International Journal of Innovative
Computing, Information and Control
Volume **10**, Number **1**, February **2014**

ICIC International ©2014 ISSN 1349-4198
pp. 151–167

# EFFECTIVE AUDIO CLASSIFICATION ALGORITHM USING SWARM-BASED OPTIMIZATION

Changseok Bae[1], Noorhaniza Wahid[2], Yuk Ying Chung[3]
Wei-Chang Yeh[4,5], Neil William Bergmann[6] and Zhe Chen[3]

[1]Human Computing Research Section
Electronics and Telecommunications Research Institute
218 Gajeongro, Yuseonggu, Daejeon 305-700, Korea
csbae@etri.re.kr

[2]Faculty of Computer Science and Information Technology
University Tun Hussein Onn Malaysia
86400 Parit Raja, Batu Pahat, Johor, Malaysia
nhaniza@uthm.edu.my

[3]School of Information Technologies
University of Sydney
Sydney, NSW 2006, Australia

[4]Advanced Analytics Institute
University of Technology Sydney
PO Box 123 Broadway, NSW 2007, Australia

[5]Department of Industrial Engineering and Engineering Management
National Tsing Hua University
No. 101, Section 2, Kuang-Fu Road, Hsinchu 30013, Taiwan

[6]School of Information Technology and Electrical Engineering
The University of Queensland
Brisbane St. Lucia, QLD 4072, Australia

Received January 2013; revised May 2013

Abstract. *The effectiveness and usefulness of large audio databases is greatly dependent on the ability to classify and retrieve audio files based on their properties or content. Automatic classification using machine learning is much more practical than manual classification. In this paper, a new audio classification algorithm using Simplified Swarm Optimization (SSO) based on Particle Swarm Optimization (PSO) is presented. The performance of the new algorithm is compared with two existing state-of-the-art classifiers, PSO and Support Vector Machine (SVM), for an audio dataset being classified into five classes of musical instruments. The experimental results show that the proposed SSO-based classifier has improved classification accuracy (91.7%) when compared with PSO (87.2%) and SVM (88.5%). Additionally, the algorithm is shown to have simpler particle update calculations than PSO, and also requires fewer particles for classification training.*

**Keywords:** Audio classification, Swarm-based optimization, Classification

1. **Introduction.** As the amount of multimedia data which is being stored in databases increases, it is important to be able to analyze data based on its content. This is needed for indexing, searching, sorting and content-based classification. For textual data, this can be done on the text itself. However, for audio and video data, content-based management has traditionally relied mostly on metadata, i.e., textual data about the content which is added manually at the time of creation or at the time data is entered into the database.

This manual classification is extremely time consuming, highly labor intensive, expensive, and difficult. Furthermore, this information is often incomplete or totally absent. In recent years, there has been an increasing interest in the automatic classification of multimedia data. This paper investigates the classification of audio data, i.e., assigning segments of audio to different classes that can be used for searching, indexing or analysis.

In general, the content-based classification of audio data involves two stages: the first stage is to extract the key low-level audio features, and the second stage is to classify the audio data based on these extracted key features. The novel contribution of this paper is in the second stage, i.e., audio classification based on extracted features. Feature extraction uses a commonly applied method called Mel Frequency Cepstral Coefficients (MFCC). Then, a new swarm-based classification algorithm using these features will be presented which is used to classify the audio. The performance of the new algorithm, Simplified Swarm Optimization (SSO), is compared with two existing high performance classifiers: Particle Swarm Optimization (PSO) and the Support Vector Machine (SVM).

The remainder of this paper is organized as follows. Related research is surveyed in Section 2, and a detailed explanation of the proposed algorithm is presented in Section 3. The performance results of SSO, compared with SVM and PSO, are presented and analyzed in Section 4. Some concluding remarks and future research directions are given in Section 5.

## 2. **Related Works.**

There has recently been considerable research on audio classification using many different types of machine learning and data mining techniques. Techniques are generally divided into two types: rule-based systems and machine learning models [1-3]. In the rule-based approach, the audio category is determined based on a predetermined set of rules about the audio features. For example, the average energy of an audio segment can be used to classify telephone audio into speech and silence. The machine learning approach has recently become more popular for audio classification since it does not require prior knowledge of the audio features. Instead, the machine learning model is based on labeled information called training data, and this training data is used to generate the classifier. A different set of test data can then be used to verify the classification performance. Designing a machine learning classifier is basically an optimization problem of how to choose the best set of classifier parameters to optimize classification accuracy.

Optimization is a kind of stochastic problem which can be used for guaranteed cost control [4-7]. This paper proposes a new swarm-based optimization algorithm for audio classification based on machine learning. Swarm-based optimization techniques such as Particle Swarm Optimization (PSO) [8], Artificial Bee Colony (ABC) [9], Ant Colony Optimization (ACO) [10], and Immune System (IS) [11] are relatively new approaches, and they have not been widely explored in the area of mining audio data patterns. Among them, PSO (also known as the swarm intelligence algorithm) is one of the popular swarm-based algorithms [12,13]. PSO has a shown superior performance compared with Genetic Algorithms (GA), particularly in optimization applications [14]. In recent years, PSO has been successfully applied in many different applications due to its robustness and simplicity [15,16]. When compared with other stochastic optimization techniques such as GA, PSO has fewer complicated operations and parameters to adjust, and can be coded in just a few lines. In addition, PSO has been shown to be a promising technique for discovering useful and interesting knowledge from databases [17]. Because of these advantages, PSO has received an increasing amount of attention in the data mining community in recent years [18-20]. However, PSO suffers from the problem of premature convergence, especially in high-dimensional multimodal problems, and the convergence speed decreases

as the number of iterations is increased, leading to difficulties in quickly finding the best fitness value [21].

