V_3 = (CR_{1,3}, WL_{2,1}, WR_{1,4}, CL_{2,2}, WL_{3,1}, CR_{2,3}, WR_{1,4}, CL_{3,3})$	$V_{15} = (WL_{2,1}, CR_{1,3}, WR_{1,4}, CL_{2,2}, WL_{3,1}, CR_{2,3}, WR_{2,4}, CL_{3,3})$
$V_4 = (CR_{1,3}, WL_{2,1}, WR_{1,4}, CL_{2,2}, WL_{3,1}, CR_{2,3}, CL_{3,3}, WR_{2,4})$	$V_{16} = (WL_{2,1}, CR_{1,3}, WR_{1,4}, CL_{2,2}, WL_{3,1}, CR_{2,3}, CL_{3,3}, WR_{2,4})$
$V_5 = (CR_{1,3}, WL_{2,1}, WR_{1,4}, WL_{3,1}, CL_{2,2}, CR_{2,3}, WR_{2,4}, CL_{3,3})$	$V_{17} = (WL_{2,1}, CR_{1,3}, WR_{1,4}, WL_{3,1}, CL_{2,2}, CR_{2,3}, WR_{2,4}, CL_{3,3})$
$V_6 = (CR_{1,3}, WL_{2,1}, WR_{1,4}, WL_{3,1}, CL_{2,2}, CR_{2,3}, CL_{3,3}, WR_{2,4})$	$V_{18} = (WL_{2,1}, CR_{1,3}, WR_{1,4}, WL_{3,1}, CL_{2,2}, CR_{2,3}, CL_{3,3}, WR_{2,4})$
$V_7 = (CR_{1,3}, WL_{2,1}, CL_{2,2}, WR_{1,4}, CR_{2,3}, WL_{3,1}, WR_{2,4}, CL_{3,3})$	$V_{19} = (WL_{2,1}, CR_{1,3}, CL_{2,2}, WR_{1,4}, CR_{2,3}, WL_{3,1}, WR_{2,4}, CL_{3,3})$
$V_8 = (CR_{1,3}, WL_{2,1}, CL_{2,2}, WR_{1,4}, CR_{2,3}, WL_{3,1}, CL_{3,3}, WR_{2,4})$	$V_{20} = (WL_{2,1}, CR_{1,3}, CL_{2,2}, WR_{1,4}, CR_{2,3}, WL_{3,1}, CL_{3,3}, WR_{2,4})$
$V_9 = (CR_{1,3}, WL_{2,1}, CL_{2,2}, WR_{1,4}, WL_{3,1}, CR_{2,3}, WR_{2,4}, CL_{3,3})$	$V_{21} = (WL_{2,1}, CR_{1,3}, CL_{2,2}, WR_{1,4}, WL_{3,1}, CR_{2,3}, WR_{2,4}, CL_{3,3})$
$V_{10} = (CR_{1,3}, WL_{2,1}, CL_{2,2}, WR_{1,4}, WL_{3,1}, CR_{2,3}, CL_{3,3}, WR_{2,4})$	$V_{22} = (WL_{2,1}, CR_{1,3}, CL_{2,2}, WR_{1,4}, WL_{3,1}, CR_{2,3}, CL_{3,3}, WR_{2,4})$
$V_{11} = (CR_{1,3}, WL_{2,1}, CL_{2,2}, CR_{2,3}, WR_{1,4}, WL_{3,1}, WR_{2,4}, CL_{3,3})$	$V_{23} = (WL_{2,1}, CR_{1,3}, CL_{2,2}, CR_{2,3}, WR_{1,4}, WL_{3,1}, WR_{2,4}, CL_{3,3})$
$V_{12} = (CR_{1,3}, WL_{2,1}, CL_{2,2}, CR_{2,3}, WR_{1,4}, WL_{3,1}, CL_{3,3}, WR_{2,4})$	$V_{24} = (WL_{2,1}, CR_{1,3}, CL_{2,2}, CR_{2,3}, WR_{1,4}, WL_{3,1}, CL_{3,3}, WR_{2,4})$

The first condition maintains the order of the trapezoidal centers, and the second condition preserves the trapezoidal shape. Fig. 4a and Fig. 4b shows two examples of unsuitable membership functions.

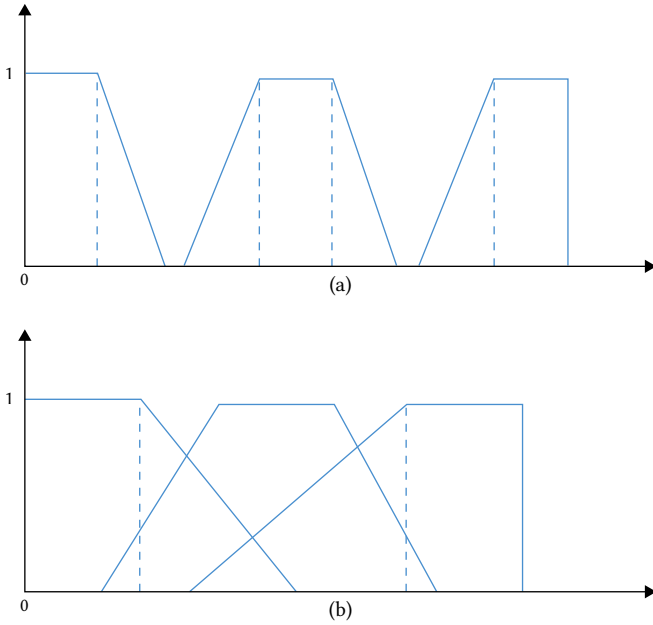


Fig. 4. Two sets of unsuitable membership functions.

By considering (15) and (16) the search space is greatly reduced. Additionally, (17) and (18) satisfy the complete coverage, and (19) satisfies the appropriate overlap.

$$wl_{i,1} \leq wl_{i,1} \leq cwr_{i-1,4} \quad (17)$$

$$wl_{i+1,1} \leq wr_{i,4} \leq wr_{i+1,4} \quad (18)$$

$$wl_{i,4} \leq wl_{i+2,1} \quad (19)$$

To reduce the domain of the search space and eliminate improper membership functions a new approach was proposed. The proposed approach is depicted below: Each $R_i (1 \leq i \leq m)$ is illustrated by quadruple $(wl_{i,1}, cl_{i,2}, cr_{i,3}, wr_{i,4})$. Let $P = \{cr_{1,3}, wl_{2,1}, wr_{1,4}, \dots, wl_{i,1}, wr_{i-1,4}, cl_{i,2}, cr_{i,3}, wr_{i,4}, \dots, wl_{m-1,1}, cl_{m-1,2}, cr_{m-1,3}, wr_{m-1,4}, cl_{m,2}\}$ be the set of all TMFs parameters.

Various permutations can be considered from this set. However, many of these permutations will be invalid because they do not maintain the trapezoidal shape. For each R_i , any non-duplicate Cartesian product is determined as a suitable membership function which is shown in (20).

$$\{wr_{i-1,4}, cl_{i,2}\} \times \{wr_{i-1,4}, cl_{i,2}, cr_{i,3}, wl_{i+1,1}\} \times \{wr_{i-1,4}, cl_{i,2}, cr_{i,3}, wl_{i+1,1}\} \times \{cr_{i,3}, wl_{i+1,1}\} \quad (20)$$

By considering the Cartesian product $R_1 \times R_2 \times \dots \times R_m$ and removing non-repetitive sequences, the final suitable trapezoidal membership functions are determined. For example, for three TMFs, we have $12! = 479,001,600$ sequences. While the proposed method produces only 24 membership functions ($L_3 = \{v_1, v_2, \dots, v_{24}\}$) as valid membership functions. Where each v_i shows the points of trapezoidal membership functions. In Table I all 24 valid membership functions are shown.

C. Representation of the Learning Automata

In Fig. 5, the representation of learning automata for the team of CALA is given. Let k_{\max} be the maximum number of fuzzy regions which is determined by the user. To build a team of CALA, we require $k_{\max} + \sum_{j=1}^{k_{\max}} k_{\max}$ number of LA. The first k_{\max} LA is used to specify the active TMFs. These LA are labeled as $LA^1, LA^2, \dots, LA^{k_{\max}}$. The actions generated by each LA of $LA^i (1 \leq i \leq k_{\max})$ are considered as $\alpha(n) \in N(\mu(n), \phi[\delta(n)])$, where $\alpha(n)$ is a normal distribution with mean $\mu(n)$ and standard deviation $\phi[\delta(n)]$. Let $MF_i (1 \leq i \leq k_{\max})$ consist of i fuzzy regions (i linguistic terms), where $MF_i = \{R_1, \dots, R_i\}$. Each MF_i includes i fuzzy regions, and each fuzzy region in TMFs is determined by four parameters. Consequently, to represent each $MF_i, i \times 4$ learning automata are required. Each fuzzy region $R_j (1 \leq j \leq i)$ of MF_i is equipped with four LA which are labeled as $LA_{i,j}^{k_j} (1 \leq i \leq k_{\max}, 1 \leq j \leq i \text{ and } k_j \in \{wl, cl, cr, wr\})$. The corresponding actions for $LA_{i,j}^{k_j}$ are shown by $\alpha_{i,j}^{k_j}(n) \in N[\mu_{i,j}^{k_j}(n), \phi_{i,j}^{k_j}(n)]$.

D. Defining the Action-Set and Generating Actions

In the proposed method, to increase the speed of convergence using the maximum value of the dataset, all values are normalized between 0 and 1. The action-set of $LA^i (1 \leq i \leq k_{\max})$, is considered as a $\alpha(n) \in N(0,1)$. For each automaton $LA_{i,j}^{k_j}$ the action-set is defined as $\alpha_{i,j}^{k_j}(n) \in N[0,1]$. Thus, to generate the actions at instant n , all LA of CALA, namely LA^i and $LA_{i,j}^{k_j}$, choose an action $\alpha(n) \in N[0,1]$ and $\alpha_{i,j}^{k_j}(n) \in N[0,1]$, respectively. The actions generated at instant n are denoted by (21). Also, using the generated $\alpha(n)$ and the mean value of each LA, $\mu(n)$ is determined as (22).

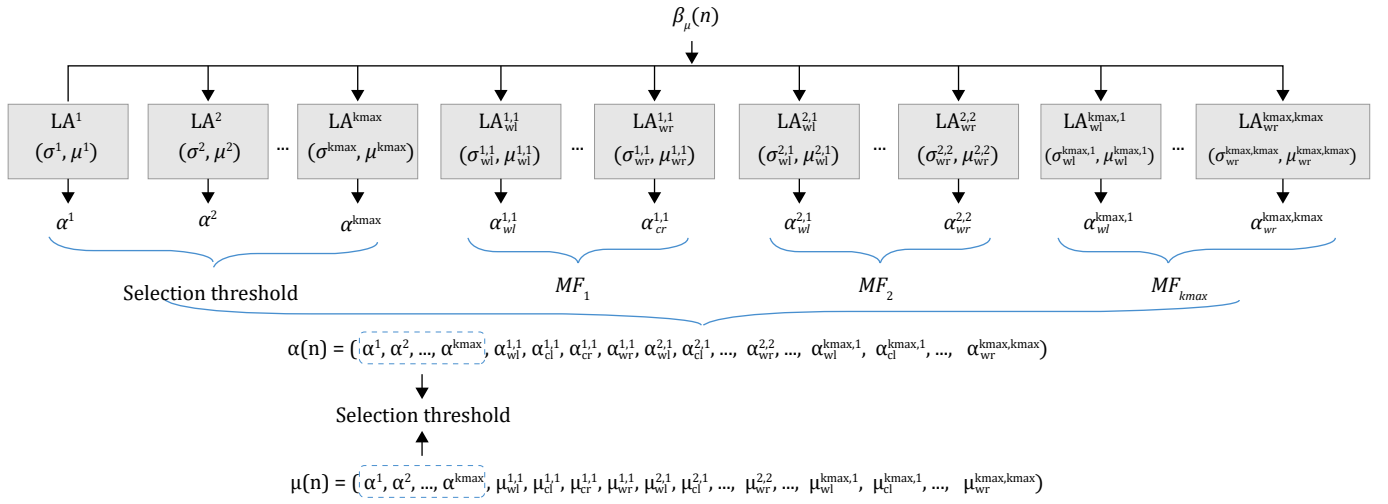


Fig. 5. Representation of CALA in the proposed algorithm.

