Journal of AI and Data Mining
Vol 7, No 1, 2019, 77-96
DOI: 10.22044/JADM.2018.5506.1662

S3PSO: Students' Performance Prediction Based on Particle Swarm Optimization

S. M.- H. Hasheminejad* and M. Sarvmili

Department of Computer Engineering, Alzahra University, Tehran, Iran.

Received 11 March 2017; Revised 04 November 2017; Accepted 30 January 2018
**Corresponding author: SMH.Hasheminejad@Alzahra.ac.ir (SMH. Hasheminejad).*

Abstract

Nowadays new methods are required to take advantage of the rich and extensive gold mine of data, given the vast content of data particularly created by educational systems. Data mining algorithms have been used in educational systems, especially e-learning systems, due to the broad usage of these systems. Providing a model to predict the final student results in an educational course is a reason for using data mining in educational systems. In this paper, we propose a novel rule-based classification method called *S3PSO* (*Students' Performance Prediction based on Particle Swarm Optimization*) to extract the hidden rules that could be used to predict the students' final outcome. The proposed *S3PSO* method is based upon the *Particle Swarm Optimization* (PSO) algorithm in a discrete space. The *S3PSO* particle encoding inducts more interpretable even for normal users like instructors. In *S3PSO*, the *Support*, *Confidence*, and *Comprehensibility* criteria are used to calculate the fitness of each rule. Comparing the results obtained from *S3PSO* with other rule-based classification methods such as CART, C4.5, and ID3 reveals that *S3PSO* improves 31% of the value of fitness measurement for the Moodle dataset. Additionally, comparing the results obtained from *S3PSO* with other classification methods such as SVM, KNN, Naïve Bayes, Neural Network, and APSO reveals that *S3PSO* improves 9% of the *accuracy* value for the Moodle dataset and yields promising results for predicting the students' final outcome.

Keywords: *Educational Data Mining, Particle Swarm Optimization, Rule-Based Classification.*

1. Introduction

Exponential growth of computer-based (especially web-based) educational systems in the recent years has been spurred by the fact that neither students nor teachers are bound to a specific location. These types of educational systems are not dependent upon any hardware platform [1]. Computer-based systems are stand-alone educational applications installed on a local computer to resolve primary educational requirements. Today the global use of internet as well as computer-based educational systems encourages the educational environments to use web-based systems such as e-learning systems, e-training systems, and online instruction systems. The increasing use of artificial intelligence techniques is an incentive to employ them in developing new and adapting educational systems. Learning and Management System (LMS), Intelligence Tutoring System (ITS), Adaptive and Intelligent Hypermedia System (AIHS), and test and quiz systems are among the educational computer-based systems [2]. In particular, due to their communication and collaboration strength, LMSs are becoming much more common in universities, colleges, schools, and businesses, and are even used by individual instructors to add web technology to their courses and supplement traditional face-to-face courses. LMSs gather a massive amount of students' data for analyzing their behavior, and could create a gold mine of educational data [3]. These systems capture any student activities such as reading, writing, taking tests, performing various tasks, and even communicating with peers. These systems provide a database that stores the users (profile information), their academic data, and so on.

However, due to the high quantity of information that is generated daily by the student's activities, it is difficult for a teacher to interpret and analyze the behavior of each student. Difficulties in data management and unfamiliar knowledge discovered from large volumes of data and/or news have revealed the necessity of a specific tool that can help educators to thoroughly track all the learners' activities and evaluate the structure and contents of the course and its effectiveness in the learning process. The use of data mining is a promising area to solve these weaknesses. Data mining or knowledge discovery in databases (KDD) is the automatic extraction of implicit and interesting patterns from large data collections [4]. Data mining has been used in different areas such as diagnosing heart diseases [5], text-mining [6], designing software architecture [7] [8] [9], and selecting design pattern [10] [11]. In this area, several computing paradigms converge such as decision tree construction, rule induction, artificial neural networks, instance-based learning, Bayesian learning, logic programming, and statistical algorithms. In this area, several computing paradigms converge such as decision tree construction, rule induction, artificial neural networks, instance-based learning, Bayesian learning, logic programming, and statistical algorithms, to name a few. Moreover, the most useful data mining tasks and methods include statistics, visualization, clustering, classification, association rule mining, sequential pattern mining, text mining, etc. [12]. Educational Data Mining (EDM) is a new research field associated with developing, researching, and applying computerized methods for exploration and exploitation of unique data types that come from educational environments and their use to better understand the learner's progress as well as detecting patterns in large collections of educational data that would otherwise be difficult or impossible to analyze due to the enormous volume of data [13]. The large quantity of complex educational data that come in different formats requires the use of special data interpretation and processing methods. Data mining has proved to be very useful in this context, presented in the following review papers: [13] [14] [15]. Dealing with assessment of students' learning performance, providing learning recommendation based on students' behavior, providing feedback for students, teachers and system administrators, and detection of students' abnormal educational behavior are among the problems facing the educational environment. A majority of these problems could be solved by the prediction and classification of students' performance. Prediction models and classification are prominent in this respect. The main goal of classification is to learn a model that can be used to predict the class value (called class label) with given attributes. In e-learning datasets, classification can be used for discovering the students' final performance, detecting students with dropout probability, and identifying learners with low motivation among others [13]. Several studies have been conducted to predict the students' performance. Barber et al. [16] have used the logistic regression technique for this purpose. Roberge et al. [17] have also used linear regression based on log data to predict the students' final results. Myller et al. [18] have used linear regression to predict the students' exam results (pass or fail). Kotsiantis and Pintelas [19] have performed a basic study based on relationships rather than model development, where the mathematical format of these models and analytical expertise are required to interpret, which is a difficult task for many users. Calvo-Flores et al. [20] have used neural network models for prediction of the students' failing or passing grade. A disadvantage of the neural network models for prediction of the students' final results is that they look like difficult to implement "black-box" that is problematic for a teacher to interpret and understand in order to provide individualized feedback to students. Pardos et al. [21] have employed Bayesian networks to model the user knowledge and predict the students' success. Bayesian networks including Naïve Bayes classifiers are difficult to understand for the end users (teachers and students). In contrast, the "if-then" rule model presentation format is more understandable than the others.

In this paper, we present a novel method based on the meta-heuristic PSO (Particle Swarm Optimization) algorithm in a discrete space called S3PSO (Students' Performance Prediction based on Particle Swarm Optimization) for rule induction to predict a student's final results. The main innovation of S3PSO is to consider a control part in addition to the parametric part of particles in order to control the presence or absence of features in identified rules. This capacity enables a more accurate extraction for this method. Enabling or disabling features in their particles (identified rules) and new optimized fitness function to consider comprehensibility of each rule are the main advantages of the proposed S3PSO method. S3PSO uses integer-based representation to encode classification rules in each particle, and searches the space to find the correct and accurate rules. In order to increase the interpretability and comprehensibility of each identified rule, data discretization has been used at the initial steps. The S3PSO particles encode inducts more interpretable even for normal users like instructors. Additionally, in S3PSO, important and impressive features could be kept in the rules in order to increase the rules' interest based on recommendations of the educational system administrators. In S3PSO, Support, Confidence, and Comprehensibility criteria are used to find the fitness of the rules, which are new ideas that help our final rules with a high precision and quality. We compare the results obtained by S3PSO with other rule-based classification methods like ID3, CART, and C4.5. Accordingly, the results obtained reveal that S3PSO is far better than ID3, CART, and C4.5, given three measurements evaluated by the rules obtained. Moreover, we compare S3PSO with other classification methods according to the classification measurements such as Accuracy. The results obtained reveal that S3PSO outperforms other classification methods such as SVM and neural networks based on the classification measurements.

This paper is arranged in the following order: Section 2 describes the background of the educational systems, educational data mining methods, and PSO algorithm; Section 3 describes a brief review of the related works; Section 4 describes the details of the proposed S3PSO method and its features; Section 5 presents the classification criteria, implementations, experimental set-ups, and the results obtained by S3PSO in comparison to the other classification methods; and finally, the conclusion is presented in Section 6.

2. Background

In this section, we describe the learning management systems, classification problem, association rule mining, and PSO algorithm as backgrounds.

2.1. Learning management systems

Educational computer-based systems include LMSs, ITSs, AIHSs, test and quiz systems, and the other types that are described in table 1.

Examples of LMSs include Edmodo, Blackboard, Schoology, Moodle, etc. Moodle is one of the most commonly used e-learning methods, which is free and powerful in the creation of flexible and engaging online courses and is the base system used in this paper. Moodle is an open source virtual learning system developed by programmers from all over the world, which enables the educators to create an effective online learning community and is thus a dynamic and evolving system. This system has also been used in many universities and institutions all over the world to add the web technology to their course and use its benefits in their traditional face-to-face course. Moodle offers various channels and workspaces such as forums and file storage media to facilitate information sharing between the participants in the course. This system can be used from any place even in organizational intranet. As we chose Moodle in our research work, we extracted rules from students' usage data of Moodle system. Like any other e-learning system, Moodle captures and saves all the students' usage data in a database. The instructors can use the students' log files to determine every single detail of their activities in the course. Moodle uses a feature that enables teachers to get full reports of individual or total students' activities for a specific activity. It keeps this data in a single database such as MySQL or PostgreSQL. In this system, an administrator is in charge of managing virtual course classrooms and users (teachers, students, etc.) to whom it assigns permits for different tool resources. There is a difference between the viewpoints of different users with respect to systems (administrator, students, and teachers). In figure 1, a view of virtual classrooms of Moodle is shown. The Moodle data includes all the student's interactions with an educational system (such as input in quizzes and interactive exercises), student's collaborations (such as text chat), demographic data (such as student's grade), student's affectivity (such as motivation and emotional state), etc.

2.2. Classification problem

A supervised classification provides a collection of labeled patterns. Actually, the goal of these methods is to learn a model to predict the class value, taking into account other attribute values. In classifiers, the predicted variable can be either a binary or a categorical variable [22]. Decision tree, random forest, and decision rules are the commonly used classification methods in EDM. A decision tree is a set of conditions [23] containing zero or more internal nodes as well as one or more leaf nodes. All internal nodes have two or more child nodes, and each leaf node or external node represents a class label. The paths from root of each one of the leaves represent the classification rules.