3. **The Proposed Simplified Swarm Optimization (SSO) for Audio Classification.** In this paper, a new audio classification approach based on a new swarm optimization algorithm is proposed and presented. A common issue in classifying music audio sequences is identification of the different musical instruments in the audio clip. This may be used to identify particular music clips which contain certain instruments (e.g., find clips with guitar but no piano), or it may be used to index the position within a clip (e.g., find the piano solo in a particular clip). Therefore, testing for this SSO classifier is based on classifying five categories of audio content, which are based on the piano, trumpet, violin, guitar and accordion.

Twelve features are extracted to categorize the audio content, as described in Section 3.1. All of the extracted feature vectors were then fed into the proposed Simplified Swarm Optimization (SSO) algorithm to mine the audio content pattern. Two existing classifiers, namely, the Support Vector Machine (SVM), which is based on a statistical learning theory algorithm, and a PSO rule-based classifier based on a swarm optimization algorithm, are used to compare the classification performance of the new SSO algorithm. The overall structure of the proposed audio classification system is shown in Figure 1.

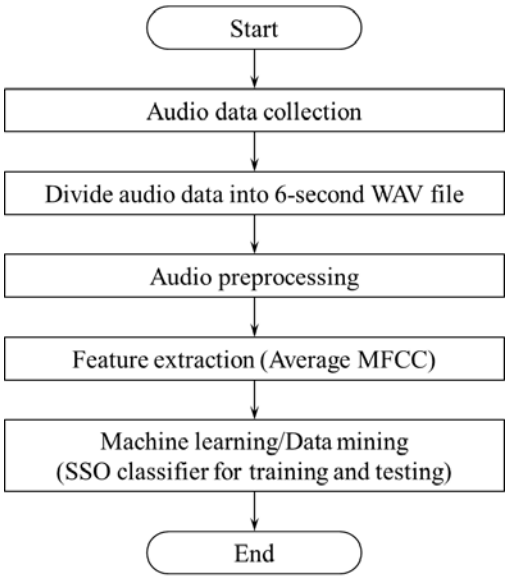



FIGURE 1. A block diagram of the new digital audio classification system

3.1. **Audio feature extraction.** The first step is to extract the features in an audio signal. An audio signal can be considered from two different domains, the time domain and the frequency domain. In the time domain, a silence ratio (SR), which is the time ratio of no sound during a given time interval, is one of the most useful features. In the frequency domain, features are spectral features such as pitch, harmony, bandwidth, and intensity.

3.1.1. *Original MFCCs feature.* A Mel-Frequency Cepstral Coefficients (MFCC) is a non-parametric representation of an audio signal which reflects the human auditory system quite well. The term mel-frequency indicates a frequency band spaced on the mel-scale, a perceptual scale of pitches [22]. There are several speech processing algorithms that

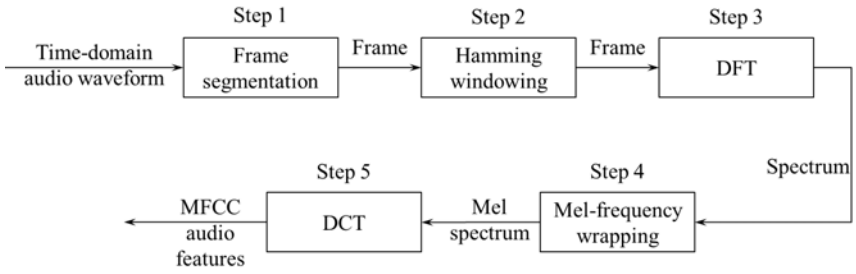



FIGURE 2. MFCC transformation procedure

have employed MFCCs for classifying the features in an audio signal, owing to its good discriminating ability [23,24]. In MFCC, the mel-frequency scale has a linear frequency spacing below 1 kHz and a logarithmic spacing above 1 kHz. Figure 2 summarizes the procedure used for MFCC feature extraction.

3.1.2. *Average MFCCs feature.* Using test data from many audio clips to generate separate MFCC features for each segment results in too much training data for efficient learning. For example, suppose there are five classes of audio with 40 audio files of 6 seconds for each class. With a 128 kbps audio bit rate, 16 bits per sample, 512 samples per frame, and 50% overlap between frames, 186 frames for each audio file in each class will be extracted using the original MFCCs, which yield 7,440 instances in each class. Altogether, there are 37,200 training instances to be fed into the classifier. It is inefficient to feed all 37,200 data sets into a classifier. Therefore, another normalized MFCC array will be applied by partitioning one audio clip into three parts in the time domain. MFCC features will be averaged within each partition.

Suppose that there are $n$ Hamming windows for one audio clip, then each partition will have $n/3$ windows. If there are $k$ MFCC coefficients in one window, the mean of the corresponding coefficients of each partition will be calculated, and each partition will produce $k$ MFCC coefficients, summarizing the data in the $n/3$ windows. The vectors from these three partitions will then be connected to form a new vector that will be used for the classification process in the next step. Figure 3 illustrates the process used to calculate the average MFCC feature vector. This technique has been used successfully in our previous research [25].