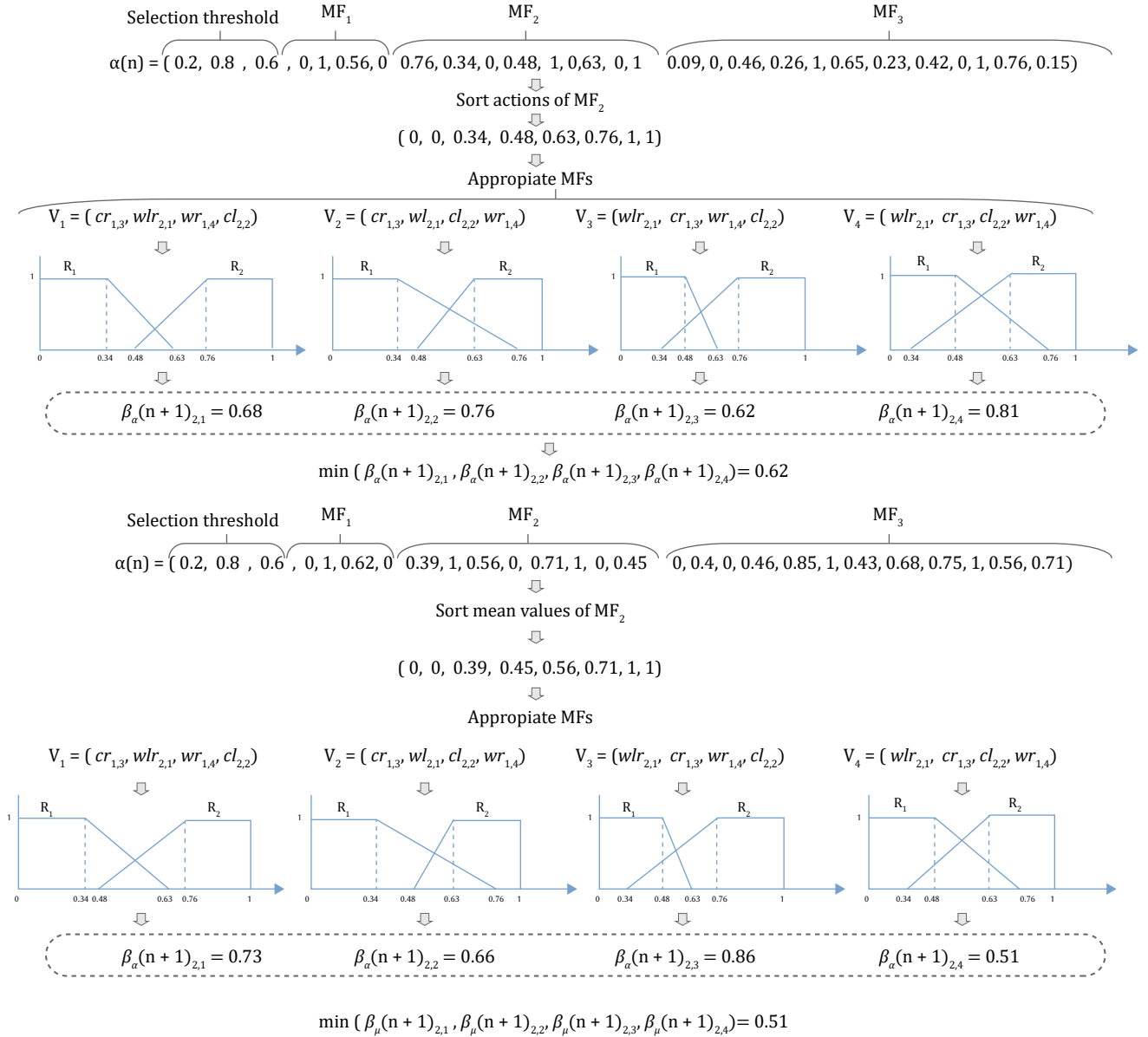


Fig. 6. An example of finding noisy function values in the proposed algorithm.

$$\alpha(n) = \begin{pmatrix} \alpha^1, \alpha^2, \dots, \alpha^{kmax}, \\ \alpha_{wl}^{1,1}, \alpha_{cl}^{1,1}, \alpha_{cr}^{1,1}, \alpha_{wr}^{1,1}, \\ \dots, \alpha_{wl}^{kmax,1}, \alpha_{cl}^{kmax,1}, \alpha_{cr}^{kmax,1}, \\ \alpha_{wr}^{kmax,1}, \dots, \alpha_{wl}^{kmax,kmax}, \alpha_{cl}^{kmax,kmax}, \\ \alpha_{cr}^{kmax,kmax}, \alpha_{wr}^{kmax,kmax} \end{pmatrix} \quad (21)$$

$$\mu(n) = \begin{pmatrix} \alpha^1, \alpha^2, \dots, \alpha^{kmax}, \\ \mu_{wl}^{1,1}, \mu_{cl}^{1,1}, \mu_{cr}^{1,1}, \mu_{wr}^{1,1}, \\ \dots, \mu_{wl}^{kmax,1}, \mu_{cl}^{kmax,1}, \mu_{cr}^{kmax,1}, \\ \mu_{wr}^{kmax,1}, \dots, \mu_{wl}^{kmax,kmax}, \mu_{cl}^{kmax,kmax}, \\ \mu_{cr}^{kmax,kmax}, \mu_{wr}^{kmax,kmax} \end{pmatrix} \quad (22)$$

E. Modification of the Generating Actions

After generating the actions, if the value of all action $\alpha^i(n)$ and $\alpha_k^{ij}(n)$ chosen by each automaton is less than 0 or greater than 1, reinitialize it with a value between 0 and 1. Also, if the values of all chosen actions of $\alpha^i(n)$ are less than 0.5, we randomly choose an action $\alpha^i(n)$ and reinitialize it with a value between 0.5 and 1.

F. Finding the Active TMFs

Using the generated actions for the first kmax automaton, the active TMFs at instant n are determined. If i is the number of actions with values greater than 0.5, then the membership functions MF_i and the corresponding actions, $\alpha_{wl}^{i,1}, \alpha_{cl}^{i,1}, \alpha_{cr}^{i,1}, \alpha_{wr}^{i,1}, \dots, \alpha_{wl}^{i,j}, \alpha_{cl}^{i,j}, \alpha_{cr}^{i,j}, \alpha_{wr}^{i,j}$ are activated.

G. Evaluation of the Noisy Functions

After computing $\alpha(n)$ and $\mu(n)$, the next step is to evaluate the values of the noisy function to find the values of $\beta_\alpha(n)$ and $\beta_\mu(n)$. CALA at instant n needs to compute two functions, namely, $\beta_\alpha(n) = f(\alpha(n))$ and $\beta_\mu(n) = f(\mu(n))$. To compute the value of $\beta_\alpha(n)$, perform the following steps:

1. Find the active TMFs (see Section F).
2. For active TMFs such as MF_i , the corresponding actions, $\alpha_{wl}^{i,1}, \alpha_{cl}^{i,1}, \alpha_{cr}^{i,1}, \alpha_{wr}^{i,1}, \dots, \alpha_{wl}^{i,j}, \alpha_{cl}^{i,j}, \alpha_{cr}^{i,j}, \alpha_{wr}^{i,j}$ are assigned in a vector, and their value is sorted in the ascending order.
3. Using the appropriate list for each MF_i and considering the sorted actions, TMFs for each element of valid membership functions are constructed.
4. For each constructed trapezoidal membership function, the value of $\beta_\alpha(n)$ is determined using the cost function.
5. For active membership functions such as MF_i , the minimum value of $\beta_\alpha(n)$ from among all appropriate membership functions is considered as the best solution at instant n.

$\beta_\mu(n)$ is calculated similarly to $\beta_\alpha(n)$, with the difference that $\mu(n)$ is used here.

H. Updating the Parameters of Each Learning Automaton

After determining the value of $\beta_\alpha(n)$ and $\beta_\mu(n)$, each LA of CALA updates its parameters, such as mean $\mu(n)$ and variance $\sigma(n)$. The updating rule for the LA of CALA is as follows: Each LA^i , ($1 \leq i \leq k_{max}$) updates the $\mu(n)$ and $\sigma(n)$. In this case, if $\mu^i(n)$ less than 0 or greater than 1, then, then its value is reinitialized with 0 and 1, respectively. In addition, each automaton of LA_k^{ij} updates its internal permeates of $\mu(n)$ and $\sigma(n)$ using (4) and (5). In this case, the reinitialization of μ_k^{ij} is not performed.

Fig. 6 shows an example of computing $\beta_\alpha(n)$ and $\beta_\mu(n)$. Assume that, at instant n, the action values for the two vectors $\alpha(n)$ and $\mu(n)$ are generated according to (4) and (5) which is a normal distribution with the mean value of 0 and standard deviation of 1. In this example, k_{max} is set at 3, so the team of CALA uses 27 learning automata. According to the number of actions for threshold values > 0.5 , it is

clear that MF2 can be active. According to the action values generated for MF2, we sort their values and then the TMFs are constructed for each vector by considering the appropriate membership functions that are specified by the four vectors v1 to v4. Subsequently, the value of each vector is evaluated using the cost function. In this example, $\beta_\alpha(n)_{ij}$ is used to represent the j-th appropriate membership function for MF_i . By considering the minimum value obtained among four membership functions, the value 0.62 will be selected as $\beta_\alpha(n)$. The procedure of computing $\beta_\mu(n)$ is similar to $\beta_\alpha(n)$, except that vector $\beta_\mu(n)$ is used instead of $\beta_\alpha(n)$. In this case, the value of 0.51 computed for $\beta_\mu(n)$. It is noteworthy that the value of $\beta_\mu(n)$ will be utilized as the reinforcement signal.

I. Pseudo-Code for the CALA-AFTM

In Algorithm 2, the pseudo-code of CALA-AFTM algorithm is presented. The CALA-AFTM algorithm uses four functions, as presented in Algorithms 3 to 6.