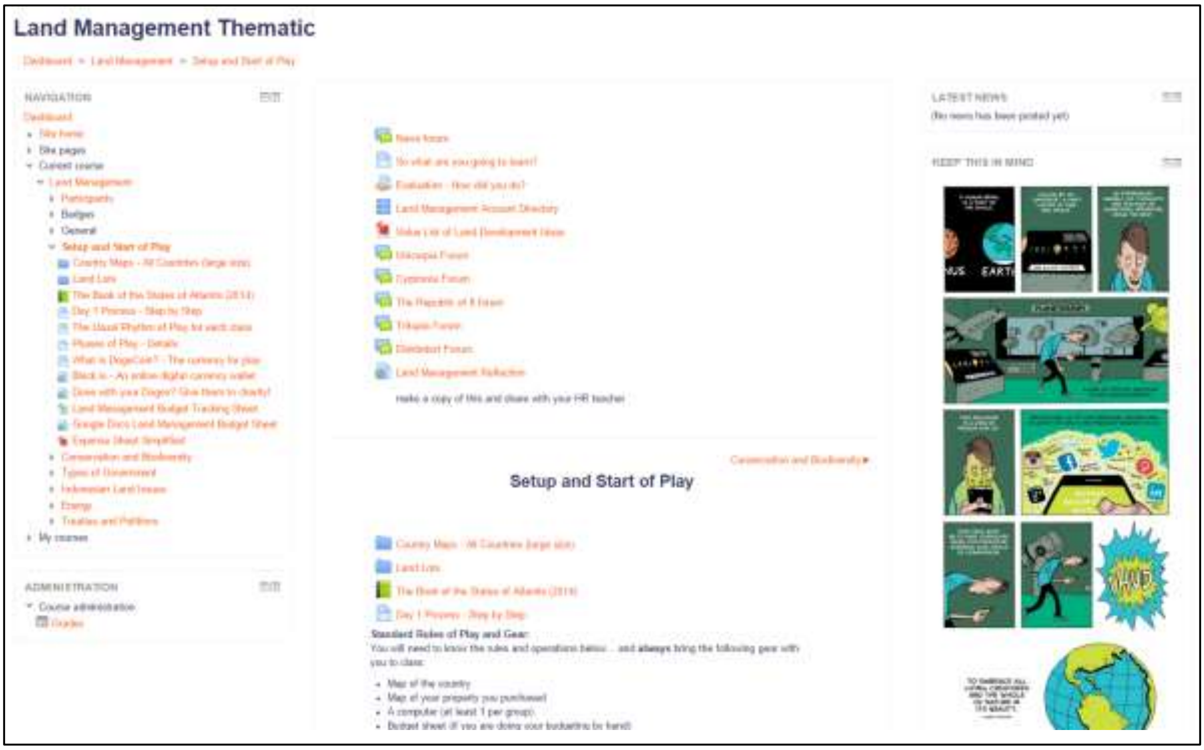

Figure 1. View of Moodle virtual classroom.

There is an easy way to convert the decision tree into a set of classification rules. C4.5, CART, and ID3 are the most well-known decision tree methods. ID3 is invented to generate decision from a dataset. C4.5 also builds decision trees from a set of data for data classification. The generated decision tree in this method is used as a classification rule. Moreover, in EDM, the predicted model is typically validated using cross-validation, where a part of dataset is used systematically for model extraction and the other part is used for model testing. One iteration of cross-validation will partition the data into complementary subsets, on some of which testing and analysis are performed and the extracted model will be validated on the other subsets considered as the testing set. *Precision*, *Recall* [24], *Accuracy* [25], and *F_1-measure* [26] are the common measurements used for classifiers. Accuracy criterion, often popular in other fields, is not sensitive to base rates and should only be used if base rates are also reported. In this work, our purpose was to induct the "if-then" rules. In the consequent part of the rule, the predicted value for class assigns a data instance to the class pointed out by the predicted value if the consequent attribute satisfies the conditions mentioned in the antecedent, and therefore, a classifier is represented as a rule set.

2.3. Association rule mining

The main objective of association rule mining is to extract the important correlation, frequent pattern, class of important regularities, and association or casual association among the set of items in the transactional and relational database or other information repositories. Association Rule Mining (ARM) was first introduced in [27], which describes the association rule mining methods and the importance of rule interestingness measure. It also points out that hidden relationships exist between purchased items in transactional database. Apriori is the most representative association rule mining method, which was proposed in [27] to mine large itemsets to figure out the association among the items. This method repeatedly generates candidate itemsets and uses minimal confidence and support for filtering these candidate itemsets to find high-frequency itemsets. After generation of the frequent itemsets in Apriori, the association rules can be produced. Whenever the calculated confidence of a frequent itemset becomes larger than the pre-defined minimal confidence in Apriori, its corresponding association rule can be accepted. Classification rules try to discover a set of rules to predict the class of unseen data while figuring out all rules that represent the association between the attributes in data.

Table 1. The Learning and Management System Descriptions.

System	Description
Learning Management Systems (LMSs) [13]	Type of software application for providing the administration, documentation, tracking, reporting, and delivery of electronic educational technology (called e-learning) courses or training programs. They offer a wide range of channels and workspaces for sharing information and participant's communication. These systems accumulate a vast amount of log data on the students' and teachers' activities in database and have built-in student monitoring systems.
Intelligence Tutoring Systems (ITSs) [15]	These systems model student's behavior to provide direct and customized instruction or feedback to students without intervention of human teacher. ITS system consists of a domain model, student model, and pedagogical model. The main goal of this system is to enable learning in a meaningful and effective manner using a variety of computing technologies.
Adaptive and Intelligence Hypermedia Systems (AIHSs)	These systems attempt to be more adaptive by building a model of the goals, preferences, and knowledge of each individual student and use this model to adapt to students' requirements.
Test and quiz systems	These systems aim to measure the students' level of knowledge with respect to one or more subjects using a series of questions/items and other prompts for gathering information from respondents.
Other types	Educational social networks, educational blogs, students forums, learning object repositories

A formal definition of association or classification rules is as follows: Let $I = \{i_1, i_2, \ldots, i_n\}$ be a set of (n binary or numerical attributes) items. Let D be a set of transactions, where each transaction T is a set of items such that $T \subseteq I$. Each transaction is associated with a unique identifier TID. A transaction T is said to contain X, a set of items in I, if $X \subseteq T$. A classification rule is an implication of the form "$X \rightarrow Y$", where $X \subseteq I$, $Y \subseteq I$, and $X \cap Y = \Phi$, where X is known as antecedent and Y as consequent. The rule $X \rightarrow Y$ has support s in the transaction set D, if s% of the transactions in D contain X U Y. In other words, the support measure for a rule shows the probability that X and Y hold together among all the possible presented cases. It is said that the rule $X \rightarrow Y$ holds in the transaction set D with confidence c if c% of transactions in D that contain X also contain Y. In other words, the confidence of the rule is the conditional probability that the consequent Y is true under the condition of the antecedent X.

Support: It measures the probability of items or itemsets in the given transactional database. The support indicates how often the rule holds in a set of data given by (1) and (2), where n is the total number of transactions and $n(X)$ is the number of transactions that contains the itemset X.

$$\mathrm{Support}(X) = n(x) / n \tag{1}$$

$$\mathrm{Support}(X \rightarrow Y) = n(x \cup y) / n \tag{2}$$

Confidence: This measure shows the conditional probability for a classification rule $X \rightarrow Y$, which is defined as the following equation. To measure the strength of classification or association rules and how often the consequent is true, *Confidence* is a good solution, given that the antecedent is true.

$$\mathrm{Confidence}(X \rightarrow Y) = \mathrm{Support}(X \rightarrow Y) / \mathrm{Support}(X) \tag{3}$$

Lift: Another measure of association rule mining is lift. It means the ratio of rule's *Confidence* to probability of occurrence of the consequent, which reflects positive or negative correlation of antecedent and consequence of rules and is defined as the following equation:

$$\mathrm{Lift}(X \rightarrow Y) = \mathrm{Support}(X \rightarrow Y) / ((\mathrm{Support}(X) \times \mathrm{Support}(Y)) \tag{4}$$

Note that if the lift is > 1, it shows that those two occurrences are dependent and make those rules potentially useful for predicting the consequent in future datasets.

2.4. Particle swarm optimization (PSO)

PSO is an artificial intelligence and computational technique. It has the capability of optimizing a non-linear and multi-dimensional problem, while requiring minimal parameterization usually reaches good solution efficiently and has a better convergence among the evolutionary algorithms such as Genetic Algorithm (GA). The PSO concept and algorithm were introduced in [28]. It is an evolutionary algorithm belonging to the class of swarm intelligence algorithms, which are inspired from the social dynamics and emergent behavior that arise in socially organized colonies like social behavior of bird flocking or fish schooling. The main strength of PSO is its fast convergence in comparison with many global optimization algorithms.

In PSO, each particle tries to achieve the best result by updating its position and speed according to its own past as well as information of current particle, which is best among all particles in swarm (PSO combines self-experiences with social-experiences). The basic concept of PSO is to create a swarm of particles that fly through the search space. However, this search process is not carried out entirely randomly, and a number of factors influence this process as follows: the best position visited by itself (its own best experience), the position of the best particle in its neighborhood (the social experience), and its current velocity.

If a particle takes the entire population as its neighbors, the best value for best social experience is a global best (called *Gbest*), and when it takes the smaller group as its neighbors, the best value is a local best (called *Lbest*). The performance of each particle is measured according to a pre-defined fitness function. The position and velocity of each one of the particles will be updated using the following equations:

$$v_i^{t+1} = w \times v_i^t + c_1 \times r_1 \times (Pbest_i^t - x_i^t) + c_2 \times r_2 \times (Gbest^t - x_i^t) \quad (5)$$

$$x_i^{t+1} = x_i^t + v_i^{t+1} \quad (6)$$

where, x_i^t and v_i^t are defined as the current position and the velocity of *i*th particle at iteration *t*, respectively. $Gbest^t$ and $Pbest_i^t$ are the global and personal best position coming out from particles' experience of *i*th particle during iterations 1 to *t*, respectively. *w* is the inertia weight that controls the impact of previous velocities on the current velocity. r_1 and r_2 are uniformly distributed random variables in the range [0,1] to provide casual weighting of different components participating in the particle velocity definition. c_1 and c_2 are used to control the impact of personal best and global best, respectively. After updating the velocity and position of a particle, its *Pbest* is updated according to (7).

$$Pbest_i^{t+1} = \begin{cases} x_i^{t+1} & \text{If } f(x_i^{t+1}) < f(Pbest_i^t) \\ Pbest_i^t & \text{Otherwise} \end{cases} \quad (7)$$

where, $f(x_i^{t+1}) < f(Pbest_i^t)$ means that the new position x_i^{t+1} is better than the current *Pbest* of the *i*th particle. After updating the velocity, position and *Pbest* of all particles, the particle with the best fitness is selected as $Gbest^{t+1}$. These operations are repeated until a termination criterion is met (e.g. the number of iterations is performed or the adequate fitness is reached).

3. Related works

A wide range of EDM methods have appeared in the last several years. Some applications of EDM are abridged in the following [29]:

- Predicting student final grade
- Detecting student behavior pattern
- Provisioning information for teacher
- Around what is going on in the classroom
- Modeling students including their behavior
- Visualizing educational data
- Recommend a learning strategy for students
- Clustering students
- Analyzing educational systems
- Predicting student's final result in their academicals time
- Improving academic performance and quality of educational process

The main methods frequently used by the EDM community are as follow: 1) Prediction models, 2) Structure discovery, 3) Relationship Mining, and d) Discovery with models [30].