Because each partition generates a feature vector with the same length, the Euclidean distance can be used to calculate the distance between the average feature vectors of two different audio files. The Euclidean distance is an effective way to measure the similarity between two arrays with the same dimension and phase [26]. For any dimension of $i$, it

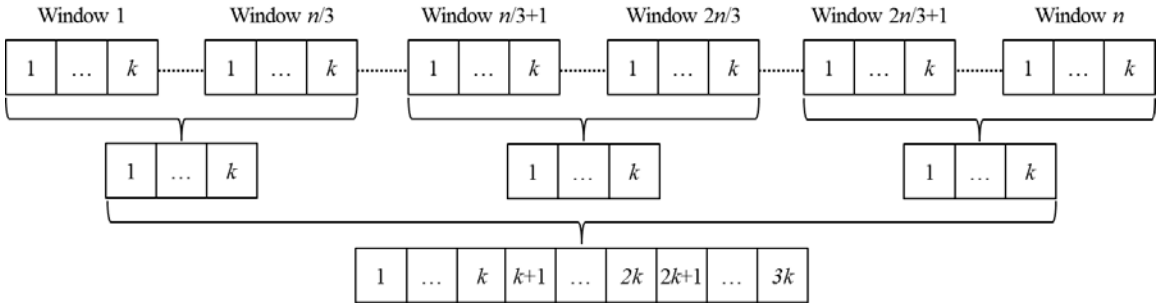



FIGURE 3. The process used to calculate the average MFCC feature vector [25]

is defined as

$$D = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}. \tag{1}$$

3.2. **Particle swarm optimization.** Particle Swarm Optimization imitates the intelligent behavior of members of a group sharing their experience in a society. PSO is based on adaptive social behaviors with individual reactions to the environment or search space. It was originally proposed by Kennedy and Eberhart as a simulation of the social behavior of organisms, such as flocking birds and schooling fish [8]. PSO can also be applied to classification problems. PSO comprises a set of individual particles searching for the optimal point in their neighborhood. The movement of particles depends on the behavior of other particles in the search space and the current best solutions that have been found in the search space. During PSO training, the particles move around in a multidimensional search space with their velocities being constantly updated by the best found position (*pbest*) of the particles and the best solution met by their neighbors (*gbest*).

Each particle represents an individual position within a $D$-dimensional search space. The $i$-th particle is represented as $X_i = (x_{i1}, x_{i2}, \cdots, x_{iD})$, the best previous position *pbest* of any particle is $P_i = (p_{i1}, p_{i2}, \cdots, p_{iD})$, and the velocity for particle $i$ is $V_i = (v_{i1}, v_{i2}, \cdots, v_{iD})$. Each particle is characterized by its current position $x_{id}$, its previous position $p_{id}$, and its velocity $v_{id}$. The globally best particle that represents the fittest particle found thus far in the entire swarm is denoted by $P_g$. During each iteration, Equation (2) is used to update the velocity of each particle.

$$v_{id}^t = w \cdot v_{id}^{t-1} + c_1 \cdot rand_1 \cdot (p_{id} - x_{id}^{t-1}) + c_2 \cdot rand_2 \cdot (p_{gd} - x_{id}^{t-1}). \tag{2}$$

In this equation, $c_1$ and $c_2$ denote the acceleration coefficients, $d = 1, 2, \cdots, D$, and $rand_1$ and $rand_2$ are random numbers uniformly distributed within $[0, 1]$. Acceleration coefficients $c_1$ and $c_2$ control the exploration of the particle movement in a single iteration. Typically, both coefficients are set to a value of 2.0, and the inertia weight $w$ in Equation (2) is also used to control the convergence behavior of the PSO [27]. Each particle then moves to a new potential position as follows:

$$x_{id}^t = x_{id}^{t-1} + v_{id}^t, \quad d = 1, 2, \cdots, D. \tag{3}$$

The algorithm of a standard PSO is as follows:

Step 1: Initialize a population of particles with random positions and velocities.
Step 2: Evaluate the fitness value of each particle in the population.
Step 3: Obtain *pbest* if the fitness value of particle $i$ is better than its *pbest* fitness value, and then set a new fitness value as a new *pbest* of particle $i$.
Step 4: Obtain *gbest* if any *pbest* is updated and is better than the current *gbest*, and then set *gbest* to the current value of particle $i$.
Step 5: Update the particles velocity and position according to Equations (2) and (3).
Step 6: Stop the iteration if the best fitness value or maximum generation is met; otherwise return to Step 2.

3.3. **Simplified swarm optimization algorithm for audio classification.** In this section, a new data mining approach based on the idea of a traditional PSO [8] and our previously proposed Discrete PSO (DPSO) [28,29] is introduced. The proposed approach, Simplified Swarm Optimization (SSO), is developed with each particle coded as a positive integer number. We can apply SSO to data mining applications by accelerating the occurrence of high-quality potential solutions in the population. The basic idea of this

approach is to discover the patterns from the corresponding rule that are commonly found in the best solution of the population. This approach is significantly different from other methods, which have combined data mining and PSO together. To the best of our knowledge, the use of a swarm-based algorithm such as PSO algorithm in the context of audio data mining and classification has not been previously reported. An introduction to the proposed SSO algorithm is presented in the following subsection.

3.3.1. *SSO algorithm.* The proposed SSO algorithm is adopted from our previous work [29,30]. Initially, the swarm population size, the number of maximum generations, and three predetermined parameters will be set. In every generation, the particles positional value in each dimension will be maintained or updated by its *pbest* value, or will be updated by the *gbest* value or replaced by generating a new random number according to the procedure described in Equation (4).

$$x_{id}^{t} = \begin{cases} x_{id}^{t-1}, & \text{if } rand() \in [0, C_w), \\ p_{id}^{t-1}, & \text{if } rand() \in [C_w, C_p), \\ g_{id}^{t-1}, & \text{if } rand() \in [C_p, C_g), \\ x_{id}, & \text{if } rand() \in [C_g, 1]. \end{cases} \tag{4}$$

In this equation, $i = 1, 2, \cdots, m$, where $m$ is the swarm population; $X_i = (x_{i1}, x_{i2}, \cdots, x_{iD})$, where $x_{iD}$ is the positional value of the $i$-th particle with respect to the $D$-th dimension $(d = 1, 2, \cdots, D)$ of the feature space; and $C_w, C_p$ and $C_g$ are three predetermined positive constants with $C_w < C_p < C_g$. The best solution achieved by each particle (*pbest*) is denoted by $P_i = (p_{i1}, p_{i2}, \cdots, p_{iD})$, and the best solution achieved thus far by the whole swarm (*gbest*) is represented by $G_i = (g_{i1}, g_{i2}, \cdots, g_{iD})$. In addition, $x_{id}$ represents the new values for the particle in every dimension, which are generated from the random function $rand()$, using a uniform distribution in $[0, 1]$. The updated strategy for the particles positional values in SSO is presented in the following steps and illustrated in Figure 4.