Algorithm 2: CALA-AFTM

Input: K_{max} // K_{max} is the maximum number of fuzzy regions.
Output: Number and positions of optimized trapezoidal membership functions.

```

// Save appropriate membership functions.
1 for  $i \leftarrow 1$  to  $K_{max}$  do
2    $L_i = \text{ValidMembershipFunctions}(i)$ 
3 end for
// Build CALA and initialize parameters.
4 Build a team of CALA with the size of  $K_{max} + \sum_{i=1}^{K_{max}} (i \times K_{max})$  LA.
5 For each LA of CALA, initialize the parameters of C,  $\sigma$ ,  $\lambda$ ,  $\sigma_L$  and  $\mu$ .
6  $n \leftarrow 0$ , ActiveMF  $\leftarrow 0$ , TempAction[]
7 repeat
  // steps for computing  $\beta_\alpha(n)$ 
  // Generate actions of  $\alpha(n)$ .
8 All LA of CALA, namely  $LA^i$  and  $LA_k^{i,j}$ , ( $1 \leq i \leq K_{max}$ ,  $1 \leq j \leq i$  and  $k \in \{w, c, r, w\}$ ) choose an action  $\alpha^i(n) \in N(0, 1)$  and  $\alpha_k^{i,j}(n) \in N(0, 1)$  respectively.
  // Check all selected actions are in the range of  $[0, 1]$ 
9 Modify Actions()
  // Count number of active membership functions.
10 ActiveMF  $\leftarrow \text{FindActiveMF}()$ 
  // Copy all associated actions of ActiveMF.
11 TempAction  $\leftarrow$  Copy all actions of  $MF_{ActiveMF}$ 
12 TempAction  $\leftarrow \text{Sort}(\text{TempAction})$ 
13 Call modifyActions( $MF_{ActiveMF}$ )
  /* assume, each item of any  $L_i$  correspond with one appropriate membership function. */
14 for  $i \leftarrow 1$  to  $L_{ActiveMF} \text{Size}$  do
15    $v_i \leftarrow \text{item}(i)$ 
16   Assign each value of TempAction to each element of  $v_i$  respectively.
17   TMF  $\leftarrow$  Build trapezoidal membership function using  $v_i$ .
18    $\beta_\alpha(n)_{ActiveMF_i} \leftarrow$  apply objective function to the TMF.
19 end for
20  $\beta_\alpha(n) = \min(\beta_\alpha(n)_{ActiveMF_i} \mid 1 \leq i \leq L_{ActiveMF} \text{Size})$ 
  // steps for computing  $\beta_\mu(n)$ 
21 The steps for computing  $\beta_\mu(n)$  is similar to  $\beta_\alpha(n)$  with difference that the steps (10) and (11) are eliminated and in step (13) instead of  $\alpha(n)$  the  $\mu(n)$  is used.
  // Update parameters of LA
22 Update()
23  $n \leftarrow n + 1$ 
24 until for each LA of CALA, the  $\mu$  value does not change noticeably and the value  $\sigma$  converge to  $\sigma_L$ ;

```


Algorithm 3: ValidMembershipFunctions

Input: m // m is the number of fuzzy regions.
Output: L // L is the list of valid membership functions.

```

1  $L = []$ 
2  $R_1 = \{cr_{1,3}, wr_{1,4}\}$  //  $R_1$ , is first fuzzy region.
3  $R_m = \{wl_{m,1}, cl_{m,2}\}$  //  $R_m$ , is last fuzzy region.
4 if  $m = 1$  then
5    $L \leftarrow R_1$ 
6 else if  $m = 2$  then
7    $L \leftarrow R_1 \times R_2$  // The product Cartesian of  $R_1 \times R_2$  is assigned in  $L$ .
8 else
9   for  $L \leftarrow 2$  to  $m - 1$  do
10     $R_i \leftarrow \{wr_{i-1,4}, cl_{i,2}\} \times \{wr_{i-1,4}, cl_{i,2}, cr_{i,3}, wl_{i+1,1}\} \times$   

        $\{wr_{i-1,4}, cl_{i,2}, cr_{i,3}, wl_{i+1,1}\} \times \{cr_{i,3}, wl_{i+1,1}\}$ 
11  end for
12  // The product Cartesian of  $R_1 \times R_2, \dots, R_m$  is assigned in  $L$ .
13   $L \leftarrow R_1 \times R_2 \times \dots \times R_m$ 
14 end if
15 return  $L$ 
```

Algorithm 4: FindActiveMF

Output: ActiveMF
 // Count number of active membership functions.

```

1 ActiveMF  $\leftarrow 0$ 
2 for  $i \leftarrow 1$  to  $K_{max}$  do
3   if  $\alpha^i(n) \geq 0.5$  then
4     ActiveMF  $\leftarrow$  ActiveMF + 1
5   end if
6 end for
7 return ActiveMF
```

Algorithm 5: ModifyActions

// Check all selected actions are in the range of $[0,1]$

```

1 foreach chosen action  $\alpha^i(n)$  and  $\alpha_k^{i,j}(n)$  do
2   if its value is less than 0 or greater than 1 then
3     reinitialize it with the value between  $[0,1]$ .
4   end if
5 end foreach
6 // at least one action greater than 0.5 is exist.
7 if value of all chosen actions of  $\alpha^i(n), (1 \leq i \leq K_{max})$  is less than 0.5 then
8   randomly chose an action  $\alpha^i(n)$  and reinitialize it with a value between  $[0.5,1]$ .
9 end if
```

V. EXPERIMENTS AND ANALYSIS OF RESULTS

A. Experimental Setting

We conducted several experiments to show the effectiveness of the proposed approach. All algorithms implemented in Java on a 1.80 GHz Intel Core i7 processor with Windows 10 and 16 GB RAM. Two real data sets were used to assess the proposed CALA-AFTM, namely the DePaul CTI dataset [75] and NASA dataset [76]. These datasets were selected because the time parameter can be represented as TMFs [29],[44]. All HTTP requests in the NASA dataset from 23:59:59 PM on August 3, 1995 to 23:59:59 PM on August 31, 1995 were collected by NASA Kennedy Space Center in Florida. This dataset includes 45464 user sessions and 863 page views. The CTI data set was collected by the users who visited the website during two weeks in April 2002.

The CTI dataset after cleaning data contained 13745 sessions and 683 pages. Part of CTI dataset is depicted in Table II. In Table II, the page view and the related page are shown. In Table III, for each user the browsing sequences are represented.

TABLE II. PAGE VIEW AND CORRESPONDING PAGEVIEW IDS ON THE CTI DATASET

Page view Id	Page view
0	/admissions/
1	/admissions/career.asp
2	/admissions/checklist.asp
3	/admissions/costs.asp
:	
681	/shared/404.asp?404; http://www.cs.depaul.edu/msoffice/cltreq.asp
682	/shared/404.asp?404; http://www.cs.depaul.edu/resources/grad_scholarships.asp

TABLE III. THE BROWSING SEQUENCES ON THE CTI DATASET

Client ID	Browsing sequence
1	(679, 2) (574, 7) (585, 5) (604, 4)
2	(387, 37) (558, 20)
3	(387, 24) (400, 125) (71, 26) (228, 34)
...	...
13563	(54, 11) (358, 55)

To improve the convergence speed and generate actions between 0 and 1, we used the maximum value of time duration stored in two datasets and normalized these datasets values. In the proposed algorithm, for CTI and NASA datasets, web page number 387 (/news/default.asp) and web page number 588 (/shuttle/resources/orbiters/challenger) are considered for finding their appropriate membership functions, respectively.

The proposed CALA-AFTM algorithm compared with fuzzy web mining algorithm (FWMA) [31] and VSLA-AFTM algorithm. FWMA is used for fuzzy web mining applications and uses a predefined TMFs. Additionally, we developed another algorithm called VSLA-AFTM which uses VSLA to evaluate the results. The details of the implementation of VSLA-AFTM and CALA-AFTM are similar, and their differences are described below.

All learning automaton of CALA-AFTM used the continuous action-set, whereas the learning automaton of VSLA-AFTM except for the learning automata used in the threshold section used the discrete action-set. Therefore, finding the active membership functions in both algorithms is similar. VSLA-AFTM randomly selects an action.

Let r shows the number of actions, then the chosen action $\alpha_i (1 \leq i \leq r)$ corresponding to the value of $\frac{i-1}{r-1}$. CALA-AFTM algorithm needs to compute $\beta_a(n)$ and $\beta_b(n)$, while the VSLA-AFTM algorithm only requires to compute $\beta_a(n)$. The final value for $\beta_b(n)$ in CALA-AFTM is regarded as the best value for the cost function, while in VSLA-AFTM, the final value for $\beta_a(n)$ is taken as the best value for the cost function. In VSLA-AFTM, if the value of $\beta_a(n)$ is smaller than the previous step, all chosen actions are rewarded; otherwise, they are penalized according to (4) and (5), respectively.

Fig. 7 shows the predefined membership functions used in the experiments. To assess the results, parameters such as overlap, suitability, fuzzy support, coverage, the average value of cost function (AVCF), number of large 1-sequences (L_1), and execution time were considered. Each algorithm was run independently 30 times, and the mean and standard deviation of these 30 runs were considered. Table IV represents the parameter settings used in these algorithms.

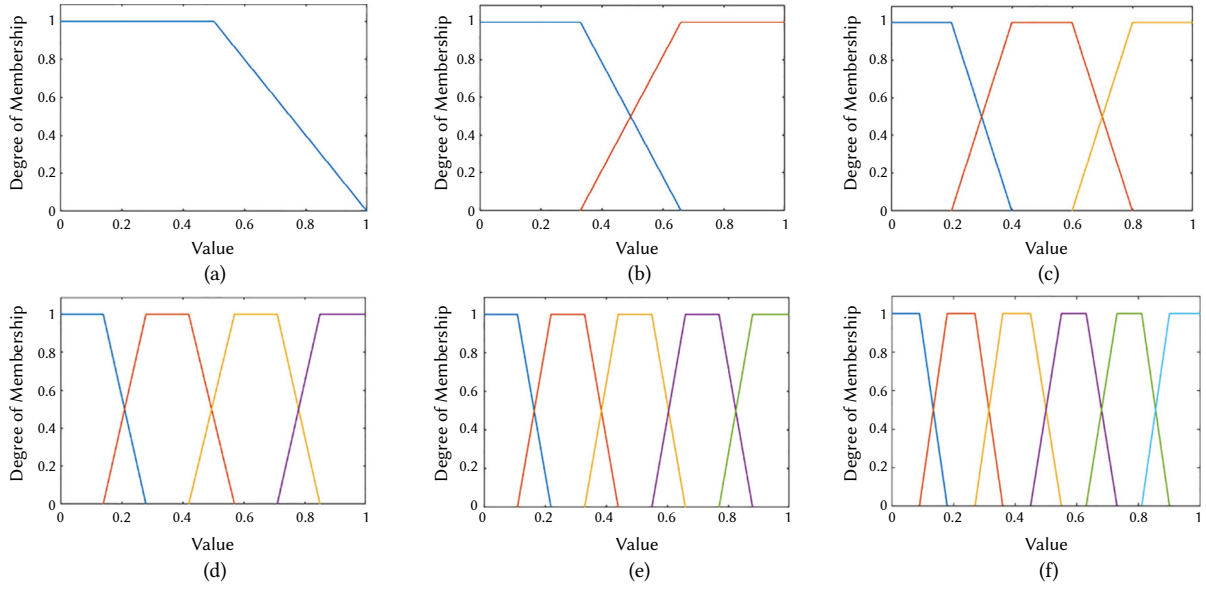


Fig. 7. The fixed trapezoidal membership functions used in experiments.