In prediction models, the main goal is to construct a model that can infer a single aspect of data from some combinations of other aspects of the data. In prediction models, the predicated outputs are educational environments or some aspects of learning process (such as the final result of system). Some types of prediction models are as follow: 1) classification, in which the predicted value is a categorical value, 2) regression, in which the predicted value is a continuous value, and 3) density estimation, in which the predicted value is a probability density function. In the last few years, many researches have worked to apply data mining methods to help instructors, students, courseware authors, administrators, and others by predicting the students' performance. Iam-On and Boongoen [31] have proposed a data transformation model based on ensemble-information clustering, which is used for classification in data mining to predict the students' performance but the time complexity of this method is a disadvantage of it, which is not scaled up well for large databases. Agudo-Peregrina [32] used three classification methods to determine the existing relationships between different interactions with students' academic performance. In [33], clustering and association discovery that is based upon CRISP-DM methodology (CRoss Industry Standard Process for Data Mining) and Growth algorithm for association rule mining are used to extract knowledge form students' questionnaires. However, this research work did not include the identification of all the processes involved in providing education in higher education. Lara et al. [29] have tried to build a historical reference model of students to predict and to discover knowledge based on the academic data gathered from the Moodle platform. They have used PESFAM (Probabilistic Ensemble SFAM), which combines a number of simplified fuzzy ARTMAP (SFAM) modules (self-organizing neural networks) with a plurality voting strategy.

Romero et al. [13] have also used classification and clustering prediction of students' final performance and compared different data mining classification methods and association rule mining methods for their usefulness and performance on Moodle datasets but the lack of comprehensibility measure for each classification model is the main disadvantage of this research work. Xing et al. [34] have combined learning analytics approaches, EDM, and HCI (Human Computer Interaction) theory to predict the students' performance. This research work examined small datasets, and the qualitative aspects of collaborative work was considered in construction measure to a lesser extent. Rachburee et al. [35] have used five methods to classify the students' performance, and their results revealed that neural network had the highest accuracy among decision tree, naïve Bayses, KNN (K-Nearest Neighbor), and SVM (Support Vector Machine). Guruler and Istanbullu [36] have combined CRISP-DM with PDCA (Plan, Do, Check, Act) cycle and also used decision tree for generating rules. These research works have implemented the data mining methods in educational datasets for educational purposes such as final performance prediction, and students' modeling.

Latent knowledge estimation is a case of classification that is particularly important in EDM. In this field, the knowledge of students in specific skills and concepts is recognized by their patterns of correctness on those concepts [30]. In this EDM method, the knowledge that is not directly measurable must be inferred from the student's activities. Inference of students' knowledge can be very useful for several goals like deciding when to advance a student in a curriculum.

However, as discussed earlier, the neural networks and SVM models are like black boxes difficult to understand and interpret for teachers. This disadvantage has led us to use methods that can be easily understood.

Due to the increasing number of research works in meta-heuristic algorithms such as ant colony and GA, they have been used in association with rule mining. These studies have demonstrated that such an integration can improve the efficiency of traditional association rule mining algorithms and discover more accurate rules. However, there are still some problems. The requirement of setting parameters (like cross-over and mutation) makes the procedure more complicated. Although GA discovers high-level prediction rules, it tends to increase the computational complexity, whereas PSO equally finds a better solution in a shorter period of time (convergence speed). To make the rule-based classification, which is a special version of association rule mining procedure free from mutation and cross-over to improve time efficiency, PSO algorithm has been used in this field. Indira and Kanmani [37] have proposed an adaptive PSO (APSO) for association rule mining and Kuo et al. [38] have used PSO for association rule mining. Tyagi and Bharadwaj [39] have proposed MOPSO-DAR (Multi-Objective Particle Swarm Optimization based Direct Association Rule mining) in the domain of CF (collaborative filtering). Sousa et al. [45] have used the original PSO algorithm in classification rule discovery and performed comparisons between PSO, GA, and C4.5.

In [46], a literature survey on the PSO algorithm and its variants to clustering high-dimensional data is presented. Additionally, in [47], the applications of multi objective PSO in miscellaneous areas are reviewed. The Combinatorial problems are real world problems with discrete choices, in which the solutions are represented as integers. One of these problems is the 0/1 Knapsack problem. In order to solve combinatorial problems, several discrete PSOs are presented. In [48], a discrete PSO called COMPSO (Combinatorial PSO) is presented to solve the 0/1 Knapsack problem. According to the evaluation results in this research work, COMPSO outperforms other evolutionary algorithms like GA.

4. Proposed S3PSO method

As already discussed, the PSO algorithm is an extension of the evolutionary algorithm that finds an equal solution to other algorithms in a shorter period of time (the convergence speed of PSO is less than others). Evolutionary computing algorithms like PSO provide solutions with reducing the number of passes over the database and efficiently sustaining the search space. The original edition of PSO algorithm has been used for continuous space, and PSO is basically developed for continuous optimization problems. In a previous work [40], we proposed a discrete PSO algorithm called CPSOII to solve partitional clustering problems, and the results obtained revealed that CPSOII outperforms other PSO and GA algorithms in partitional clustering problems. Due to the advantages of PSO in association rule mining (as discussed in Section 3), in this paper, we propose a novel rule-based classification method based on CPSOII idea known as *S3PSO* (*Students' Performance Prediction based on Particle Swarm Optimization*) in order to extract the hidden rules that could be used to predict the final outcome of students. *S3PSO* has been designed to induct the classification rules through the discrete data. The classification rules are encoded in particles, which means that each particle represents a specific classification rule. The *S3PSO* method is an extension of the original PSO algorithm but essentially differs from the original (or continuous) PSO in two features: particle design and computational operators in equations of velocity and position.

The general process of *S3PSO* is given in figure 2. The *S3PSO* method consists of eight steps. This method uses the student's data extracted from an educational resource as input. The first step includes two sub-steps: 1) discretizing all the continuous numerical values of the input data according to the number of bins, and 2) randomly partitioning the input data into 10 equal-sized folds. Among these 10 folds, a single one is considered as the test dataset and the remaining 9 folds are used as the training dataset. At the second step, the algorithm parameters such as swarm size, maximum number of iterations, the parameters used in velocity equation, and the parameters P_1, P_2, and P_3 in fitness function are initialized. At the third step, a random initial population is generated, and at the fourth and fifth steps, the fitness function for each one of the particles is evaluated and the best particle is set as *Gbest*. The sixth step includes 5 sub-steps that will be repeated until the constraints are satisfied or the maximum number of iterations are exceeded (step 6.5 of *S3PSO*). At first, *S3PSO* computes the velocity of particles, and the particles are then moved to new positions. After updating the velocity and position of each particle, its fitness is again computed to update the global best and local best. At the seventh step, the method saves the extracted classification rules. Finally, the method classifies the input test data using the classification rules.

In the above-mentioned steps, step 1 is executed only once but steps 2-8 are repeated 10 times for each one of the 10 folds. The main concepts of *S3PSO* and its steps are described in the following sections.

Any heuristic search is specified by two basic behaviors: exploration and exploitation. There is a trade-off between these concepts if exploitation of a search is increased, then exploration decreases, and vice versa. The control parameters in PSO are defined to make a balance between exploration and exploitation. These parameters have a significant influence on the performance of the methods. The parameters involved in PSO are given in table 2.

Table 2. Parameters involved in PSO.

Parameter	Parameter role
Inertia weight (*w*)	Controlling the impact of the velocity history into the new velocity
Acceleration coefficient r_1	Maintaining the diversity of swarm
Acceleration coefficient r_2	Converging toward the global optima

It should be noted that in table 3, the S3PSO method and CPSOII [40] are compared. As shown in this table, the main similarity of both is the use of discrete PSO, i.e. steps of PSO algorithm, particle encoding, swarm initialization, particle moving, and termination conditions.

4.1. Particle encoding in *S3PSO*

In the original PSO, a real-based representation is employed to encode each particle. Analogously, *S3PSO* uses the integer-based representation to encode particles, and it adds control attributes to the particles. In *S3PSO*, each particle consists of two types of attributes: the control and parametric attributes. Control attributes determine which parametric attribute should be utilized and which one can be disabled. In general, the value of each control attribute is binary, while the value of each parametric attribute is an integer. In fact, the goal of control attribute is to enable or disable the parametric attributes. If the value of a control attribute equals to 1, then the associated parametric attribute is enabled; otherwise, the other one is disabled. Figure 3 shows an example of the *S3PSO* particle representation.

In figure 3, a two-level particle presentation is shown, in which the attributes 12, 21, and 14 are enabled, and the others are disabled.

$$\left[\overbrace{0\quad 1\quad 0\quad 1\quad 1\quad 0}^{\text{Control attributes}} :: \overbrace{4\quad 12\quad 9\quad 21\quad 14\quad 5}^{\text{Parametric attributes}}\right]$$

Figure 3. An example of the *S3PSO* particle representation.

With S3PSO encoding, the X_i position of each particle will simply be a vector that provides the numerical value of each attribute. The V_i velocity of each particle will be a combination of integer and binary numbers vector, representing the recent move. In this paper, we represent the swarm as a set of *M* particles.

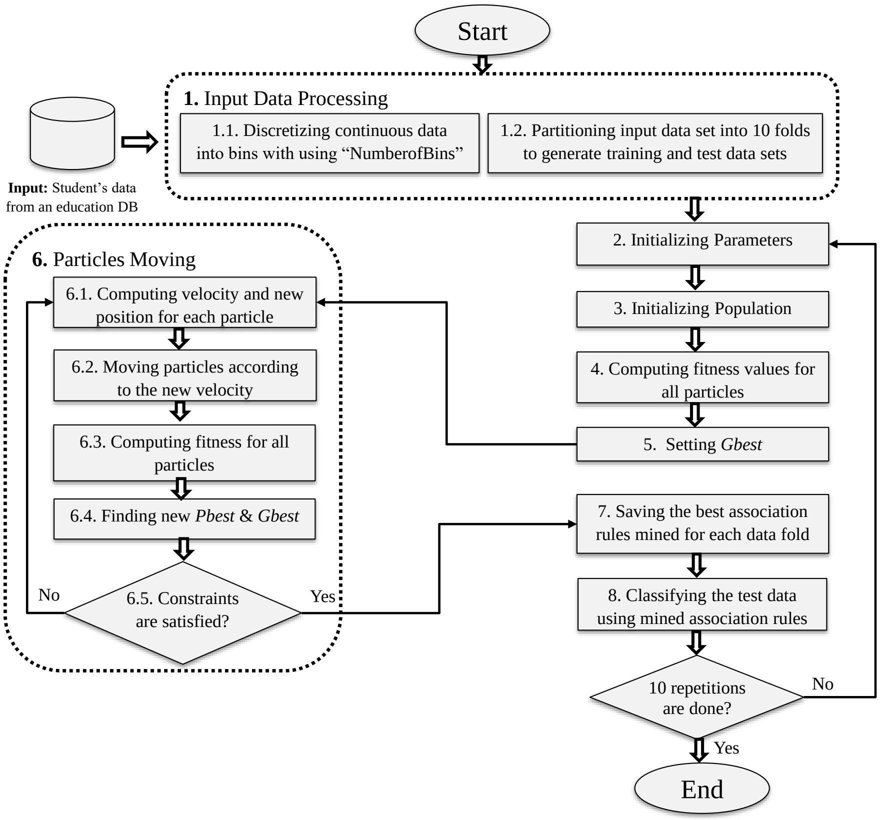

Figure 2. General process of the proposed *S3PSO* method.