Step 1: Initialize the swarm size $(m)$, the maximum generation $(\max Gen)$, the maximum fitness value $(\max Fit)$, $C_w$, $C_p$ and $C_g$.

Step 2: During every iteration, a random number, $R$, from $[0, 1]$ will be generated for each dimension.

Step 3: Perform a comparison strategy where

$$\begin{cases} \text{If } (0 \le R < C_w), & \text{then } x_{id}^{t} = x_{id}^{t-1}, \\ \text{Else if } (C_w \le R < C_p), & \text{then } x_{id}^{t} = p_{id}^{t-1}, \\ \text{Else if } (C_p \le R < C_g), & \text{then } x_{id}^{t} = g_{id}^{t-1}, \\ \text{Else if } (C_g \le R \le 1), & \text{then } x_{id}^{t} = new(x_{id}). \end{cases} \tag{5}$$

Step 4: This process will be repeated until the termination condition is satisfied (e.g., the global maximum has been found or a sufficient number of iteration steps have been reached); otherwise, return to Step 2.

3.3.2. *The rule mining scheme.* In the context of a classification task in audio data mining, the optimization task is to select a best set of classification rules based on the values of individual features. One particle of the swarm population consists of a conjunction of conditions composing IF-THEN prediction rules. Such rules have the advantage of being a high-level symbolic knowledge representation which can be optimized to find a small number of rules with higher fitness [31]. In the proposed algorithm, we introduce a new threshold for each attribute, which comes from the lowest and highest data range values

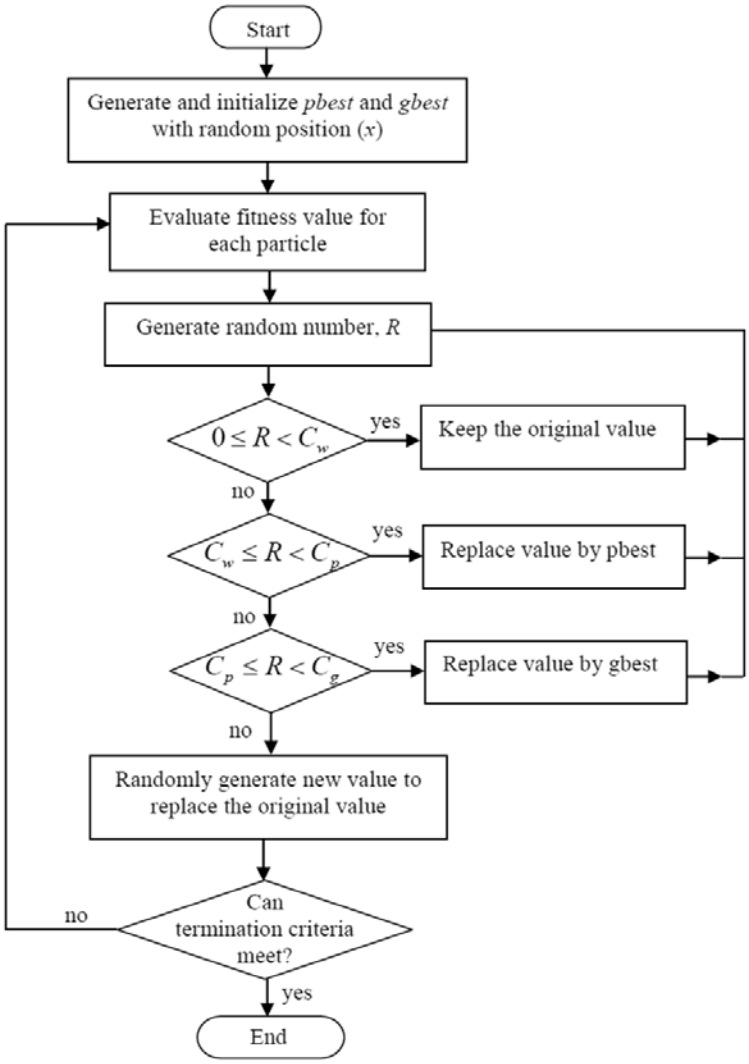



FIGURE 4. The particles update position scheme for the proposed SSO

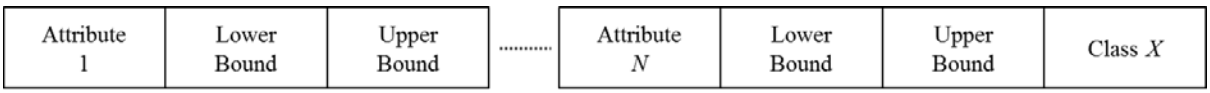



FIGURE 5. Rule mining encoding scheme

of the given feature in the training dataset. The former is called a $LowerBound$ and the latter is called an $UpperBound$. The $LowerBound$ and $UpperBound$ act as a threshold for the prediction condition. Figure 5 shows the encoding scheme of the particles position (term), which consists of three parts: the attribute, $LowerBound$ and $UpperBound$. Each particle contains a number of attributes (dimensions) ranging from 1 to $N$. The last cell, Class $X$, represents the predicted class of the rule.