TABLE IV. PARAMETER SETTINGS

Parameter Name	Description	CALA-AFTM	VSLA-AFTM	FWMA
c	Penalty Constant	5		
σ_L	Lower bound for the variance	0.01		
λ	Step size for learning	0.02		
α	Minimum support	[0.002 – 0.01]	[0.002 – 0.01]	[0.002 – 0.01]
a	Reward		0.1	
b	Penalty		0.01	
r	Number of actions		[10, 20, 30, 40]	
k_{max}	Maximum number of MFs	6	6	6

TABLE V. APPROPRIATE NUMBER OF TMFs, COVERAGE, FUZZY SUPPORT, SUITABILITY, OVERLAP, AND AVERAGE VALUE OF COST FUNCTION ACQUIRED FROM CALA-AFTM, VSLA-AFTM, AND FWMA ON THE CTI DATASET

Algorithm	Optimal number of TMFs	Overlap	Coverage	Suitability	Fuzzy support	AVCF
CALA-AFTM	3.01±0.0100	0.0050±0.0023	1.0000±0.0000	1.0050±0.0023	1.4761±0.0864	0.6763±0.0389
VSLA-AFTM	4.13±0.3000	0.0674±0.0045	1.0000±0.0000	1.0674±0.0045	0.8667±0.0289	1.2321±0.0873
FWMA	3±0.0000	0.0000±0.0000	1.0000±0.0000	1.0000±0.0000	0.6524±0.0145	1.5328±0.0000

TABLE VI. APPROPRIATE NUMBER OF TMFs, COVERAGE, FUZZY SUPPORT, SUITABILITY, OVERLAP AND AVERAGE VALUE OF COST FUNCTION ACQUIRED FROM CALA-AFTM, VSLA-AFTM, AND FWMA ON THE NASA DATASET

Algorithm	Optimal number of TMFs	Overlap	Coverage	Suitability	Fuzzy support	AVCF
CALA-AFTM	4.15±0.2101	0.0452±0.0053	1.0000±0.0000	1.0452±0.0053	0.8742±0.0367	1.1923±0.1132
VSLA-AFTM	2.20±0.3420	0.1476±0.0075	1.0000±0.0000	1.1476±0.0075	0.7566±0.0158	1.5652±0.1745
FWMA	3±0.0000	0.0000±0.0000	1.0000±0.0000	1.0000±0.0000	0.5416±0.0236	1.8463±0.0000

B. Experimental Evaluations

In this experiment, we evaluated the results obtained using the CALA-AFTM, VSLA-AFTM, and FWMA algorithms on the CTI and NASA datasets. Parameters such as the optimal number of TMFs, fuzzy support, overlap, coverage, suitability, and AVCF were used to evaluate the results. The results of Tables V and VI shows that the proposed CALA-AFTM for NASA and CTI datasets produced three and four TMFs, respectively. VSLA-AFTM for CTI and NASA datasets produced four and two TMFs, respectively. FWMA algorithm used a fixed number of TMFs. So, by checking each $k_{max} \in [2, 6]$, it is found that the optimal number of TMFs is three.

By comparing the results, it can be seen that CALA-AFTM algorithm for the parameters AVCF, fuzzy support, and overlap has produced

better results. CALA-AFTM in the CTI datasets improved the value of AVCF and fuzzy support by 51% and 49%, respectively. Additionally, in the NASA dataset, CALA-AFTM improved the value of AVCF and fuzzy support by 30% and 26% respectively. The AVCF for both CTI and NASA datasets is depicted in Fig. 8a, b. Additionally, the results show that CALA-AFTM generated the minimum value of AVCF in the two datasets. Additionally, since the VSLA-AFTM algorithm uses a small number of actions, the convergence speed of VSLA-AFTM is faster than CALA-AFTM.

The suitability value for both CTI and NASA datasets is given in Fig. 9a, b. In the CTI dataset, the suitability value for FWMA and CALA-AFTM was almost the same. However, in the NASA dataset, FWMA produced better results.

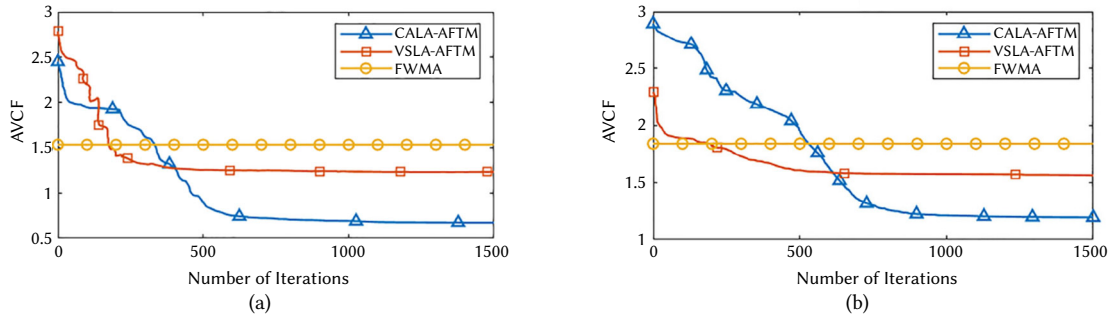


Fig. 8. The value of AVCF with different numbers of iterations on CTI (a) and NASA (b) datasets.

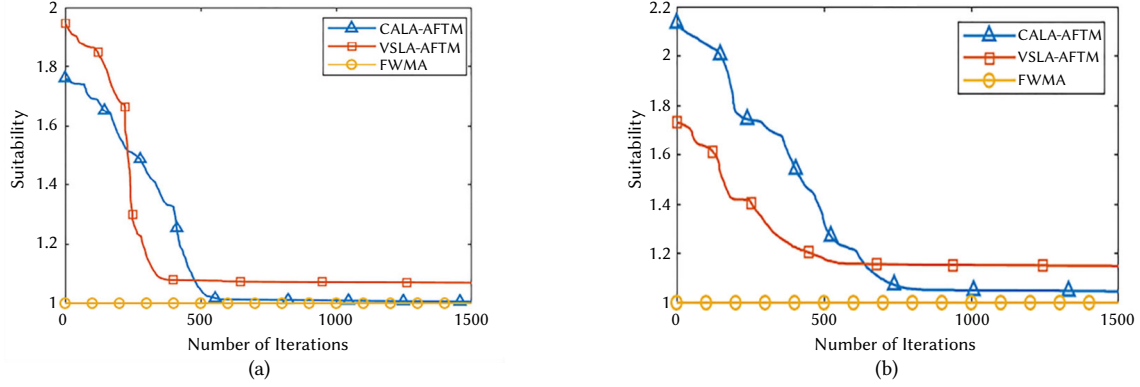


Fig. 9. The suitability value with different numbers of iterations on CTI (a) and NASA (b) datasets.

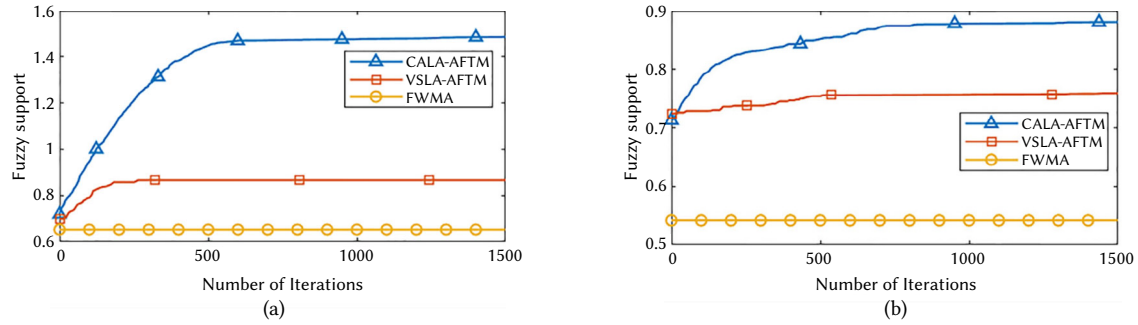


Fig. 10. The fuzzy support value with different numbers of iterations on CTI (a) and NASA (b) datasets.

The fuzzy support value for both CTI and NASA datasets is shown in Fig. 10a, b. The proposed CALA-AFTM produced a high fuzzy support. Therefore, the proposed cost function had a good performance. Figures 11 and 12 show the membership functions of the CTI and NASA datasets before and after 12,000 iterations of CALA-AFTM and VSLA-AFTM. Based on Figs. 11 and 12, the initial shape of membership functions are not appropriate. After optimization, the proper TMFs are produced. To assess the effect of dataset sizes on the efficiency of CALA-AFTM algorithm, another experiment was performed. The results of this experiment are shown in Tables VII and VIII. Tables VII and VIII compare the results for the overlap, coverage, fuzzy support, suitability, and the AVCF obtained from three test algorithms. By comparing the results between CALA-AFTM and VSLA-AFTM, we found that CALA-AFTM in the CTI and NASA datasets produces better results for AVCF, fuzzy support, and suitability parameters. VSLA-AFTM has a limited number of actions, so it cannot find the accurate value for the optimal parameters.

Additionally, the statistical significance of the AVCF was analyzed using unpaired t-test. The unpaired t-test results between CALA-AFTM and the other two algorithms for CTI and NASA datasets are

given in Tables VII and VIII, respectively. In this experiment, datasets of different sizes were tested. Tables VII and VIII show the values obtained for AVCF. Also, by considering the value of 0.95 for the confidence level, the p-value and t-value are specified. Let X and Y represent two algorithms respectively. In this case, X is statistically better than Y if the t-test (X, Y) is less than zero and the positive p-value is less than 0.05. The results in Tables IX and X show that CALA-AFTM is statistically significant than other algorithms. To assess the effect of number of extracted large 1-sequences and rules, another experiment was performed. In this experiment, the minimum confidence value was assumed to be 0.1 and the results were tested with different minimum support values. The results in Figs. 13 and 14 show that the CALA-AFTM algorithm produces a large number of 1-sequences and more rules than other algorithms. To assess the effect of association rules, another experiment was performed with different values for the minimum support and minimum confidence. The results are shown in Fig. 15a, b respectively. The result in Fig. 15a show that when the minimum support value is greater than 0.002, the number of extracted rules decreases significantly. Also, these result show that when the minimum support is greater than 0.004, the number of extracted rules will be less than 500 rules. Additionally, the result in