Table 3. Comparison between S3PSO and CPSOII [40].

Metric	S3PSO	CPSOII [40]
Goal	Rule-based classification	Clustering
Method	Discrete PSO	Discrete PSO
Input Data	A discretization of continuous values	Integer values
Particle Encoding	Control and Parametric attributes + Label (For classification)	Label-based Integer encoding
Fitness Function	Support, Confidence, and Comprehensibility	Sum of Squared Error (SSE), Variance Ratio Criterion (VRC) and Davies-Bouldin Index (DBI) [40]

Each particle is determined with its position, velocity, and *Pbest*. Each particle position X_i and particle velocity V_i, $i = 1.,..M$ is characterized by $2N$ elements, x_{i1}, x_{i2} , … , x_{ij} , …., x_{i2N}, where N is the number of student's attribute, x_{ij} represents the value of jth attribute in ith particle, and v_{ij} represents the recent move of jth attribute in ith particle. A representation of the particle position and velocity is shown in figure 4. It should be noted that k_{ij} is the value of jth attribute in ith particle, and is assumed to lie in the range of [K_{min} , K_{max}], where this value is defined in the range of attribute intervals. According to definition of association or classification rules, the rules are the implication of "if-then" rules defined in the "X -> Y" form, where X and Y are the itemsets, and the intersection of association rule of itemset X to itemset Y (X→Y) must be empty. Items that appear in itemset X do not appear in itemset Y, and vice versa. Figure 5 shows an example of encoding format of classification rules in the particles.

As mentioned earlier, each value in the parametric part of the particle represents the value of attribute in its interval. For these attributes, we chose Moodle, which, similar to most LMS systems, records all students' usage information not only in a log file but also directly in a database. We used MySQL for our objective in the Moodle system. Moodle database has about 145 inter-related tables but not all these tables are required for our research work. In fact, these tables are raw and need to be clean and transformed into an appropriate format for our purpose (inducting classification rules). This transformation converts the original data into a suitable shape. Data extraction from Moodle database has to be done when the instructor or courseware authors have enough information about the students, which is possible after the end of the semester or course. Therefore, they can choose between creating a summary data file from different Moodle tables to view the data in an appropriate way. The summarization process integrates the most important students' information from students' and their interactions' logs over several Moodle tables (see Table 4). The process of summarization needs to create a new table (mdl_summarization) in our Moodle database. Data of students and their interactions, which are spread over several tables, should be gathered into a summary table for our purpose. This table has summary per row about all the activities done by each student in the course as well as the final mark obtained by the students. In order to collect information about suitable students and course, we should make some queries to the database including three steps. At the first step, the instructor chooses a specific course, and then the users (each student) have to add the final marks in the second step and can basically filter the attribute for their importance to be used in summarization file at the third step. After generating the summarization file, we can use it as the input source of *S3PSO*.

As discussed earlier, we encode classification rules in the particle and map each of the particles' attributes to input data features. An example of a particle is illustrated in figure 6. This particle has ten different attributes, indicating that several possible classification rules can be generated. Table 4 summarizes students and their important interaction attributes and the final mark obtained in this course. Although many factors can affect the effectiveness in the e-learning, we use the attributes already defined in table 4 for encoding our particle and each of these attributes is considered as a dimension in particle. Note that in figure 6, the resulting attribute is a Boolean one that can predict the students' success or failure.

4.2. Data discretization

It was necessary to perform a discretization of continuous values in order to increase the interpretation and comprehensibility of rules. Discretization divides the continuous numerical data into categorical classes that are easier to understand for the teacher (categorical values are more user friendly for the teacher than precise magnitudes and intervals). We discretized all the continuous numerical values of the attributes of input data. There are a number of unsupervised methods to transform attributes in continuous space into discrete space [41] such as equal-width method (which divides the range of the attribute into a fixed number of intervals of equal length), equal-frequency method (which divides the range of the attribute into a fixed number of intervals with the same or approximately the same number of instances in it) or the manual method (in which the user has to specify the cut-off points). In this case, we used the manual method to discretize continuous numerical attributes. The mark attribute is a real number. We discrete it into some intervals using the following equation (8).

$$\text{Range} = (\text{Max} - \text{Min}) \,/\, \text{NumOfBin} \qquad (8)$$

There are different discretization methods. Concretely, the manual method (where cut-off points have to be specified) was applied to the attributes of input data. Table 5 shows the interval discretization of attributes presented in table 4.

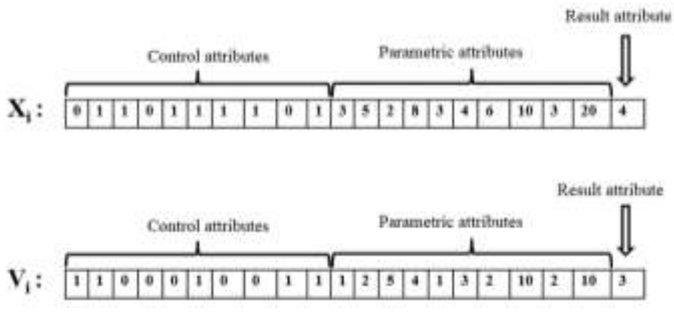

Figure 4. Examples of the *S3PSO* particle position and velocity.

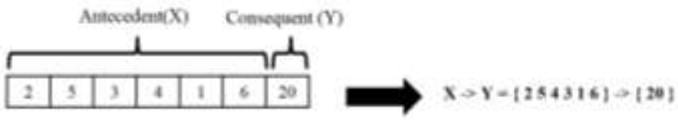

Figure 5. Encoding format of classification rules in the particles.

Table 4. Students' attributes in summary file in Moodle system.

No.	Name	Abbreviation	Type	Description
1	*course*	CIN	Input attribute	Course identification number
2	*n_assignment*	NOA	Input attribute	Number of assignments done
3	*n_quiz_a*	NQP	Input attribute	Number of quizzes passed
4	*n_quiz_s*	NQF	Input attribute	Number of quizzes failed
5	*n_posts*	NMS	Input attribute	Number of messages send to forum
6	*n_read*	NMR	Input attribute	Number of messages read on the forums
7	*total_time_assignments*	TTA	Input attribute	Total time used on assignments
8	*total_time_quiz*	TTQ	Input attribute	Total time used on quizzes
9	*total_time_forum*	TTF	Input attribute	Total time used on forum
10	*mark*	MRK	Input attribute	Final mark the student obtained in the course

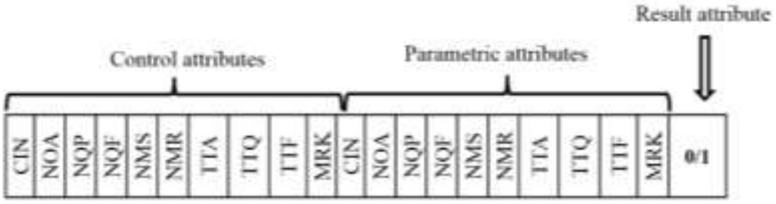

Figure 6. Labels of particle attributes.

4.3. 10-Fold cross-validation

Cross-validation is a method for measuring how the result of an analysis is generalized to an independent dataset. In *10*-fold cross-validation, which is used in step 1.2 of the *S3PSO* method, as shown in figure 2, the original data is randomly partitioned into 10 equal sized sub-folds. One of the 10 sub-folds was retained as the validation data for testing the model, and the remaining 9 sub-folds were used as training data. The cross-validation process was then repeated 10 times, with each of the 10 sub-folds used exactly once as the validation data.

4.4. Swarm initialization

Random uniform initialization is the most common technique for initialization population, in which each particle of the initial swarm and consequently the initial best positions are drawn by sampling a uniform distribution over the search space. In order to apply the evolution process of *S3PSO* method in step, 3 as shown in figure 2, it was necessary to first generate the initial population. Therefore, the position of each particle was initialized with a uniformly distributed random vector with K_{min} and K_{max} for each specific attribute in the search space. After generating the initial particles of swarm, initial velocity of all particles was assigned to zero and the initial *Pbest* of each particle was assigned to its current position.

4.5. Fitness value calculation

In this paper, we present another measure to quantify the understandability of the rule by the number of attributes involved in the rule. Note that the lower the number of conditions involved in antecedent part of a rule, the more comprehensible the rule [44]. Therefore, the following equation was used to quantify the comprehensibility of a rule.

$$\text{Comprehensibility}(X \rightarrow Y) = 1/(\ln(2 + |X|)) \qquad (9)$$

where, N is the number of attributes used, which is equal to 10 according to table 4. The optimality of the particle is determined by calculating the fitness value of the particle. The fitness value was derived from the fitness function. The objective of the fitness value is extracting efficient and accurate rules based on *Support* (Equation 2), *Confidence* (Equation 3), and *Comprehensibility* values. There are three parameters defined as P_1, P_2, and P_3 that control the importance and impact of *Support*, *Confidence*, and *Comprehensibility* of a rule based on the value of fitness function.