The general form of the IF-THEN rules generated by PSO and SSO can be performed in all dimensions as follows:

IF ($LowerBound \le x_{id} \le UpperBoun$) is true, THEN the prediction is Class $X$.

This approach will most likely produce a seeding position outside the range of values seen within the training dataset. The most likely place that a particle will be seeded is around the lowest and highest values in the training dataset. However, the seeding

examples are from the class being predicted by the rule that the particle is encoding. Hence, if the manner in which the values from these examples are distributed is different from all the examples, then the search will hopefully be biased in an useful way. The *LowerBound* and *UpperBound* values are obtained using Equations (6) and (7) as below.

$$LowerBound = x_{id} - \rho \cdot range(X_{\max} - X_{\min}), \tag{6}$$

$$UpperBound = x_{id} + \rho \cdot range(X_{\max} - X_{\min}), \tag{7}$$

where $X_i = (x_{i1}, x_{i2}, \cdots, x_{iD})$ denotes the $i$-th seed value of the $N$-th corresponding attribute in each $D$-dimension; $\rho$ is a random number in the range of 0 and 1; and $range(X_{\max} - X_{\min})$ is the range of values of the training data source for each feature.

In the case of SSO rule mining, the values of *LowerBound* and *UpperBound* will not be updated during or after the generation process. However, in the case of PSO rule mining, the values of *LowerBound* and *UpperBound* will be updated according to the new position updated by the velocity after the generation process. Here, we employed Equation (3) to update the new current value of the corresponding position. The update strategy of the *LowerBound* and *UpperBound* values is performed according to the comparison strategy as follows:

$$\begin{cases} \text{If } (0 \le R < C_w), & \text{then } \begin{cases} LowerBound(x_{id}^t) = x_{id}^{t-1}, \\ UpperBound(x_{id}^t) = x_{id}^{t-1}, \end{cases} \\ \text{Else if } (C_w \le R < C_p), & \text{then } \begin{cases} LowerBound(x_{id}^t) = LowerBound(p_{id}^t), \\ UpperBound(x_{id}^t) = UpperBound(p_{id}^t), \end{cases} \\ \text{Else if } (C_p \le R < C_g), & \text{then } \begin{cases} LowerBound(x_{id}^t) = LowerBound(g_{id}^t), \\ UpperBound(x_{id}^t) = UpperBound(g_{id}^t), \end{cases} \\ \text{Else if } (C_g \le R \le 1), & \text{then } \begin{cases} LowerBound(x_{id}^t) = x_{id}^{t-1} - rand() \times range(x_{\max} - x_{\min}), \\ UpperBound(x_{id}^t) = x_{id}^{t-1} + rand() \times range(x_{\max} - x_{\min}). \end{cases} \end{cases} \tag{8}$$

This technique is capable of generating a high-quality potential solution in the population, which is supported by an improvement in the classification performance observed in these initial experiments.

3.3.3. *Rule evaluation.* In the previous literature, several different evaluation metrics for rule evaluation have been presented, and the most commonly used metric is the classification accuracy, which has a more direct relation to the generalization accuracy. It is necessary to estimate the quality of every candidate rule. To evaluate the goodness of the rules, the quality of the rule (fitness function) is computed in the solution space, and returns a single number that can represent the fitness value of the related position. In data mining, the data will typically be divided into two parts: training data and test data. Training data is used to generate a model according to the given rules in the target problem, and that model will be used later for the test data to obtain the validation accuracy. How well the rule will perform in the testing phase will depend on the reliability of the classification accuracy measurement. In general, the standard classification accuracy rate can be written as

$$\text{Standard classification accuracy rate} = \frac{TP + TN}{TP + FP + FN + TN}. \tag{9}$$

However, the class distribution in many nonlinear classification problems is highly unbalanced. Therefore, Equation (9) cannot be used to measure the accuracy rate for this

model effectively [32]. For this reason, a more comprehensive metric for a rule evaluation has been used, where quality is expressed as follows:

$$\text{Quality}(Q) = \text{sensitivity} \times \text{specificity} = \frac{TP}{TP + FN} \times \frac{TN}{TN + FP}, \tag{10}$$

where $TP$, $FP$, $TN$ and $FN$ are the number of true positives, false positives, true negatives, and false negatives associated with the rule, respectively [33].

(1) True Positive ($TP$): the number of cases covered by a rule that have the class predicted by that rule.
(2) False Positive ($FP$): the number of cases covered by a rule that have a class different from the class predicted by that rule.
(3) True Negative ($TN$): the number of cases that are not covered by a rule and do not have the class predicted by that rule.
(4) False Negative ($FN$): the number of cases that are not covered by a rule but have the class predicted by that rule.

In the classification problem, $Q$ ranges from 0 to 1, and the optimization goal is achieve the maximum $Q$ value.

3.3.4. *Rule pruning.* In data mining, the main goal of rule pruning is to eliminate the irrelevant attributes that might have been unnecessarily included in the rule, i.e., to reduce the number of terms in the rule. Rule pruning can potentially increase the predictive power of the rule, and can also avoid over-fitting to the training set [33]. Also a shorter rule is more easily understood by users [34]. During the rule discovery process, once the highest-quality rule for each class has been found using the training set, the best rule will be added to the rule set after being pruned using a pruning procedure. The idea of the pruning process is to iteratively remove each attribute (term) one at a time from the rule, and at the same time keep improving the quality of the rule. During the initial iteration, the process will start with the full rule. The rule quality is then computed according to Equation (10). After that, the attribute pairs are checked in the reverse order to which they were selected to see if a pair can be removed without decreasing the rule quality. This involves tentatively removing terms from each rule and seeing if the removal of each term affects the accuracy of the entire rule set. If the removal of individual terms does not affect the accuracy it is permanently removed. If it does affect the accuracy, then it is replaced, and the algorithm moves onto the next term, and eventually the next rule. To ensure that the process used for minimizing the length of the rule continues, the quality of the obtained rule must be greater or equal to the original rule. After the pruning procedure, the examples covered by the rule are removed from the training set. An example is said to be covered by the rule if the example satisfies all terms (attribute value pairs) in the rule antecedent (IF part). A WHILE loop is performed as long as the number of uncovered examples of each class in the training set is greater than zero. Once this threshold has been reached, the training set is reset by adding the previously covered examples. After completing the pruning process, all rules that do not contribute to the classification accuracy will be removed. This is achieved by classifying the training set using the rule list. If any rules do not classify any examples correctly, they are removed. Later, a series of test data are used to measure the classification accuracy according to the rule set obtained. For each instance, a prediction value is computed by examining every element in the rule set for the corresponding class if it is covered by the rule. The prediction value is calculated according to the prediction function, as shown in Equation (11).