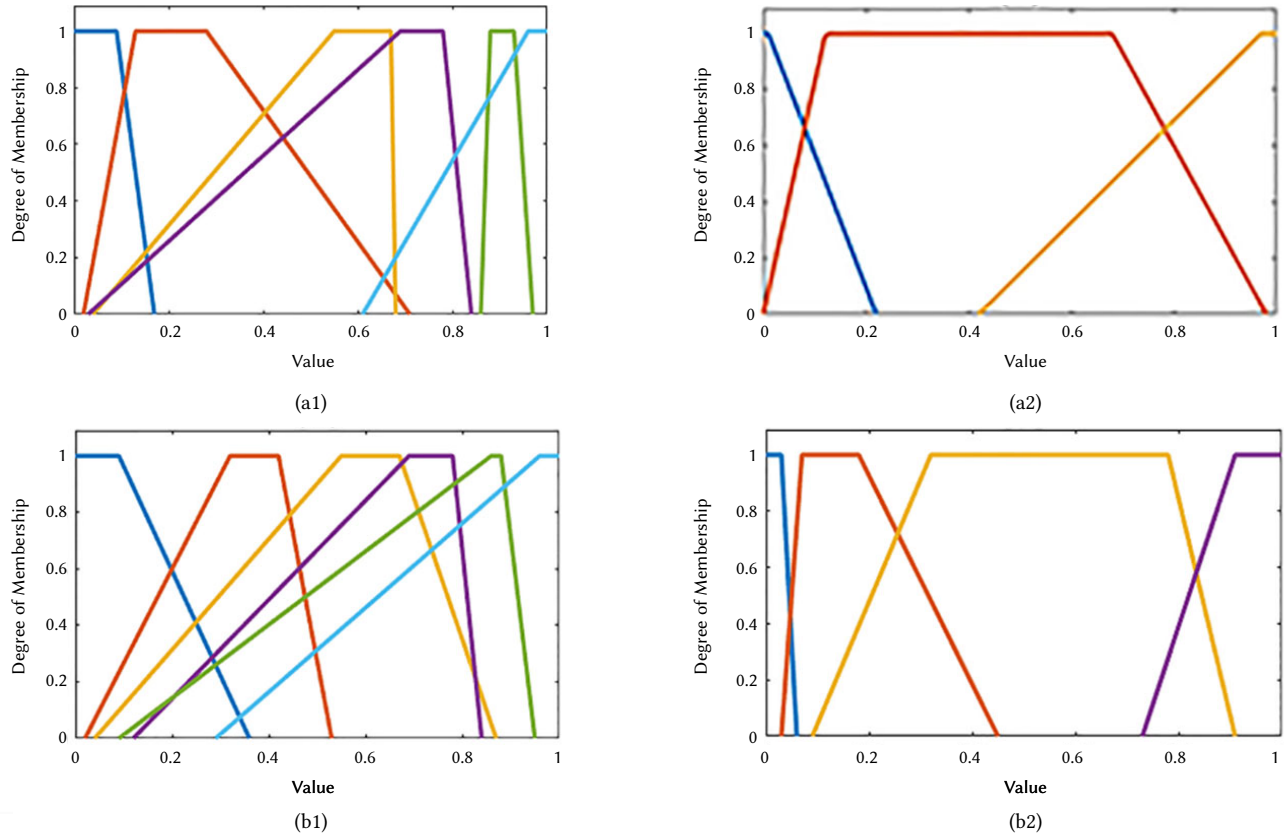


Fig. 11. The initial and optimized TMs derived by CALA-AFTM and VSLA-AFTM on the CTI dataset. (a₁) Initial TMs for CALA-AFTM, (b₁) Initial TMs for VSLA-AFTM, (a₂) Optimized TMs for CALA-AFTM, (b₂) Optimized TMs for VSLA-AFTM.

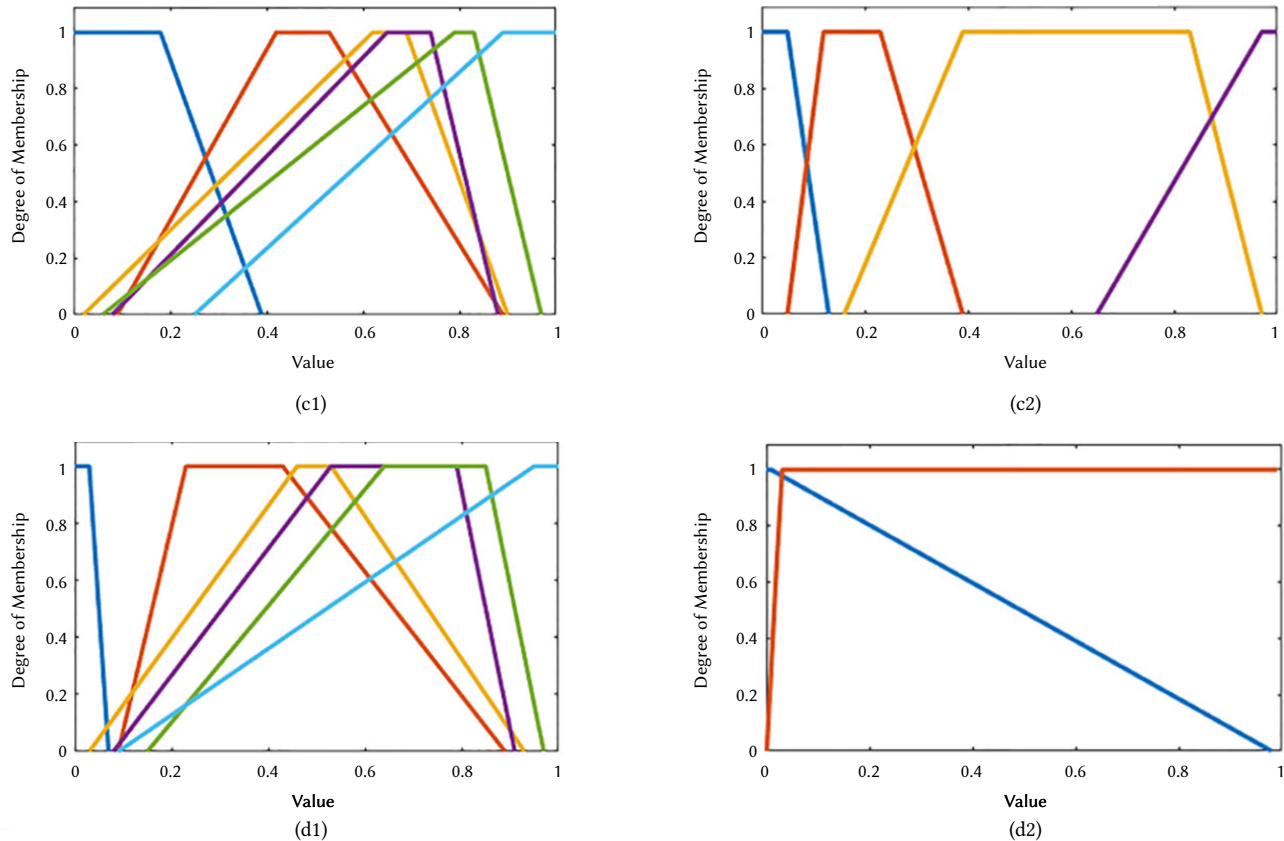


Fig. 12. The initial and optimized TMs derived by CALA-AFTM and VSLA-AFTM on the NASA dataset. (c₁) Initial TMs for CALA-AFTM, (d₁) Initial TMs for VSLA-AFTM, (c₂) Optimized TMs for CALA-AFTM, (d₂) Optimized TMs for VSLA-AFTM.

TABLE VII. COMPARISON OF THE RESULTS OF DIFFERENT SIZES OF DATA ON THE CTI DATASET

Data	Algorithm	Overlap	Coverage	Suitability	Fuzzy support	AVCF
50K	CALA-AFTM	0.0018	1.0000	1.0018	1.4918	0.6715±0.0232
	VSLA-AFTM	0.0245	1.0000	1.0245	0.8365	1.2246±0.0652
	FWMA	0.0000	1.0000	1.0000	0.6509	1.5362±0.0000
100K	CALA-AFTM	0.0025	1.0000	1.0025	1.5075	0.6650±0.0161
	VSLA-AFTM	0.04121	1.0000	1.04121	0.8547	1.2182±0.0580
	FWMA	0.0000	1.0000	1.0000	0.6594	1.5164±0.0000
150K	CALA-AFTM	0.0043	1.0000	1.0043	1.6156	0.6216±0.0672
	VSLA-AFTM	0.0386	1.0000	1.0386	0.8665	1.1986±0.1139
	FWMA	0.0000	1.0000	1.0000	0.6656	1.5023±0.0000
200K	CALA-AFTM	0.0039	1.0000	1.0039	1.5826	0.6343±0.0524
	VSLA-AFTM	0.0583	1.0000	1.0583	0.8784	1.2048±0.0943
	FWMA	0.0000	1.0000	1.0000	0.6632	1.5075±0.0000
250K	CALA-AFTM	0.0050	1.0000±0.0000	1.0050	1.4860	0.6763±0.0389
	VSLA-AFTM	0.0674	1.0000±0.0000	1.0674	0.8663	1.2321±0.0873
	FWMA	0.0000	1.0000±0.0000	1.0000	0.6524	1.5328±0.0000

TABLE VIII. COMPARISON OF THE RESULTS OF DIFFERENT SIZES OF DATA ON THE NASA DATASET

Data	Algorithm	Overlap	Coverage	Suitability	Fuzzy support	AVCF
130K	CALA-AFTM	0.0165	1.0000	1.0165	0.8618	1.1794±0.0923
	VSLA-AFTM	0.0724	1.0000	1.0724	0.6968	1.5388±0.1235
	FWMA	0.0000	1.0000	1.0000	0.5514	1.8132±0.0000
260K	CALA-AFTM	0.0268	1.0000	1.0268	0.8673	1.1838±0.0856
	VSLA-AFTM	0.0952	1.0000	1.0952	0.7087	1.5452±0.1361
	FWMA	0.0000	1.0000	1.0000	0.5478	1.8252±0.0000
390K	CALA-AFTM	0.0325	1.0000	1.0325	0.8684	1.1889±0.1230
	VSLA-AFTM	0.0863	1.0000	1.0863	0.6974	1.5576±0.1971
	FWMA	0.0000	1.0000	1.0000	0.5460	1.8312±0.0000
520K	CALA-AFTM	0.0409	1.0000	1.0409	0.8813	1.1810±0.0762
	VSLA-AFTM	0.1153	1.0000	1.1153	0.7236	1.5413±0.1426
	FWMA	0.0000	1.0000	1.0000	0.5496	1.8193±0.0000
650K	CALA-AFTM	0.0452	1.0000	1.0452	0.8766	1.1923±0.1132
	VSLA-AFTM	0.1476	1.0000	1.1476	0.7331	1.5652±0.1745
	FWMA	0.0000	1.0000	1.0000	0.5416	1.8463±0.0000

Fig. 15b show that the extracted rules reach to zero, when the value for the minimum support is greater than 0.05. The result in Fig. 16a, b shows that by increasing the confidence value the number of extracted rules decreases. Also, the results in Fig. 16b shows that when the value of minimum support is greater than 0.6 the curve becomes smoother. Additionally, when the value for the minimum confidence reaches to 0.9, the number of generated rules is zero. Another experiment was performed to evaluate the execution time of the proposed CALA-AFTM algorithm. In this experiment, different action sizes(r) with the values of 10, 20, 30, and 40 were considered for VSLA-AFTM algorithm. We also used values of 3 and 6 for maximum number of TMFs to evaluate the results.

To specify the number of actions in VSLA-AFTM, we employed the notation VSLA (r). The results of both CTI and NASA datasets are shown in Fig 17a, b and Fig18a, b. Figs. 17a, b illustrates average execution time increases when the maximum number of membership functions (k_{max}) was set at 3 for CTI and NASA dataset, respectively.

The execution time in VSLA-AFTM is directly related to the number of actions. VSLA-AFTM with the number of actions 10 and 20 converges faster and has a lower execution time than CALA-AFTM. The execution time of CALA-AFTM is less than VSLA-AFTM with the number of actions 30 and 40. Figs. 18a, b shows average execution time increases when the maximum number of membership functions (k_{max}) was set at 6 for CTI and NASA dataset, respectively. It can be seen that by increasing k_{max} and the size of the datasets the average execution time dramatically increases.