Table 5. Attributes interval discretization

Interval discretization	Label
x<4	1
4<x<8	2
8<x<12	3
12<x<16	4
x>16	5
x: Mark attribute	
y<10	1
10<y<20	2
20<y<30	3
30<y<40	4
40<y<50	5
50<y<60	6
y>60	7
y: NOA (*n_assignment*), NQP (*n_quiz_a*), NQF (*n_quiz_s*), NMS (*n_posts*), and NMR (*n_read*) attributes	
z<1 hour	1
1 hour <z< 3 hours	2
3 hours <z< 5 hours	3
5 hours <z< 7 hours	4
z>7 hours	5
z: TTA (*total_time_assignments*), TTQ (*total_time_quiz*), and TTF (*total_time_forum*) attributes	

We defined a customized fitness function to trade-off among *Support*, *Confidence* and *Comprehensibility* and find out the rules with a higher quality (see Equation 10).

$$\text{Fitness}\ (X \rightarrow Y) = P_1 \times (\text{Support}(X \rightarrow Y)) + P_2 \times (\text{Confidence}(X \rightarrow Y)) + P_3 \times (\text{Comprehensibility}(X \rightarrow Y)) \quad (10)$$

The sum of P_1, P_2, and P_3 is equal to 1. In fact, we could control the impact of *Support*, *Confidence*, and *Comprehensibility* by changing these weight parameters in the interval of zero to one. We used PSO as a module to mine the best fitness value, which is the highest fitness value in the fourth step of *S3PSO* shown in figure 2.
According to the defined fitness function, the position of each particle will be estimated. The current position of the particle will be updated, if the estimated fitness value for the current position is better than the previous positions' fitness value. If the calculated fitness value for the position is higher than its history, then the best particle position will be updated to the current position. Moreover, the maximum fitness value obtained by the particles will be considered as "*Gbest*" in the fifth step of *S3PSO* shown in figure 2.

4.6. Velocity computation and particle position movement

There is a velocity vector that drives the optimization process in PSO algorithms and their extension. In *S3PSO* (in the step 6.1 shown in Figure 2), the velocity equation of each particle is modified by (11).

$$V_i^{t+1} = W \otimes V_i^t \oplus (R_1 \otimes (Pbest_i^t \ominus X_i^t)) \oplus (R_2 \otimes (Gbest^t \ominus X_i^t)) \quad (11)$$

where X_i^t and V_i^t are the position and velocity of particle i at iteration t, respectively. $Pbest_i^t$ and $Gbest^t$ are the best positions obtained by particle i and swarm of particles during time t, respectively. W, R_1, and R_2 are vectors comprising 0 or 1 elements such that the values for these vectors are randomly generated with w, r_1, and r_2 probability, respectively. Figure 7 illustrates samples of particle position, *Pbest* and *Gbest* particles, velocity particle, R_1, R_2, and W vectors. In the following, we explain the overall process of the movement and generation of new position with an example.
Note that an advantage of *S3PSO* is that it can freeze some attributes among 10 attributes introduced in table 4, which are important and interesting for instructors or their educational system. This feature leads to finding the rules based only on the attributes that are interesting for *S3PSO* user. Therefore, when instructors or their educational systems ask to keep some attributes in the obtained rules, *S3PSO* always enables their control attributes in all the particles; accordingly, it is able to freeze their corresponding control attributes to always hold them in all the particles.
In (11), a number of operators including *Difference, Multiply*, and *Merge* are introduced and defined as follow:
Difference operator **(⊖)*:*** This operator computes the difference between current position of *i*th particle (X_i^t) and *Pbest* or *Gbest* particles. This operator is defined in (12). Note that in difference operator, the λP_i^t and λG_i^t vectors indicate the difference between X_i^t and its $Pbest_i^t$, X_i^t and $Gbest^t$, respectively. In figure 8, a sample of performing the difference operator according to particles shown in figure 9 is illustrated.

$$\lambda P_i^t = Pbest_i^t \ominus X_i^t$$
$$\lambda p_{ij}^t = \begin{cases} pbest_{ij}^t & \text{if } x_{ij}^t \neq pbest_{ij}^t \\ 0 & \text{otherwise} \end{cases} \quad (12)$$

Multiply operator **(⊗)*:*** This operator manages the exploration and exploitation abilities of the swarm using W, R_1, and R_2 vectors. The values for these vectors are controlled by the w, r_1, and r_2 variables. Generally, the value for w controls movement of particles in the previous direction, and the values for r_1 and r_2 control the movement of particles in direction of *Pbest* and *Gbest*, respectively. The multiply operator is equivalent to multiplication of values of same indices of the two vectors.
In figure 9, a sample of performing the multiply operator according to the W, R_1, and R_2 vectors shown in figure 7 and the λP_i^t and λG_i^t vectors in figure 8 is illustrated.
Merge operator **(⊕)*:*** This operator merges two vectors of N elements, as shown in (13).
In figure 10, samples of performing the merge operator according to vectors shown in figure 9 are illustrated. In *S3PSO*, the merge operator is used to merge three velocity vectors defined in (11).

$$Z_i^t = \lambda P_i^t \oplus \lambda G_i^t \quad (13)$$
$$z_{ij}^t = \begin{cases} \lambda g_{ij}^t & \text{if } \lambda g_{ij}^t \neq 0 \text{ and } \lambda p_{ij}^t = 0 \\ \lambda p_{ij}^t & \text{if } \lambda g_{ij}^t = 0 \text{ and } \lambda p_{ij}^t \neq 0 \\ \lambda p_{ij}^t \text{ or } \lambda g_{ij}^t \text{ randomly} & \text{if } \lambda g_{ij}^t \neq 0 \text{ and } \lambda p_{ij}^t \neq 0 \\ 0 & \text{if } \lambda g_{ij}^t = 0 \text{ and } \lambda p_{ij}^t = 0 \end{cases}$$

4.7. Particle movement

After computing the velocity of a selected particle, in step 6.2 of *S3PSO* shown in figure 2, the new position of particle is generated based on the following equation (see a sample in figure 11).

$$x_{ij}^{t+1} = \begin{cases} v_{ij}^{t+1} & \text{if } v_{ij}^{t+1} \neq 0 \\ \text{a random number } m & \text{otherwise} \end{cases} \quad (14)$$

where, *m* is an integer random value that depends on the *i*th attribute. If the attribute is parametric, then *m* is uniformly distributed in the range of the corresponding attribute domain (according to Section 4.2); otherwise, if it is a control attribute, then the value is equal to 1 or 0. For example, for *mark* attribute, this random number could be an element of the set {1,2,3,4,5}. After determining the new position of each particle, the fitness function is computed for all the new particles, and then the new *Pbest*$_i$ and *Gbest* particles are found and set in step 6.4 of the *S3PSO* method.

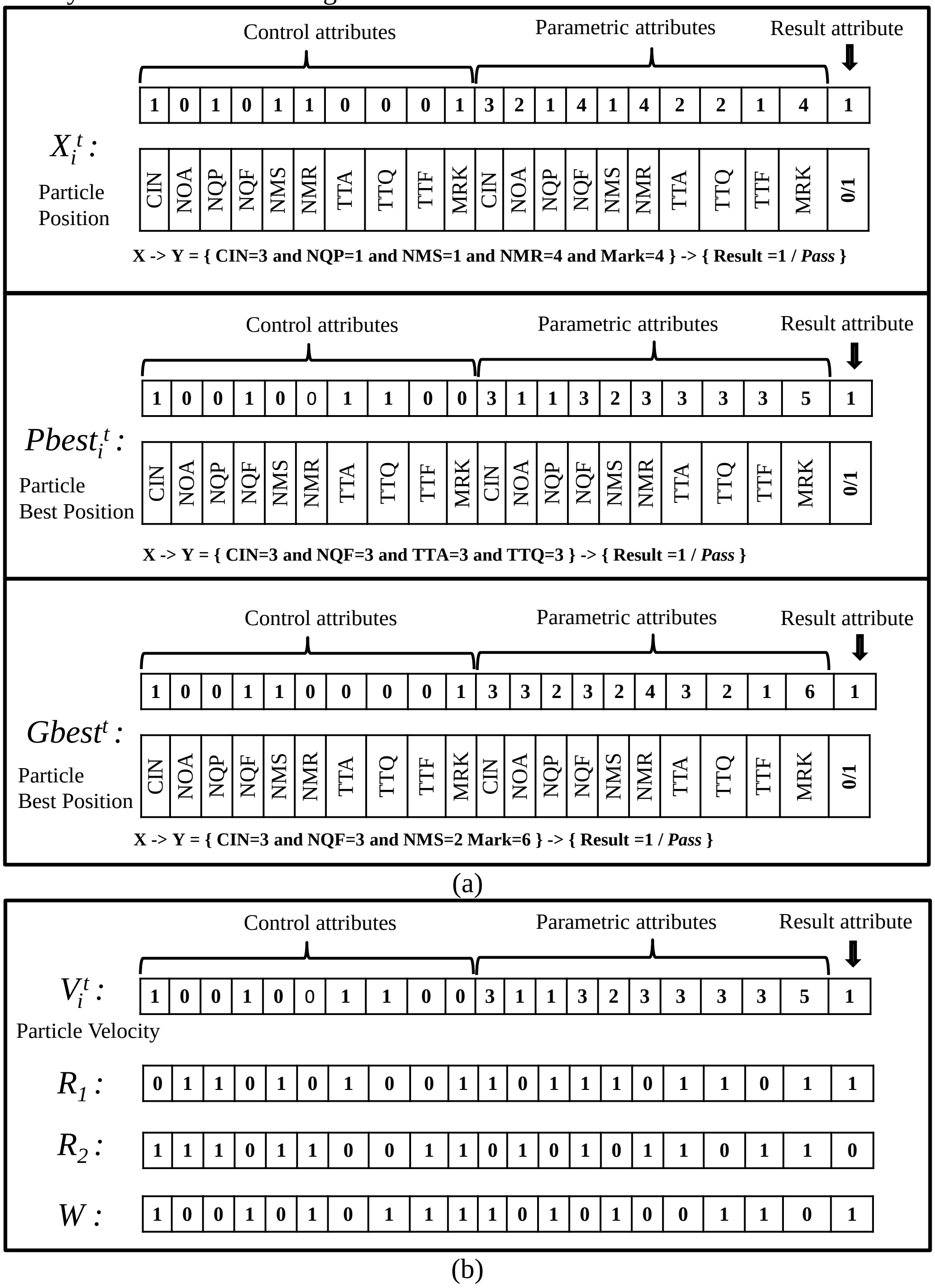

Figure 7. Samples of particle position, *Pbest* and *Gbest* particles (a), velocity particle, *R₁*, *R₂*, and *W* vectors (b).

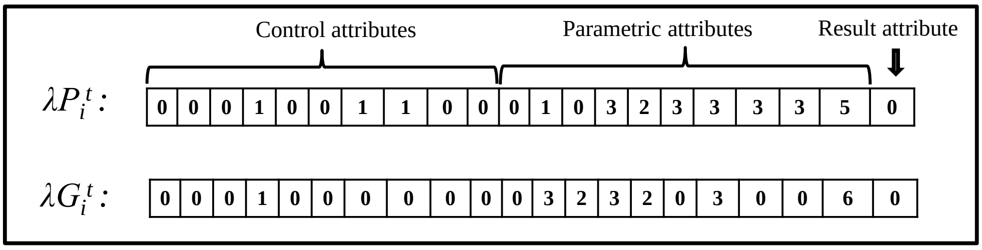

Figure 8. A sample of performing the difference operator (⊖).