$$\text{Prediction value} += \alpha \times \text{rule quality} + \beta \times \text{percentage of rule covered}, \tag{11}$$

where $\alpha$ and $\beta$ are two parameters corresponding to the importance of the rule quality and the percentage of the rule covered, respectively. The former is known as the quality weight and the latter is known as the coverage weight, $\alpha \in [0, 1]$ and $\beta = (1 - \alpha)$. The prediction value for each class is accumulated, and the final result is calculated from the class with the highest prediction value.

4. **Experimental Results.** To verify the performance of the new SSO-based algorithm on audio data, a well-known benchmark classifier SVM, and one popular swarm-based algorithm (PSO) classifier were compared with the new SSO-based algorithm. The SVM was implemented in WEKA [34], an open-source machine-learning package. A PSO rule-based classifier was designed and implemented using the same rule-mining scheme, rule evaluation, and rule pruning as described in the previous section.

4.1. **Data collection and preprocessing.** For the experiments, significant effort was needed to collect audio data from diverse sources, mainly from the Internet, as very few databases are publicly available. The audio data considered in this research were mainly in MP3 and WAV format with an assortment of sample rates of 22.1 and 44.1 kHz. The authors in [35] showed that a lower data rate mono channel has been widely used owing to its ability to provide better models, and it also can reduce the computational time compared to using a stereo file or higher sampling rates. So, before implementing the feature extraction phase, the audio signals were down-sampled to 8 kHz and converted into a consistent WAV format, with 16 bits per sample in a mono channel. This was done using sound editing software. Table 1 shows the details of the properties for the test audio data used in our experiments.

TABLE 1. A summary of audio signal properties

| Audio property | Value |
|---|---|
| Audio bit rate | 128 kbps |
| Audio sample size | 16 bit |
| Channel | 1 (mono) |
| Audio format | PCM |
| Audio length | 6 seconds |

The audio stream was divided into non-overlapping sub-clips of 6 s in length. Later, these sub-clips were manually classified according to their category and used as the training and test data. Altogether, 2,500 audio sub-clips were used in the experiments. These recordings were from five well known musical instruments with 500 recordings from each musical instrument. The five categories of musical instruments – piano, trumpet, violin, guitar, and accordion – were manually labeled in the dataset.

4.2. **Feature extraction.** In first step, the average MFCC features were extracted from the 2,500 musical recordings. Each audio sound was divided into frames where 512 samples per Hamming window were employed during the MFCC transformation process. The Hamming window shifted for every half-window size, i.e., 256 samples (32 ms) with a 73% overlap between adjacent frames. Twelve audio features were extracted from each window, and the average feature values were calculated for each frame. Twelve MFCC coefficients were used as the audio features because these have been previously shown to provide good classification [36].

4.3. **Evaluation of classification accuracy.** To evaluate the robustness of the proposed SSO-based algorithm, a cross validation for the audio classification was performed. Here, a 10-fold cross validation approach was used to estimate the predictive accuracy of the algorithms. In this approach, data instances were randomly assigned to one of 10 equally sized subsets. At each iteration, nine of these sets are merged to form the training set, while the classification accuracy of the algorithm is measured by using the remaining subset as test data. This process is repeated ten times, choosing a different subset as the test set each time until all data instances have been used nine times for training and once for testing. The final predictive accuracy is computed and reported based on the average of the classification accuracies of the ten runs of the discovered rule set. This approach has been used in all experiments to control their validity.

4.4. **Results and discussion.** The performance of the proposed SSO was measured by comparing it with a traditional PSO-rule based classifier. The aim was to discover the optimum population size and number of generations for the SSO and PSO that leads to the highest classification accuracy. Hence, in each experiment, the simulation was run using five different numbers of particles (10, 20, 30, 40 and 50) with a different number of generations ranging from 10 to 50. Since the SSO and PSO are stochastic search algorithms, it was hard to determine their performance through one or two trials. Therefore, experiments were initially run ten times. However, it was observed that after five runs, the algorithms have the same results. Therefore, each experiment was run five times, and all the presented results are averaged over five trials of the simulation.

Table 2 lists the implementation specifications and parameters used in the experiments that gave the best results. Note that the setting of the parameters in the SSO and PSO to generate the best solution was data dependent, and this is worthy of further investigation in a future study. The results of the classification accuracies and the average rule set size for both the SSO and PSO algorithms are shown in Table 3. From Table 3, although the average rule set size generated by PSO is simpler (smaller) than SSO, SSO has managed to achieve a higher classification accuracy compared with PSO over five runs.

To validate the competitiveness of SSO with other industrial benchmark classifiers, SVM was also used on our audio dataset. SVM is one of the most popular statistical techniques used for audio classification problems. SVM is a learning machine that plots the training vectors in a high-dimensional feature space, and labels each vector by its class [37]. SVM classifies data by determining a set of support vectors, which are members of the set of training inputs that outline a hyper-plane in the feature space.