VI. CONCLUSIONS

In this paper a continuous action-set learning automata-based approach named CALA-AFTM was proposed to automatically determine both the optimal number and position of trapezoidal membership functions in order to extract fuzzy association rules from quantitative transactions. In this method, a new representation

TABLE IX. THE RESULTS OF T-VALUE AND P-VALUE WITH DIFFERENT DATASET SIZES ON CTI DATASET

Size	AVCF		t-value		p-value		Significant
	Algorithm	value	Pairwise algorithms	value	Pairwise algorithms	value	
50k	CALA-AFTM	0.6715±0.0232	CALA-AFTM, VSLA-AFTM	43.7753	CALA-AFTM, VSLA-AFTM	<0.0001	Extremely statistically significant
	VSLA-AFTM	1.2246±0.0652	CALA-AFTM, FWMA	204.1447	CALA-AFTM, FWMA	<0.0001	Extremely statistically significant
	FWMA	1.5362±0.0000					
100k	Algorithm	value	Pairwise algorithms	value	Pairwise algorithms	value	
	CALA-AFTM	0.6650±0.0161	CALA-AFTM, VSLA-AFTM	50.3380	CALA-AFTM, VSLA-AFTM	<0.0001	Extremely statistically significant
	VSLA-AFTM	1.2182±0.0580	CALA-AFTM, FWMA	289.6466	CALA-AFTM, FWMA	<0.0001	Extremely statistically significant
	FWMA	1.5164±0.0000					
150k	Algorithm	value	Pairwise algorithms	value	Pairwise algorithms	value	
	CALA-AFTM	0.6216±0.0672	CALA-AFTM, VSLA-AFTM	23.8975	CALA-AFTM, VSLA-AFTM	<0.0001	Extremely statistically significant
	VSLA-AFTM	1.1986±0.1139	CALA-AFTM, FWMA	71.7826	CALA-AFTM, FWMA	<0.0001	Extremely statistically significant
	FWMA	1.5023±0.0000					
200k	Algorithm	value	Pairwise algorithms	value	Pairwise algorithms	value	
	CALA-AFTM	0.6343±0.0524	CALA-AFTM, VSLA-AFTM	28.9649	CALA-AFTM, VSLA-AFTM	<0.0001	Extremely statistically significant
	VSLA-AFTM	1.2048±0.0943	CALA-AFTM, FWMA	91.2732	CALA-AFTM, FWMA	<0.0001	Extremely statistically significant
	FWMA	1.5075±0.0000					
250k	Algorithm	value	Pairwise algorithms	value	Pairwise algorithms	value	
	CALA-AFTM	0.6763±0.0389	CALA-AFTM, VSLA-AFTM	31.8520	CALA-AFTM, VSLA-AFTM	<0.0001	Extremely statistically significant
	VSLA-AFTM	1.2321±0.0873	CALA-AFTM, FWMA	120.5975	CALA-AFTM, FWMA	<0.0001	Extremely statistically significant
	FWMA	1.5328±0.0000					

TABLE X. THE RESULTS FOR THE P-VALUE AND T-VALUE PARAMETERS ON NASA DATASET

Size	AVCF		t-value		p-value		significant
	Algorithm	value	Pairwise algorithms	value	Pairwise algorithms	value	
130k	CALA-AFTM	1.1794±0.0464	CALA-AFTM, VSLA-AFTM	19.0551	CALA-AFTM, VSLA-AFTM	<0.0001	Extremely statistically significant
	VSLA-AFTM	1.5388±0.0923	CALA-AFTM, FWMA	74.8161	CALA-AFTM, FWMA	<0.0001	Extremely statistically significant
	FWMA	1.8132±0.0000					
260k	Algorithm	value	Pairwise algorithms	value	Pairwise algorithms	value	
	CALA-AFTM	1.1838±0.0856	CALA-AFTM, VSLA-AFTM	12.9879	CALA-AFTM, VSLA-AFTM	<0.0001	Extremely statistically significant
	VSLA-AFTM	1.5452±0.1261	CALA-AFTM, FWMA	41.0408	CALA-AFTM, FWMA	<0.0001	Extremely statistically significant
	FWMA	1.8252±0.0000					
390k	Algorithm	value	Pairwise algorithms	value	Pairwise algorithms	value	
	CALA-AFTM	1.1889±0.1230	CALA-AFTM, VSLA-AFTM	8.6922	CALA-AFTM, VSLA-AFTM	<0.0001	Extremely statistically significant
	VSLA-AFTM	1.5576±0.1971	CALA-AFTM, FWMA	28.6018	CALA-AFTM, FWMA	<0.0001	Extremely statistically significant
	FWMA	1.8312±0.0000					
520K	Algorithm	value	Pairwise algorithms	value	Pairwise algorithms	value	
	CALA-AFTM	1.1810±0.0762	CALA-AFTM, VSLA-AFTM	12.2057	CALA-AFTM, VSLA-AFTM	<0.0001	Extremely statistically significant
	VSLA-AFTM	1.5413±0.1426	CALA-AFTM, FWMA	45.8807	CALA-AFTM, FWMA	<0.0001	Extremely statistically significant
	FWMA	1.8193±0.0000					
650k	Algorithm	value	Pairwise algorithms	value	Pairwise algorithms	value	
	CALA-AFTM	1.1923±0.1132	CALA-AFTM, VSLA-AFTM	9.8195	CALA-AFTM, VSLA-AFTM	<0.0001	Extremely statistically significant
	VSLA-AFTM	1.5652±0.1745	CALA-AFTM, FWMA	31.6440	CALA-AFTM, FWMA	<0.0001	Extremely statistically significant
	FWMA	1.8463±0.0000					

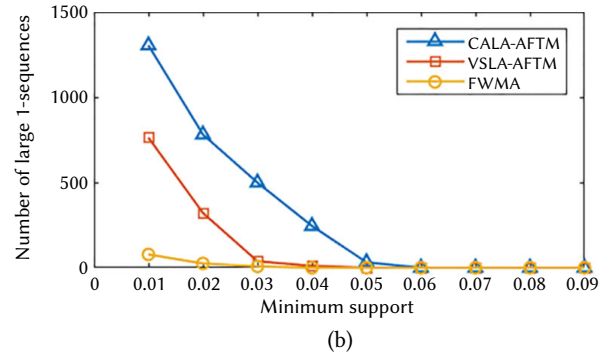
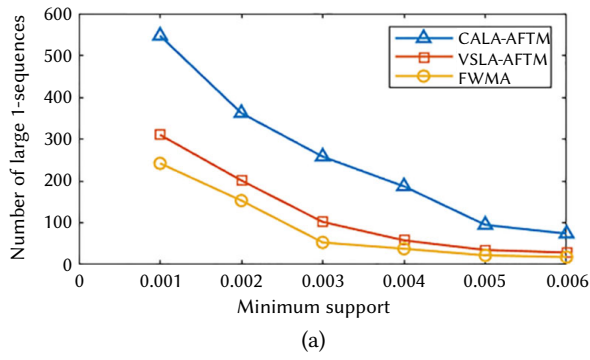


Fig. 13. The effect of number of extracted large 1-sequences on CTI dataset (a), NASA dataset (b).

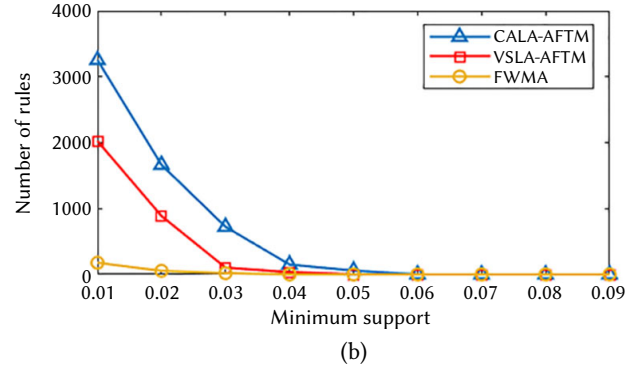
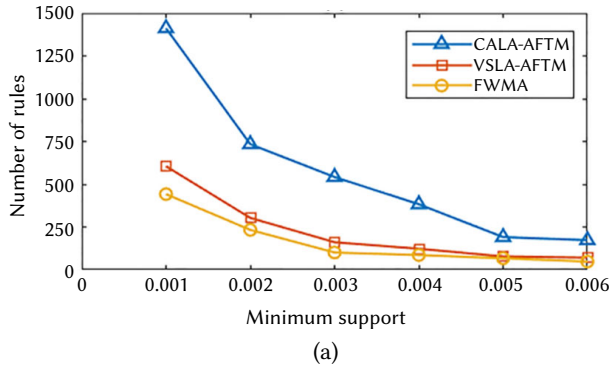


Fig. 14. The effect of number of extracted rules on CTI dataset (a), NASA dataset (b).

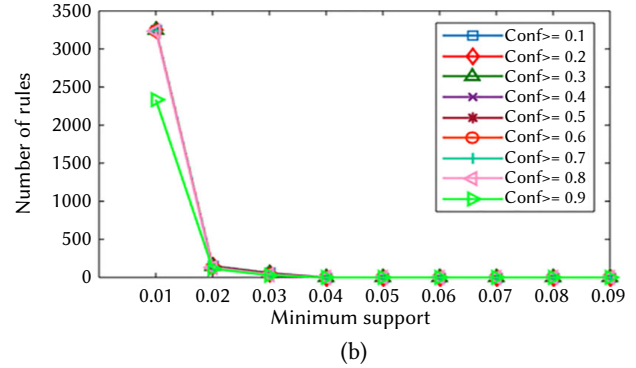
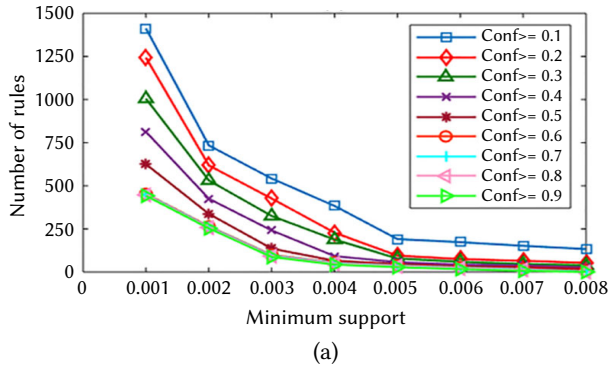


Fig. 15. The effect of numbers of extracted rules and the minimum support with different values for minimum confidence on CTI dataset (a), NASA dataset (b).

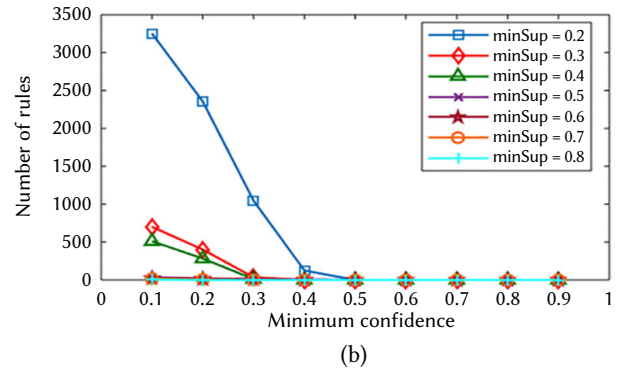
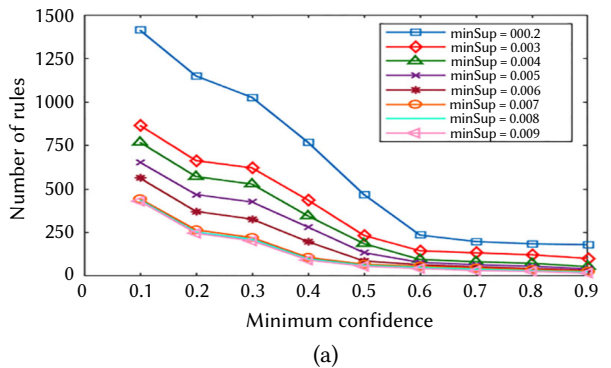


Fig. 16. The effect of numbers of extracted rules and the minimum confidence with different values for minimum support on CTI dataset (a), NASA dataset (b).