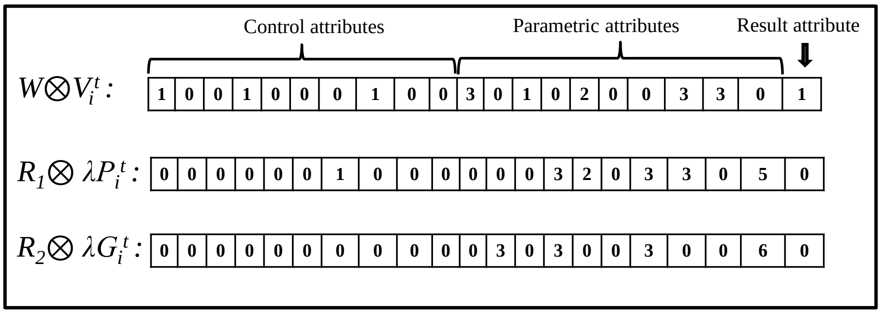

Figure 9. A sample of performing the multiply operator (⊗).

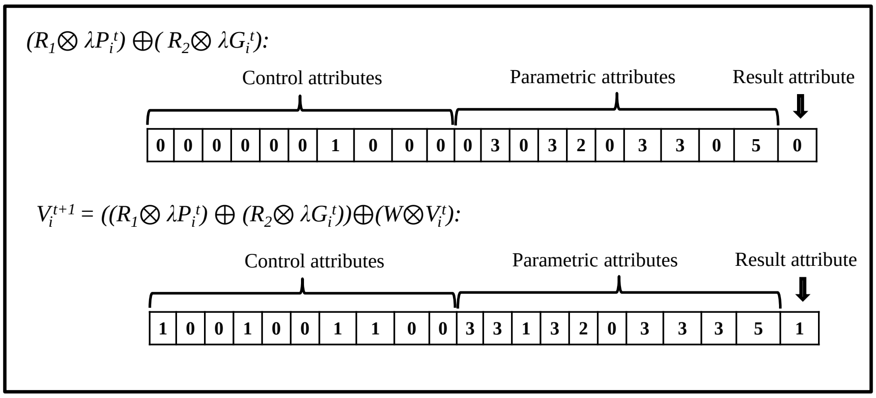

Figure 10. Samples of performing the merge operator ⊕.

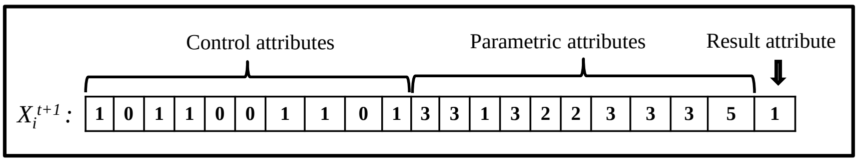

Figure 11. A sample of generating the new position of particle X_i.

4.8. Termination condition

Designing a termination condition is necessary to end the process of particle evolution. In the *S3PSO* method, the evolution is terminated when the fitness values of all particles are the same or when the number of iteration exceeds the *max number of iterations*, and the evolution of the particle swarm is then completed. Finally, after the best particle is found, its support and confidence are recommended as the values of minimal support and minimal confidence. These parameters are employed for classification rule mining to extract valuable information.

5. Experimental results

This section presents our experimental results for the proposed *S3PSO* method. We conducted experiments to evaluate the performance of *S3PSO* in classification rule induction and compared its results with other classification methods to predict student's final mark based on information in the student's usage data in Moodle defined in Section 2.1. The main goal of these experiments was to analyze the performance of the proposed *S3PSO* method. In this section, we first describe the experimental datasets and performance metrics, and study the impact of parameter setting on our work; finally, we compare the *S3PSO* method results with other classification methods.

5.1. Dataset and experimental setup

Our objective in this research work was to classify the students with similar final marks into different groups based on their activities in the Moodle system. The data in this paper was collected from 734 students studying at the Tarbiat Modares University (Tehran, Iran) in computer science courses from Moodle in the last three years.

In this section, our goal was to measure the performance of *S3PSO* as follows: 1) comparing the efficiency of its obtained rules with the efficiency of other rules obtained by C4.5, CART, and ID3, 2) comparing its classification results with those obtained by other classification methods including SVM, Naïve Bayes, K-NN, and MLP (Multi-layer Perceptron) neural network provided by Weka mining tool as an open source software providing a comprehensive collection of machine learning methods and data preprocessing tools [42].

We also had to transform *mdl_summarization* into a format required for data mining method, which is called ARFF. The ARFF (Attribute-Relation File Format) file is an ASCII text file that describes a list of instances sharing a set of attributes [43]. In S3PSO, we export the "*mdl_summarization* tables" data to the excel format and discretize it after exporting the discrete data as the CSV file. Note that we used this site [49] to convert the CSV file into an ARFF one.

S3PSO was implemented in Microsoft Visual C#.Net 2015. All experiments conducted in this work were run in windows 7 on Desktop pc with Intel Core i5, 2.4 GHz processor and 4 GB real memory.

S3PSO has multiple parameters that must be set by the user including: 1) *Swarm Size*, 2) *Max Iteration*, 3) w, 4) r_1, 5) r_2, and 6) P_1, P_2, and P_3 *probabilities in Fitness function* (Equation 10). As mentioned earlier, the values for w, r_1, and r_2 determine the exploration and exploitation abilities of the proposed *S3PSO* method. In *S3PSO*, high values for r_1 and r_2 lead to movement of particles toward *Pbest* and *Gbest*, respectively. Moreover, if the value for w is small, the particle motions will be more random, and if its value is large, the particle motions are dependent on the recent value of their velocities. In order to improve the efficiency of the *S3PSO* search process, it was necessary to determine the best value for each one of these parameters. For this reason, 50 runs of the *S3PSO* method with different values of w, r_1, and r_2 (in the range of [0, 1] with step 0.01) were performed. Therefore, the average values for the best result for 50 runs were chosen. In this experiment, other parameters were set as follow: *Swarm size* was equal to 500 particles and 1000 iterations were considered as *Max Iteration*. Figures 12 (a), (b), and (c) show the impacts of the w, r_1, and r_2 parameters on the values of fitness function, respectively. Note that in figure 12, the average results of 50 runs and their standard deviations are presented. As shown in this figure, the amounts of standard deviations are low and can be neglected. The results obtained show that the suitable value for r_1 is equal to 0.63, for r_2 is equal to 0.67, and for w is equal to 0.46, as shown in figures 12 (a), (b), and (c), respectively.

In S3PSO, according to the fitness function (Equation 10), with higher values of P_1, P_2, and P_3, the values for *Support*, *Confidence*, and *Comprehensibility* have more impacts on the fitness value for each rule, respectively. In addition, in S3PSO, the values for the P_1, P_2, and P_3 probabilities in fitness function are set according to human preferences. This means that humans can change their importance according to their preferences. For example, if according to human viewpoints the length of identified rules is so important, they can set P_3 more than P_1 and P_2. In our experiments, we assumed that all of these parameters had the same importance, and therefore, the values for all P_1, P_2, and P_3 were considered as 0.33.

5.2. Evaluation measurements

In order to evaluate the goodness of *S3PSO* and to compare *S3PSO* with other classification methods, we employed the classification performance measurements including *Precision* (*P*) (Equation 15), *Recall* (*R*) (Equation 16), *F_1-measure* (F_1 as the combination of P and R using Equation 17), and *Accuracy* (Equation 18). These criteria measure the frequency of the correct or incorrect prediction made by the classification methods. *Precision* for a class is the number of true positive (*TP*) or the number of items that have been correctly labeled (as belonging to the positive class) divided by the total number of elements that have been labeled as belonging to the positive class, actually the total number of false positives (*FP*) and true positives (*TP*), and FP denotes the number of items incorrectly labeled as belonging to that class. *Recall* is also defined as the number of correctly labeled items (*TP*) divided by the total number of correctly labeled item (*TP*) and items that have not been labeled as belonging to that class (i.e. *FN*). *Accuracy* defined by (18) is commonly used for evaluation measurement in machine learning.

In this research work, the *Precision*, *Recall*, *F₁-measure*, and *Accuracy* criteria were considered to evaluate the effectiveness of the proposed *S3PSO* method and to compare its performance with other classification methods.

$$P = \frac{\sum_{i=1}^{|c|} TPi}{\sum_{i=1}^{|c|} (TPi+FPi)} \quad (15)$$

$$R = \frac{\sum_{i=1}^{|c|} TPi}{\sum_{i=1}^{|c|} (TPi+FNi)} \quad (16)$$

$$F_1 = \frac{2 \times P \times R}{(P+R)} \quad (17)$$

$$\text{Accuracy} = \frac{TP+TN}{TP+FN+FP+FN} \quad (18)$$

5.3. Experimental results of comparisons with other classification methods

In order to examine the *S3PSO* efficiency, we first compared its result with other rule-based classification methods such as C4.5, ID3, and CART. These classification methods use a decision tree to deduce the classification rules form the tree. Then we compared the proposed *S3PSO* method with other classification methods according to *Precision* (*P*) (Equation 15), *Recall* (*R*) (Equation 16), *F_1-measure* (Equation 17), and *Accuracy* (Equation 18). Note that we used the WEKA tool to examine other methods [42].

Figure 13 shows the results of the *Support*, *Confidence*, *Comprehensibility*, and Fitness measurements for C4.5, ID3, and CART as well as the proposed *S3PSO* method on the Moodle case study. Note that the best obtained results of S3PSO on the Moodle case study for the *Support*, *Confidence*, *Comprehensibility*, and Fitness measurements were 0.79, 0.75, 0.62, and 0.72, respectively. As shown in figure 13, the proposed *S3PSO* method achieves the results far better than other rule-based classification methods for all the measurements. Note that *S3PSO* improved 31% of the value of fitness measurement for the Moodle dataset. Moreover, it should be noted that the average values of *lift* measure for C4.5, ID3, CART, and *S3PSO* were 1.72, 1.51, 1.33, and 1.58, respectively, and since all of them were more than 1, we could not make a difference among them.