We employed SVM with its default parameters as a set in WEKA. In this paper, the Sequential Minimal Optimization (SMO) [38] algorithm was implemented for training

TABLE 2. Parameter settings for the SSO and PSO algorithms

| Parameter settings | SSO | PSO |
|---|---|---|
| Number of particles ($m$) | 30 | 30 |
| Maximum generation ($\max Gen$) | 50 | 50 |
| Maximum fitness | 1.0 | 1.0 |
| $C_w$, $C_p$, $C_g$ | 0.1, 0.4, 0.9 | – |
| Coverage weight, $\alpha$ | 0.5 | – |
| Quality weight, $\beta$ | 0.5 | – |
| $c_1$, $c_2$ | – | 2.0, 2.0 |
| Maximum weight | – | 0.9 |
| Minimum weight | – | 0.4 |

TABLE 3. Comparison of the classification accuracies and average rule set size for SSO and PSO over five runs when $m = 30$ and $\max Gen = 50$

| | SSO | | PSO | |
|---|---|---|---|---|
| Number of runs | CA | ARS | CA | ARS |
| 1 | 90.8 | 15.5 | 85.8 | 14.2 |
| 2 | 91.9 | 15.8 | 86.6 | 15.4 |
| 3 | 91.9 | 15.2 | 86.6 | 15.2 |
| 4 | 91.9 | 15.0 | 86.6 | 15.4 |
| 5 | 91.9 | 15.0 | 86.6 | 15.2 |
| Average | 91.7 | 15.3 | 86.4 | 15.1 |
| Max | 91.9 | 15.8 | 86.6 | 15.4 |
| Min | 90.8 | 15.0 | 85.8 | 14.2 |
| Standard Deviation | 0.49 | 0.35 | 0.36 | 0.50 |

(Note: CA = Classification Accuracy, ARS = Average Rule set Size)

TABLE 4. The classification accuracies (%) of SSO and PSO in different settings of $m$ and $\max Gen$, and SVM

| | | SSO | | | | | PSO | | | | | SVM |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $m$ | | 10 | 20 | 30 | 40 | 50 | 10 | 20 | 30 | 40 | 50 | |
| $\max Gen$ | 10 | 80.8 | 85.6 | 86.3 | 87.6 | 87.6 | 76.5 | 78.4 | 82.3 | 83.3 | 83.8 | 88.5 |
| | 20 | 83.2 | 88.4 | 88.8 | 89.8 | 89.8 | 80.7 | 83.8 | 84.0 | 84.1 | 84.2 | |
| | 30 | 85.6 | 88.9 | 90.1 | 90.8 | 90.8 | 83.8 | 84.1 | 84.2 | 84.5 | 84.9 | |
| | 40 | 88.8 | 89.8 | 90.0 | 91.2 | 91.2 | 82.8 | 83.8 | 85.2 | 85.7 | 86.6 | |
| | 50 | 89.1 | 90.8 | 91.7 | 91.7 | 91.7 | 83.8 | 84.9 | 86.4 | 86.8 | 87.2 | |

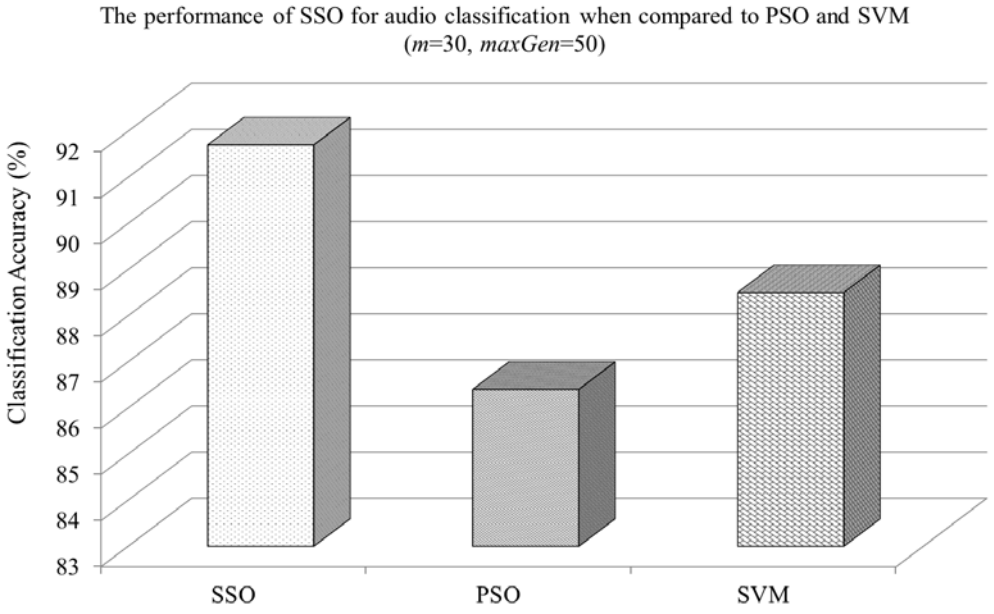



FIGURE 6. The overall performance of SSO against PSO and SVM

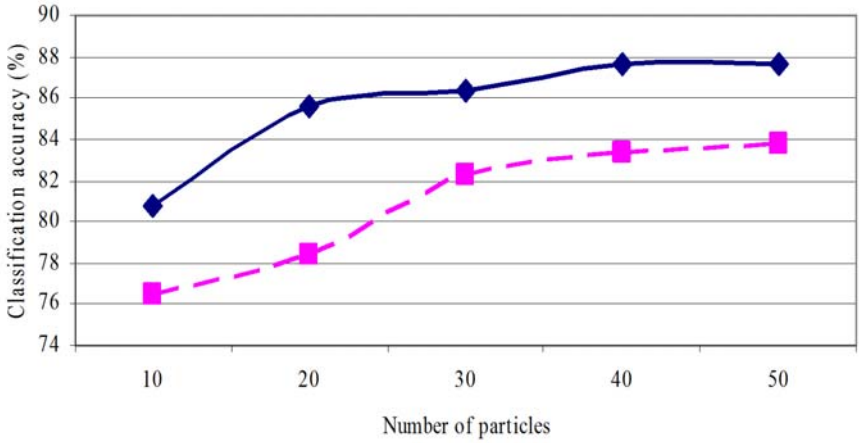
the SVM classifier. Here, the Radial Basis Function (RBF) kernel was used during the training of the known and labeled feature vectors derived from the dataset. The equation for the RBF kernel is $K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2)$.