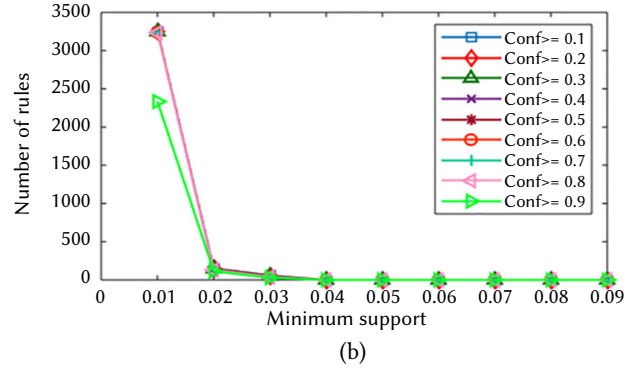
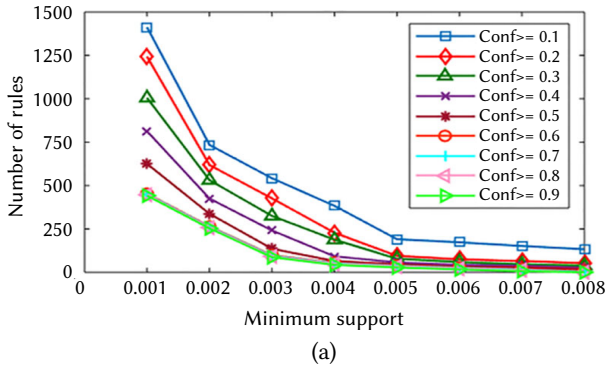


Fig. 17. The average execution time along with different dataset sizes with kmax=3 on CTI dataset (a), NASA dataset (b).

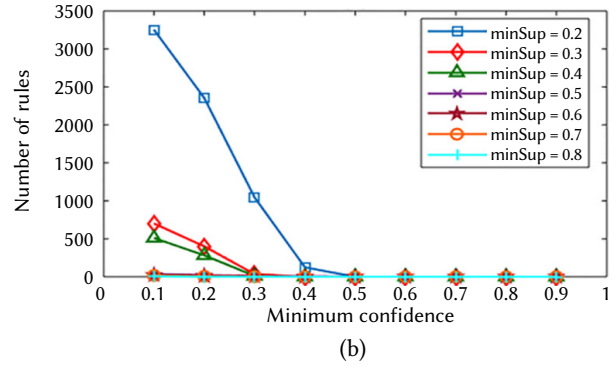
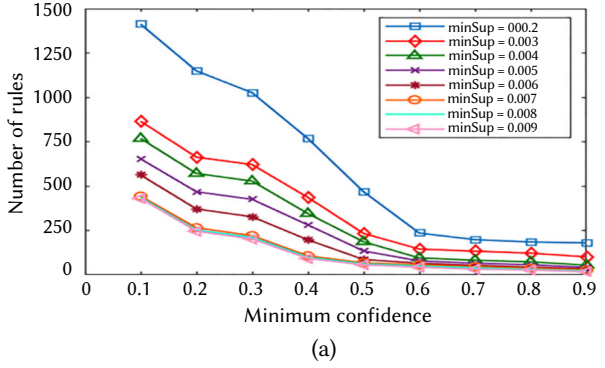


Fig. 18. The average execution time along with different dataset sizes with kmax=6 on CTI dataset (a), NASA dataset (b).

is introduced to construct a team of CALA for cooperating learning automata. The learning automaton of this team of CALA is composed of two parts. The learning automata of the first part try to determine the proper number of membership functions, and the relative learning automata in the second part optimize their positions. The proposed algorithm has been tested on web usage data, to find membership functions and significant rules. Therefore, the extracted rules represent the browsing behavior of users and can be used to give some proper suggestions to web-server administrators. In the proposed CALA-AFTM, to reduce the domain of the search space and remove unsuitable membership functions, two conditions were applied and a novel algorithm was proposed. Also, to check the effectiveness of the proposed CALA-AFTM, various experiments were performed using two real data sets. We showed that with increasing the different dataset sizes, compared with other algorithms, CALA-AFTM provides better results in terms of fuzzy support, overlap, number of large 1-sequences, coverage, number of rules, and cost function. In future work, our goal will be to develop the proposed method in topics such as multi-objective applications, fuzzy temporal rule mining, fuzzy generalized association rule mining, and 2-tuple fuzzy linguistic representation.

REFERENCES

- [1] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, "From data mining to knowledge discovery in databases," *AI Magazine*, vol. 17, no. 3, pp. 37-37, 1996.
- [2] M.-S. Chen, J. Han, and P. S. Yu, "Data mining: an overview from a database perspective," *IEEE Transactions on Knowledge and data Engineering*, vol. 8, no. 6, pp. 866-883, 1996.
- [3] K. Wagstaff, C. Cardie, S. Rogers, and S. Schrödl, "Constrained k-means clustering with background knowledge," in *Proceedings of the 8th International Conference on Machine Learning*, 2001, vol. 1, pp. 577-584.
- [4] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM transactions on intelligent systems and technology (TIST)*, vol. 2, no. 3, p. 27, 2011.
- [5] T. Hastie, R. Tibshirani, J. Friedman, and J. Franklin, "The elements of statistical learning: data mining, inference and prediction," *The Mathematical Intelligencer*, vol. 27, no. 2, pp. 83-85, 2005.
- [6] C. C. Aggarwal and C. Zhai, *Mining text data*, New York, NY: Springer, 2012.
- [7] T.-P. Hong, C.-H. Chen, Y.-C. Lee, and Y.-L. Wu, "Genetic-fuzzy data mining with divide-and-conquer strategy," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 2, pp. 252-265, 2008.
- [8] G. Sumathi and J. Akilandeswari, "Improved fuzzy weighted-iterative association rule-based ontology postprocessing in data mining for query recommendation applications," *Computational Intelligence*, vol. 36, no. 2, pp. 773-782, 2020.
- [9] C. H. Cheng and C. H. Chen, "Fuzzy time series model based on weighted association rule for financial market forecasting," *Expert Systems*, vol. 35, no. 4, p. e12271, 2018.
- [10] F. Chiclana, R. Kumar, M. Mittal, M. Khari, J. M. Chatterjee, and S. W. Baik, "ARM-AMO: an efficient association rule mining algorithm based on animal migration optimization," *Knowledge-Based Systems*, vol. 154, pp. 68-80, 2018.
- [11] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, and A. I. Verkamo, "Fast discovery of association rules," *Advances in knowledge discovery and data mining*, vol. 12, no. 1, pp. 307-328, 1996.
- [12] R. Agrawal, T. Imieliński, and A. Swami, "Mining association rules between sets of items in large databases," in *Proceedings of the 1993 ACM SIGMOD international conference on Management of data (SIGMOD'98)*, New York: ACM Press, 1993, pp. 207-216.
- [13] R. Agrawal, "Fast Algorithm for Mining Association Rules in Large Databases," in *Proceedings of International Conference on Very Large Databases (VLDB '94)*, Santiago, Chile, 1994, pp. 487-499.
- [14] J. Chen, A. Mikulicic, and D. H. Kraft, *An integrated approach to information retrieval with fuzzy clustering and fuzzy inferencing*, in *Knowledge Management in Fuzzy Databases* (pp. 247-260), Heidelberg, Germany: Physica-Verlag, 2000.
- [15] H. Ishibuchi, T. Nakashima, and M. Nii, *Classification and modeling with linguistic information granules: Advanced approaches to linguistic Data Mining*, Springer Science & Business Media, Berlin, Germany: Springer-Verlag, 2004.
- [16] W. Siler and J. J. Buckley, *Fuzzy expert systems and fuzzy reasoning*. John