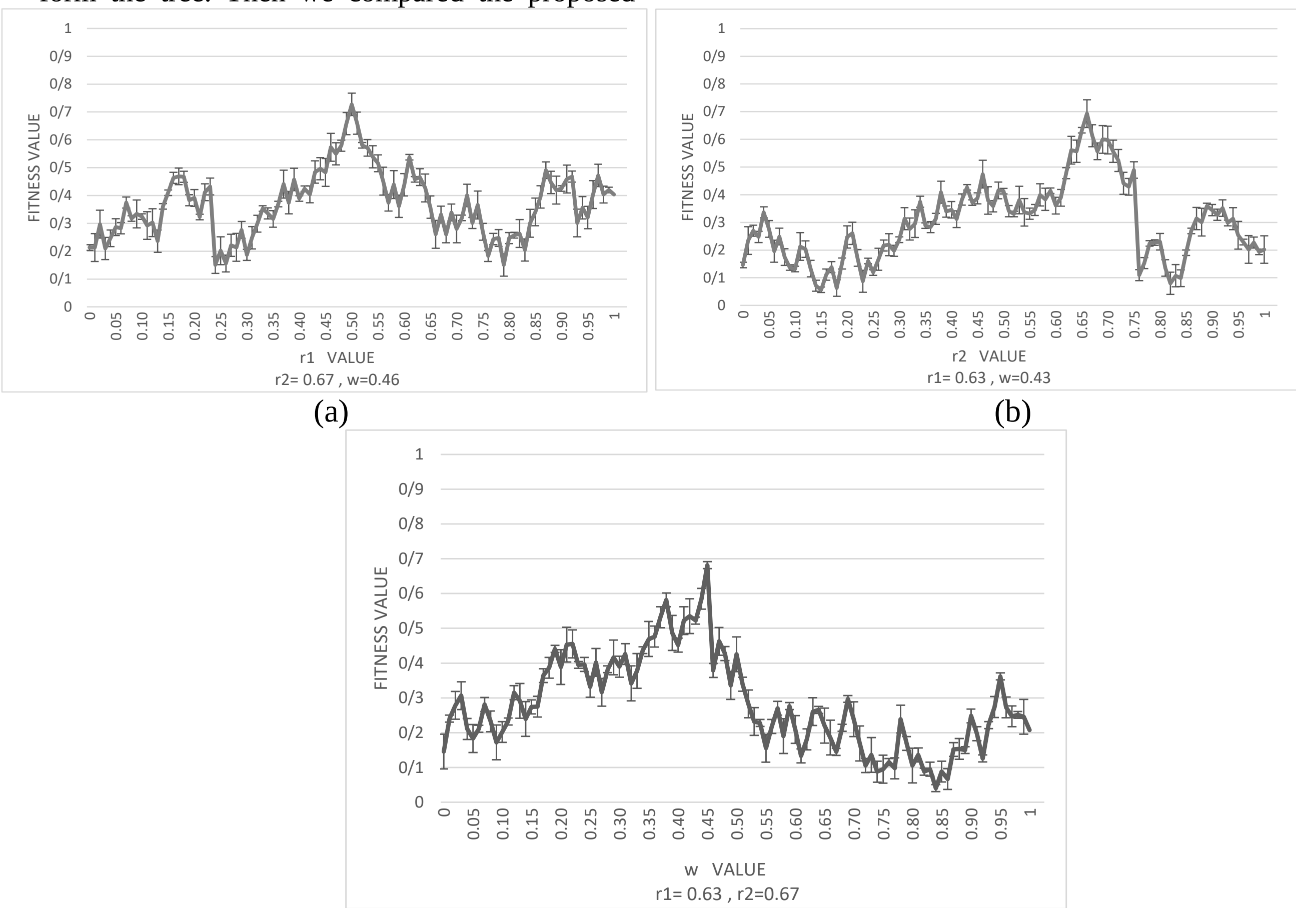

Figure 12. Effects of *w*, *r_1*, and *r_2* parameters on the values of fitness function (Average of 50 runs ± Standard deviation).

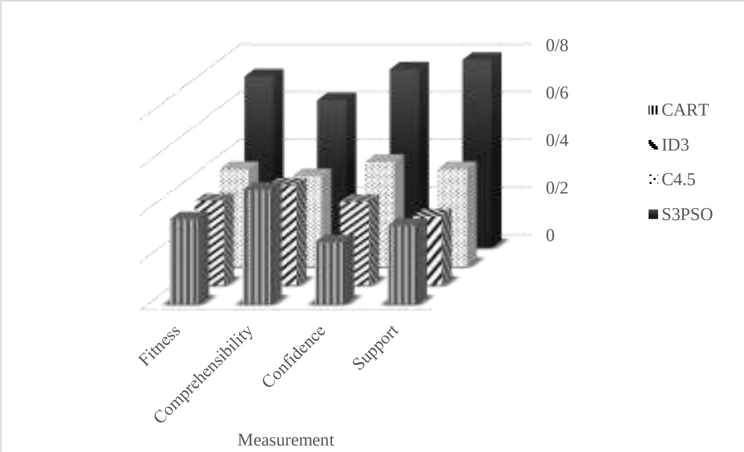

Figure 13. Results of *Support, Confidence, Comprehensibility,* and *Fitness* measurements for different methods on the used Moodle case study.

Table 6. Results of *Precision, Recall, F_1-measure,* and *Accuracy* measurements for different methods on the used Moodle case study.

Measurement: Method	*Precision*	*Recall*	*F_1-Measure*	*Accuracy*
C4.5 (J48)	0.51	0.56	0.53	0.52
CART	0.48	0.49	0.48	0.49
ID3	0.47	0.53	0.50	0.51
KNN	0.39	0.44	0.41	0.41
Naïve Bayes	0.55	0.58	0.56	0.57
SVM	0.68	0.75	0.71	0.72
APSO [37]	0.75	0.76	0.76	0.76
MLP Neural Network [35]	0.78	0.86	0.82	0.81
PESFAM [29]	0.81	0.84	0.82	0.83
S3PSO	**0.89**	**0.93**	**0.91**	**0.92**

Table 7. Examples of the rules obtained by *S3PSO.*

Measurement / *Rule*	*Support*	*Confidence*	*Comprehensibility*	*Fitness Function*
NOA=4, *NQP*=5, *TTA*=5, *MRK*=4 ==> Result (PASS)	0.88	0.86	0.56	0.77
NOA=7, *NQP*=2, *MRK*=5 ==> Result (PASS)	0.72	0.83	0.62	0.72
NOA=1, *NQP*=1, *MRK*=3 ==> Result (FAIL)	0.59	0.66	0.62	0.62
NMP=1, *NMR*=1, *TTA*=2 ==> Result (FAIL)	0.79	0.81	0.62	0.74

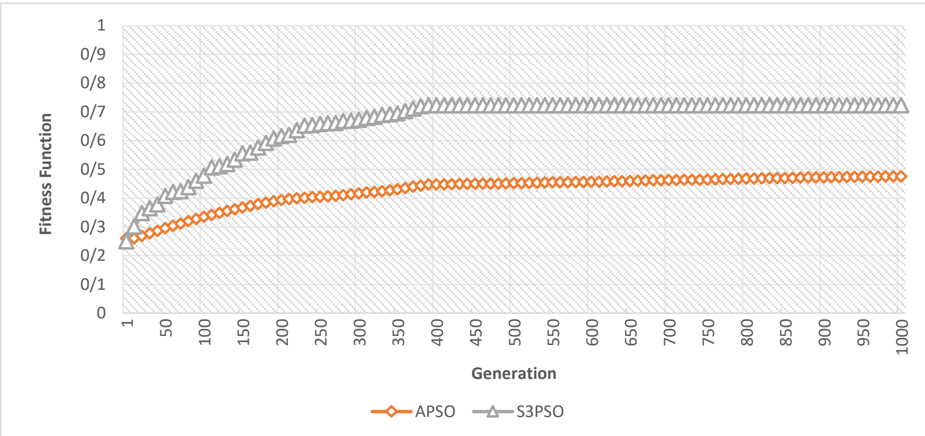

Figure 14. Comparing convergence of *S3PSO* and APSO methods.

Table 6 shows the results of several classification methods according to the *Precision*, *Recall*, *F_1-measure*, and *Accuracy* measurements for the Moodle case study defined in Section 5.2. Note that the swarm size of APSO was equal to 700 according to the suggestion presented in [37]. Furthermore, the MLP Neural Network used 3 layers according to the suggestion presented in

[35]. As shown in table 6, the proposed *S3PSO* method outperforms other classification methods including C4.5, CART, ID3, KNN, Naïve Bayes, SVM, MLP Neural Network, PESFAM, and APSO based on the *Precision*, *Recall*, *F_1-measure*, and *Accuracy* measurements. For example, it improved the value of *Accuracy* criterion for the Moodle case study by 9%. As shown in table 6, the MLP neural network and PESFAM outcomes show that these methods classify the input data with more accuracy after *S3PSO* in comparison to other related methods but the advantages of the proposed *S3PSO* method are apparent when the extracted rules appear in the form of IF-THEN rules, which can be used directly for decision-making. Indeed, classification rules are also considered as white-box models, which have a high level of knowledge representation.

6. Conclusion

Classification methods attempt to predict the students' final outcome according to their activities. In this paper, we have presented a novel method based on meta-heuristic PSO algorithm in a discrete space called *S3PSO* (*Students' Performance Prediction based on Particle Swarm Optimization*) for rule induction to predict the student's final outcome. *S3PSO* tries to find all the rules with maximum *Support*, *Confidence,* and Comprehensibility from the training dataset. In *S3PSO*, a control part is added to each particle to control the presence and absence of the features in extracted rules to increase the comprehensibility of each rule. We used data from Moodle at the Tarbiat Modares University (Tehran, Iran) in computer science courses to evaluate the performance of *S3PSO.* The experimental results reveal that *S3PSO* performs better in rule detection than the other rule-based classification method like C4.5, ID3, and CART according to the *Support*, *Confidence,* and *Comprehensibility* measurements. Moreover, comparing its results with those obtained by other classification methods reveals that *S3PSO* outperforms other classification methods like SVM and Neural networks (see Table 6). For example, it improved the value of *Accuracy* criterion for the Moodle case study by 9%. We also compared the convergence speed of S3PSO with APSO, and the results obtained indicate that the convergence of *S3PSO* toward the qualified rules in the student datasets is faster than APSO. In order to improve the predictive model, other information types regarding the students' family and university activity involvement will be examined in the future works.

References

[1] Brusilovsky, P. & Peylo C. (2003). Adaptive and intelligent web-based educational systems. International Journal of Artificial Intelligence in Education (IJAIED), vol. 13, pp. 159-172.

[2] Romero, C. & Ventura S. (2013). Data mining in education. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, vol. 3, no. 1, pp. 12-27.

[3] Mostow, J. & Beck J. (2006). Some useful tactics to modify, map and mine data from intelligent tutors. Natural Language Engineering, vol. 12, no. 2, pp. 195-208.

[4] Klösgen, W. & Zytkow J. M. (2002). Handbook of data mining and knowledge discovery, Oxford University Press, Inc.

[5] Homayounfar, E., Sepehri M. M., Hasheminejad, S. M. H. & Ghobakhloo M. (2014). Designing a chronological based framework for condition monitoring in heart disease patients-a data mining approach (DM-PTTD). Iranian Journal of Medical Informatics, vol. 3, no. 3.

[6] Sebastiani, F. (2002). Machine learning in automated text categorization. ACM computing surveys (CSUR), vol. 34, no. 1, pp. 1-47.

[7] Hasheminejad, S. M. H. & Jalili S. (2013). SCI-GA: Software Component Identification using Genetic Algorithm. Journal of Object Technology, vol. 12, no. 2, pp. 3-1.

[8] Hasheminejad, S. M. H. & Jalili S. (2014). An evolutionary approach to identify logical components. Journal of Systems and Software, vol. 96, pp. 24-50.

[9] Hasheminejad, S. M. H. & Jalili S. (2015). CCIC: Clustering analysis classes to identify software components. Information and Software Technology, vol. 57, pp. 329-351.

[10] Hasheminejad, S. M. H. & Jalili S. (2009). Selecting proper security patterns using text classification. International Conference on Computational Intelligence and Software Engineering, CiSE 2009, pp. 1-5.

[11] Hasheminejad, S. M. H. & Jalili S. (2012). Design patterns selection: An automatic two-phase method. Journal of Systems and Software, vol. 85, no. 2, pp. 408-424.

[12] Romero, C., Ventura, S. & García E. (2008). Data mining in course management systems: Moodle case study and tutorial. Computers & Education vol. 51, no. 1, pp. 368-384.

[13] Romero, C., Romero, J. R. & Ventura S. (2014). A survey on pre-processing educational data. Educational Data Mining, Springer, pp. 29-64.

[14] Castro, F., Vellido, A., Nebot, À. & Mugica F. (2007). Applying data mining techniques to e-learning problems. Evolution of teaching and learning paradigms in intelligent environment, Springer, pp. 183-221.

[15] Baker, R. S. & Yacef K. (2009). The state of educational data mining in 2009: A review and future visions. JEDM-Journal of Educational Data Mining vol. 1, no. 1 pp. 3-17.

[16] Barber, R. & Sharkey M. (2012). Course correction: using analytics to predict course success. 2nd International conference on learning analytics and knowledge, ACM.

[17] Roberge, D., Rojas, A. & Baker R. (2012). Does the length of time off-task matter?. 2nd International conference on learning analytics and knowledge, ACM, pp. 234-237.

[18] Myller, N., Suhonen, J. & Sutinen E. (2002). Using data mining for improving web-based course design. In Computers in education, International Conference on IEEE, pp. 959-963.

[19] Kotsiantis, S. B. & Pintelas P. E. (2005). Predicting students marks in hellenic open university. Advanced learning technologies, The fifth IEEE international conference on Advanced Learning Technologies, IEEE, pp. 664-668.

[20] Calvo-Flores, M. D., Galindo, M. D., Jiménez, E. G., M. P. & Pineiro O. P. (2006). Predicting students' marks from Moodle logs using neural network models. Current Developments in Technology-Assisted Education, vol. 1, pp. 586-590.

[21] Pardos, Z. A., Heffernan, N. T., Anderson, B. & Heffernan C. L. (2007). The effect of model granularity on student performance prediction using Bayesian networks. User modeling Springer, pp. 435-439.

[22] Duda, R. O., Hart, P. E. & Stork D. G. (2012). Pattern classification. John Wiley & Sons.

[23] Quinlan, J. (1993). C4.5: Program for Machine Learning Morgan Kaufmann. San Mateo, CA, USA.

[24] Davis, J. & Goadrich M. (2006). The relationship between Precision-Recall and ROC curves. 23rd international conference on Machine learning, ACM, pp. 233-240.

[25] Su, X. & Khoshgoftaar T. M. (2009). A survey of collaborative filtering techniques. Advances in artificial intelligence, vol. 4.

[26] Ma, H., King, I. & Lyu M. R. (2007). Effective missing data prediction for collaborative filtering. 30th annual international ACM SIGIR conference on Research and development in information retrieval ACM, pp. 39-46.

[27] Agrawal, R., Imieliński, T. & Swami A. (1993). Mining association rules between sets of items in large databases. ACM SIGMOD Record, vol. 22, no. 2, pp. 207-216.

[28] Kennedy, J. (2011). Particle swarm optimization. In Encyclopedia of machine learning, pp. 760-766.

[29] Lara, J. A., Lizcano, D., Martínez, M. A., Pazos, J. & Riera, T. (2014). A system for knowledge discovery in e-learning environments within the European Higher Education Area–Application to student data from Open University of Madrid, UDIMA. Computers & Education vol. 72, pp. 23-36.

[30] Baker, R. S. & Inventado P. S. (2014). Educational data mining and learning analytics. Learning Analytics, Springer, pp. 61-75.

[31] Iam-On, N. & Boongoen T. (2015). Improved student dropout prediction in Thai University using ensemble of mixed-type data clustering. International Journal of Machine Learning and Cybernetics, pp. 1-14.

[32] Agudo-Peregrina, Á. F., Iglesias-Pradas, S., Conde-González, M. Á., & Hernández-García Á. (2014). Can we predict success from log data in VLEs? Classification of interactions for learning analytics and their relation with performance in VLE-supported F2F and online learning. Computers in Human Behavior, vol. 31 pp. 542-550.

[33] Chalaris, M., Gritzalis, S., Maragoudakis, M., Sgouropoulou, C. & Tsolakidis A. (2014). Improving quality of educational processes providing new knowledge using data mining techniques. Procedia-Social and Behavioral Sciences vol. 147, pp. 390-397.

[34] Xing, W., Guo, R., Petakovic, E. & Goggins S. (2015). Participation-based student final performance prediction model through interpretable Genetic Programming: Integrating learning analytics, educational data mining and theory. Computers in Human Behavior, vol. 47, pp. 168-181.

[35] Rachburee, N., Punlumjeak, W., Rugtanom, S., Jaithavil, D. & Pracha M. (2015). A Prediction of Engineering Students Performance from Core Engineering Course Using Classification. Information Science and Applications, Springer, pp. 649-656.

[36] Guruler, H. & Istanbullu A. (2014). Modeling Student Performance in Higher Education Using Data Mining. Educational Data Mining, Springer, pp. 105-124.

[37] Indira, K. & Kanmani S. (2015). Association rule mining through adaptive parameter control in particle swarm optimization. Computational Statistics vol. 30, no. 1, pp. 251-277.

[38] Kuo, R. J., Chao, C. M. & Chiu Y. T. (2011). Application of particle swarm optimization to association rule mining. Applied Soft Computing, vol. 11, no. 1, pp. 326-336.

[39] Tyagi, S. & Bharadwaj K. K. (2013). Enhancing collaborative filtering recommendations by utilizing multi-objective particle swarm optimization embedded

association rule mining. Swarm and Evolutionary Computation, vol. 13, pp. 1-12.

[40] Masoud, H., Jalili, S. & Hasheminejad S. M. H. (2013). Dynamic clustering using combinatorial particle swarm optimization. Applied intelligence, vol. 38, no. 3, pp. 289-314.

[41] Dougherty, J., Kohavi, R. & Sahami M. (1995). Supervised and unsupervised discretization of continuous features. 12th international conference of Machine learning, pp. 194-202.

[42] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P. & Witten I. H. (2009). The WEKA data mining software: an update. ACM SIGKDD explorations newsletter, vol. 11, no. 1, pp. 10-18.

[43] Witten, I. H. & Frank E. (2005). Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann.

[44] Ghosh, A. & Nath B. (2004). Multi-objective rule mining using genetic algorithms. Information Sciences, vol. 163, no. 1, pp. 123-133.

[45] Sousa, T., Silva, A. & Neves A. (2004). Particle swarm based data mining algorithms for classification tasks. Parallel Computing, vol. 30, no. 5, pp. 767-783.

[46] Esmin, A. A., Coelho, R. A. & Matwin, S. (2015). A review on particle swarm optimization algorithm and its variants to clustering high-dimensional data. Artificial Intelligence Review, vol. 44, no. 1, pp. 23-45.

[47] Lalwani, S., Singhal, S., Kumar, R., & Gupta, N. (2013). A comprehensive survey: Applications of multi-objective particle swarm optimization (MOPSO) algorithm. Transactions on Combinatorics, vol. 2, no. 1, pp. 39-101.

[48] Roy, R., Dehuri, S., & Cho, S. B. (2013). A novel particle swarm optimization algorithm for multi-objective combinatorial optimization problem. In Trends in Developing Metaheuristics, Algorithms, and Optimization Approaches, pp. 1-16.

[49] The CSV file into an ARFF website (2018), Available: http://ikuz.eu/csv2arff/.

نشریه هوش مصنوعی و داده کاوی

ارائه روشی جهت پیش‌بینی عملکرد دانش آموزان مبتنی بر الگوریتم بهینه‌سازی ذرات تجمعی

سید محمد حسین هاشمی نژاد* و مهتاب سرومیلی

گروه مهندسی کامپیوتر، دانشگاه الزهرا (س)، تهران، ایران.

ارسال ۲۰۱۷/۰۳/۱۱؛ بازنگری ۲۰۱۷/۱۱/۰۴؛ پذیرش ۲۰۱۸/۰۱/۳۰

چکیده:

امروزه به روش‌های جدیدی برای استفاده از مزایای داده‌کاوی خصوصا در سیستم‌های آموزشی نیازمندیم. داده‌کاوی در سیستم‌های آموزشی، خصوصا آموزش الکترونیکی به دلیل گسترش آنها کاربرد وسیعی دارد. در واقع فراهم سازی روشی جهت پیش‌بینی نمره پایانی دانش آموزان یکی از دلایل بکار گیری داده‌کاوی در سیستم‌های آموزشی می‌باشد. در این مقاله، ما روش جدیدی مبتنی بر دسته‌بندی قاعده گرا برای کشف روابط پنهانی که می‌تواند نمره دانش آموزان را پیش‌بینی نماید، ارائه می‌کنیم. در روش پیشنهادی ما از یک الگوریتم بهینه‌سازی ذرات تجمعی در فضای گسسته استفاده می‌کنیم. در روش پیشنهادی، هر ذره قابل تفسیر برای هرکاربر عادی مانند معلمین می‌باشد و در این روش معیارهای پشتیبانی، اطمینان و قابلیت فهم برای ارزیابی برازندگی هر قانون پیشنهادی مورد استفاده قرار گرفته است. مقایسه روش پیشنهادی با سایر روش‌های قاعده گرا مانند ID3، C4.5 و CART نشان می‌دهد که روش پیشنهادی ۳۱٪ مقدار تابع برازندگی را بر روی مجموعه داده مودل بهبود داده است. به علاوه مقایسه نتایج روش پیشنهادی با سایر روش‌های دسته‌بندی مانند SVM، KNN، APSO، شبکه عصبی و بیز ساده نشان می‌دهد که این روش ۹٪ معیار صحت را بهبود داده است و نتایج قابل قبولی به عنوان یک روش پیش‌بینی نمره نهایی دانش آموزان ارائه می‌دهد.

کلمات کلیدی: داده‌کاوی آموزشی، الگوریتم بهینه‌سازی ذرات تجمعی، دسته‌بندی قاعده گرا.