- Wiley & Sons, Hoboken, NJ, USA, p. 424.
- [17] H. Zhang and D. Liu, *Fuzzy modeling and fuzzy control*, Boston, MA, USA: Birkhauser, 2006.
- [18] A. Kandel, *Fuzzy expert systems*, Boca Raton, Florida, USA: CRC Press, 1991.
- [19] L. A. Zadeh, "Fuzzy sets," *Information and control*, vol. 8, no. 3, pp. 338-353, 1965.
- [20] K. C. Chan and W.-H. Au, "Mining fuzzy association rules," in *Proceedings of the sixth international conference on Information and knowledge management*, Las Vegas, NV, 1997, pp. 209-215.
- [21] J. S. Yue, E. Tsang, D. Yeung, and D. Shi, "Mining fuzzy association rules with weighted items," in *proceedings of the ieee international conference on systems, man and cybernetics*, Nashville, TN, 2000, pp. 1906-1911.
- [22] C. M. Kuok, A. Fu, and M. H. Wong, "Mining fuzzy association rules in databases," *ACM Sigmod Record*, vol. 27, no. 1, pp. 41-46, 1998.
- [23] N. Marin, M. D. Ruiz, and D. Sánchez, "Fuzzy frameworks for mining data associations: fuzzy association rules and beyond," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 6, no. 2, pp. 50-69, 2016.
- [24] R. Pierrard, J.-P. Poli, and C. Hudelot, "A Fuzzy Close Algorithm for Mining Fuzzy Association Rules," in *International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, Springer, Cham, 2018, pp. 88-99.
- [25] C. Molina, M. D. Ruiz, and J. M. Serrano, "Representation by levels: An alternative to fuzzy sets for fuzzy data mining," *Fuzzy Sets and Systems*, vol. 401, pp. 113-132, 2019.
- [26] E. Cuevas, J. Gálvez, and O. Avalos, *Fuzzy Logic Based Optimization Algorithm*, in *Recent Metaheuristics Algorithms for Parameter Identification*, pp. 135-181, Springer, Cham, 2020.
- [27] S. G. Matthews, M. A. Gongora, A. A. Hopgood, and S. Ahmadi, "Web usage mining with evolutionary extraction of temporal fuzzy association rules," *Knowledge-Based Systems*, vol. 54, pp. 66-72, 2013.
- [28] R. Wu, "Mining generalized fuzzy association rules from Web logs," in *2010 Seventh International Conference on Fuzzy Systems and Knowledge Discovery*, 2010, vol. 5, pp. 2474-2477: IEEE.
- [29] T.-P. Hong, C.-M. Huang, and S.-J. Horng, "Linguistic object-oriented web-usage mining," *International Journal of Approximate Reasoning*, vol. 48, no. 1, pp. 47-61, 2008.
- [30] C. Chai and B. Li, "A novel association rules method based on genetic algorithm and fuzzy set strategy for web mining," *Journal of Computers*, vol. 5, no. 9, pp. 1448-1455, 2010.
- [31] Y.-H. Tao, T.-P. Hong, W.-Y. Lin, and W.-Y. Chiu, "A practical extension of web usage mining with intentional browsing data toward usage," *Expert Systems with Applications*, vol. 36, no. 2, pp. 3937-3945, 2009.
- [32] C.-H. Chen, G.-C. Lan, T.-P. Hong, and S.-B. Lin, "Mining fuzzy temporal association rules by item lifespans," *Applied Soft Computing*, vol. 41, pp. 265-274, 2016.
- [33] C.-K. Ting, T.-C. Wang, R.-T. Liaw, and T.-P. Hong, "Genetic algorithm with a structure-based representation for genetic-fuzzy data mining," *Soft Computing*, vol. 21, no. 11, pp. 2871-2882, 2017.
- [34] T.-P. Hong, Y.-C. Lee, and M.-T. Wu, "An effective parallel approach for genetic-fuzzy data mining," *Expert Systems with Applications*, vol. 41, no. 2, pp. 655-662, 2014.
- [35] J. C.-W. Lin, T. Li, P. Fournier-Viger, T.-P. Hong, J. M.-T. Wu, and J. Zhan, "Efficient mining of multiple fuzzy frequent itemsets," *International Journal of Fuzzy Systems*, vol. 19, no. 4, pp. 1032-1040, 2017.
- [36] G. Santharam, P. Sastry, and M. Thathachar, "Continuous action set learning automata for stochastic optimization," *Journal of the Franklin Institute*, vol. 331, no. 5, pp. 607-628, 1994.
- [37] H. Beigy and M. Meybodi, "A new continuous action-set learning automaton for function optimization," *Journal of the Franklin Institute*, vol. 343, no. 1, pp. 27-47, 2006.
- [38] K. S. Narendra and M. A. Thathachar, *Learning automata: an introduction*, Prentice-Hall, Englewood Cliffs, NJ: Prentice Hall, 1989.
- [39] M. A. Thathachar and P. S. Sastry, *Networks of learning automata: Techniques for online stochastic optimization*, Dordrecht, The Netherlands: Kluwer, 2004.
- [40] S. Sanyal, M. Bansal, S. Banerjee, and P. Kalra, "On learning shapes from shades," in *Proceedings of 4th Indian Conference on Computer Vision Graphics and Image Processing*, Calcutta, India, 2004, pp. 275-282.
- [41] H. Beigy and M. R. Meybodi, "An adaptive call admission algorithm for cellular networks," *Computers & Electrical Engineering*, vol. 31, no. 2, pp. 132-151, 2005.
- [42] B. Anari, J. A. Torkestani, and A. M. Rahmani, "Automatic data clustering using continuous action-set learning automata and its application in segmentation of images," *Applied Soft Computing*, vol. 51, pp. 253-265, 2017.
- [43] T.-P. Hong, M.-J. Chiang, and S.-L. Wang, "Mining weighted browsing patterns with linguistic minimum supports," in *IEEE International Conference on Systems, Man and Cybernetics*, 2002, Yasmine Hammamet, Tunisia, pp. 635-639.
- [44] T.-P. Hong, M.-J. Chiang, and S.-L. Wang, "Mining fuzzy weighted browsing patterns from time duration and with linguistic thresholds," *American Journal of Applied Sciences*, vol. 5, no. 12, pp. 1611, 2008.
- [45] R. Wu, W. Tang, and R. Zhao, "Web mining of preferred traversal patterns in fuzzy environments," *Springer-Verlag, Berlin Heidelberg Lecture Notes in Artificial Intelligence*, vol. 3642, 2005, pp. 456-465.
- [46] S.-L. Wang, W.-S. Lo, and T.-P. Hong, "Discovery of fuzzy multiple-level Web browsing patterns," in *Proceedings of the international conference on fuzzy systems and knowledge discovery*: Springer, Singapore, 2005, pp. 251-266.
- [47] W.-S. Lo, T.-P. Hong, and S.-L. Wang, "A top-down fuzzy cross-level Web-mining approach," in: *The 2003 IEEE International Conference on Systems, Man and Cybernetics*, 2003, vol. 3, pp. 2684-2689.
- [48] T.-P. Hong, K.-Y. Lin, and B.-C. Chien, "Mining fuzzy multiple-level association rules from quantitative data," *Applied Intelligence*, vol. 18, no. 1, pp. 79-90, 2003.
- [49] T.-P. Hong, C.-H. Chen, Y.-L. Wu, and Y.-C. Lee, "A GA-based fuzzy mining approach to achieve a trade-off between number of rules and suitability of membership functions," *Soft Computing*, vol. 10, no. 11, pp. 1091-1101, 2006.
- [50] C.-H. Chen, V. S. Tseng, and T.-P. Hong, "Cluster-based evaluation in fuzzy-genetic data mining," *IEEE transactions on fuzzy systems*, vol. 16, no. 1, pp. 249-262, 2008.
- [51] C.-H. Chen, Y. Li, and T.-P. Hong, "Type-2 genetic-fuzzy mining with tuning mechanism," in *2015 Conference on Technologies and Applications of Artificial Intelligence (TAAI)*, 2015, pp. 296-299: IEEE.
- [52] J. Alcalá-Fdez, R. Alcalá, M. J. Gacto, and F. Herrera, "Learning the membership function contexts for mining fuzzy association rules by using genetic algorithms," *Fuzzy Sets and Systems*, vol. 160, no. 7, pp. 905-921, 2009.
- [53] W. Wang and S. Bridges, "Genetic algorithm optimization of membership functions for mining fuzzy association rules," in: *International Joint Conference on information Systems, Fuzzy Theory and Technology Conference*, Atlantic City, NY, 2000, pp. 1-4.
- [54] C.-H. Chen, Y. Li, T.-P. Hong, Y.-K. Li, and E. H.-C. Lu, "A GA-based approach for mining membership functions and concept-drift patterns," in *2015 IEEE Congress on Evolutionary Computation (CEC)*, 2015, pp. 2961-2965: IEEE.
- [55] C.-H. Chen, T.-P. Hong, Y.-C. Lee, and V. S. Tseng, "Finding active membership functions for genetic-fuzzy data mining," *International Journal of Information Technology & Decision Making*, vol. 14, no. 06, pp. 1215-1242, 2015.
- [56] A. M. Palacios, J. L. Palacios, L. Sánchez, and J. Alcalá-Fdez, "Genetic learning of the membership functions for mining fuzzy association rules from low quality data," *Information Sciences*, vol. 295, pp. 358-378, 2015.
- [57] M. Kaya and R. Alhajj, "Genetic algorithms-based optimization of membership functions for fuzzy weighted association rules mining," in *Proceedings of International Conference Symposium on Computers and Communications ISCC*, 2004, vol. 1, pp. 110-115.
- [58] T.-P. Hong, Y.-F. Tung, S.-L. Wang, M.-T. Wu, and Y.-L. Wu, "An ACS-based framework for fuzzy data mining," *Expert Systems with Applications*, vol. 36, no. 9, pp. 11844-11852, 2009.
- [59] M.-T. Wu, T.-P. Hong, and C.-N. Lee, "A continuous ant colony system framework for fuzzy data mining," *Soft Computing*, vol. 16, no. 12, pp. 2071-2082, 2012.
- [60] C.-K. Ting, R.-T. Liaw, T.-C. Wang, and T.-P. Hong, "Mining fuzzy association rules using a memetic algorithm based on structure representation," *Memetic Computing*, vol. 10, no. 1, pp. 15-28, 2018.
- [61] J. Botzheim, C. Cabrita, L. T. Kóczy, and A. Ruano, "Fuzzy rule extraction

by bacterial memetic algorithms,” *International Journal of Intelligent Systems*, vol. 24, no. 3, pp. 312-339, 2009.

- [62] A. Song, J. Song, X. Ding, G. Xu, and J. Chen, “Utilizing bat algorithm to optimize membership functions for fuzzy association rules mining,” in *International Conference on Database and Expert Systems Applications*, 2017, pp. 496-504.
- [63] M. A. Chamazi and H. Motameni, “Finding suitable membership functions for fuzzy temporal mining problems using fuzzy temporal bees’ method,” *Soft Computing*, vol. 23, no. 10, pp. 3501-3518, 2019.
- [64] F. Alikhademi and S. Zainudin, “Generating of derivative membership functions for fuzzy association rule mining by Particle Swarm Optimization,” in *2014 International Conference on Computational Science and Technology (ICCST)*, 2014, pp. 1-6.
- [65] F. Rudziński, “A multi-objective genetic optimization of interpretability-oriented fuzzy rule-based classifiers,” *Applied Soft Computing*, vol. 38, pp. 118-133, 2016.
- [66] B. Minaei-Bidgoli, R. Barmaki, and M. Nasiri, “Mining numerical association rules via multi-objective genetic algorithms,” *Information Sciences*, vol. 233, pp. 15-24, 2013.
- [67] H. R. Qodmanan, M. Nasiri, and B. Minaei-Bidgoli, “Multi objective association rule mining with genetic algorithm without specifying minimum support and minimum confidence,” *Expert Systems with applications*, vol. 38, no. 1, pp. 288-298, 2011.
- [68] U. M. Patil and J. Patil, “Mining fuzzy association rules from web usage quantitative data,” *Computer Science & Information Technology*, vol. 89, 2016.
- [69] M. Ghavipour and M. R. Meybodi, “An adaptive fuzzy recommender system based on learning automata,” *Electronic Commerce Research and Applications*, vol. 20, pp. 105-115, 2016.
- [70] M. A. Thathachar and P. S. Sastry, “Varieties of learning automata: an overview,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 32, no. 6, pp. 711-722, 2002.
- [71] K. S. Narendra and M. A. Thathachar, “On the behavior of a learning automaton in a changing environment with application to telephone traffic routing,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 10, no. 5, pp. 262-269, 1980.
- [72] N. Kumar, J.-H. Lee, and J. J. Rodrigues, “Intelligent mobile video surveillance system as a Bayesian coalition game in vehicular sensor networks: Learning automata approach,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 3, pp. 1148-1161, 2014.
- [73] A. Helmzadeh and S. M. Kouhsari, “Calibration of erroneous branch parameters utilising learning automata theory,” *IET Generation, Transmission & Distribution*, vol. 10, no. 13, pp. 3142-3151, 2016.
- [74] B. Anari, J. Akbari Torkestani, and A. M. Rahmani, “A learning automata-based clustering algorithm using ant swarm intelligence,” *Expert systems*, vol. 35, no. 6, p. e12310, 2018.
- [75] <http://www.cs.depaul.edu>.
- [76] <https://ita.ee.lbl.gov/html/traces.html>.



Zohreh Anari

Zohreh Anari is currently Associate Professor in Computer Science at Department of Computer Engineering and Information Technology, Payame Noor University, Iran. She also received the Ph.D. degree in Computer Engineering from Islamic Azad University, Urmia, Iran, in 2020. She joined the faculty member of Computer Engineering Department at Payame Noor University, Shabestar, Iran in 2008. Her research interests are learning systems, data mining, web mining, fuzzy systems and soft computing. Recently, she is working in the field of fog computing and Blockchain.



Abdolreza Hatamlou

Abdolreza Hatamlou is Associate Professor of the Department of Computer Science at Islamic Azad University, Khoy Branch where he has been a faculty member since 2002. He completed his Ph.D. at National University of Malaysia in 2012. His research interests lie in the area of optimization, meta-heuristic and nature-inspired algorithms, data mining and clustering. He has collaborated actively with researchers in several other disciplines of computer science. Abdolreza Hatamlou has served on roughly twenty conference and workshop program committees.



Babak Anari

Babak Anari is currently is Assistance Professor in Department of Computer Science at Islamic Azad University, Shabestar Branch. He completed his Ph.D. at Science and Research Branch, Islamic Azad University (IAU), Tehran, in 2017. His research interests are in the areas of Learning Automata, Cellular Automata, fog computing, Blockchain, and IoT.