From Table 4, it was observed that SSO can achieve best higher accuracy when $m = 30$ and $\max Gen = 50$. On the other hand, PSO can achieve its best results with $m = 50$ and $\max Gen = 50$. Thus, to obtain the highest classification accuracy, the SSO algorithm tends to use a smaller total evaluation number of 1,500, when compared with PSO with a 2,500 total evaluation number. Furthermore, in all different settings of $m$ and $\max Gen$, SSO performs better than PSO, and it also outperforms the SVM as shown in Table 4. In terms of the overall classification performance, SSO has clearly shown its superior performance by correctly classifying the five classes from the audio data with 91.7% accuracy when compared with PSO (87.2%) and SVM (88.5%).

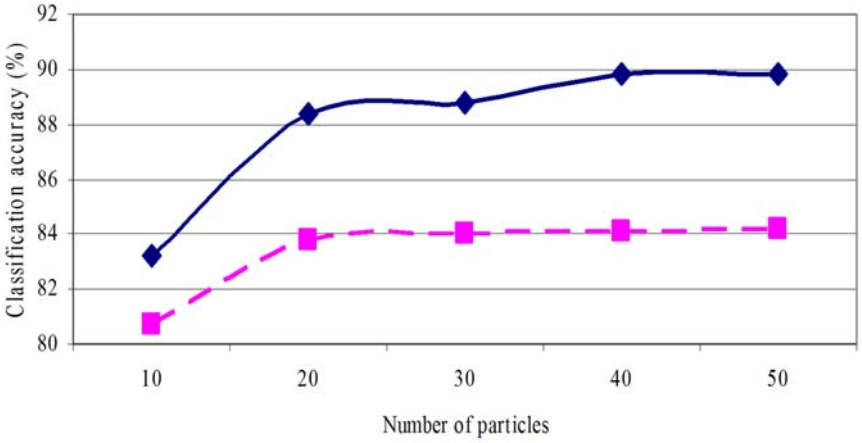
Figure 6 graphically shows the classification performance of SSO, PSO and SVM. The parameters for SSO and PSO are set according to those listed in Table 2. Interestingly, we can see a substantial improvement in accuracy between SSO and PSO of 5.3%, and 3.2% difference between SSO and SVM. As this is the first attempt to apply swarm-based optimization to classify audio data, an improved performance of 3.2% to 5.3% for SSO can be considered as a significant contribution in the data mining and data classification research areas. To the best of our knowledge, this is the first reported use of swarm-based optimization algorithms for audio data pattern classification.

Figure 7 illustrates the performance of SSO under five different iteration conditions with respect to various particle numbers when compared to PSO. From the graphs in Figure 7, we can see that the accuracies for both SSO and PSO gradually increase when the number of particles increases. Figures 7(a) through 7(d) show that the SSO performance tends to stabilize when the number of particles is equal to 40 and 50 with a classification accuracy of 87.6%, 89.8%, 90.8% and 91.2% for a maximum generation of 10, 20, 30 and 40, respectively. When the number of generations is equal to 50, SSO can persistently achieve the highest classification accuracy of 91.7% with only 30 particles, as shown in Figure 7(e). As mentioned earlier, the parameter settings for SSO and PSO are case dependent; therefore, in this experiment we only need 30 particles for SSO to achieve high audio classification accuracy. It should be noted that SSO does not require the calculation and use of the velocity and only needs to update the particles position according to some simple conditions, leading to an improvement in the computation performance. On the other hand, under all maximum generation conditions, PSO requires more particles ($m = 50$) to obtain high classification accuracy.

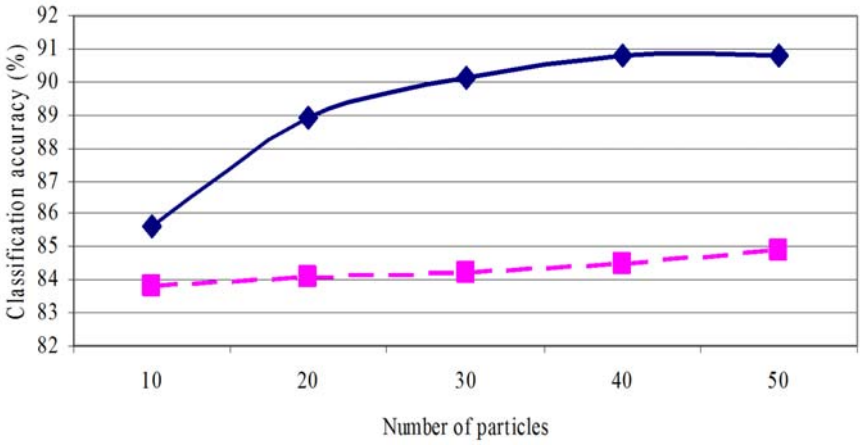
SSO consistently outperforms PSO. PSO considers each particle based on two key vectors, the velocity and position, and thus, PSO needs to allocate more memory than



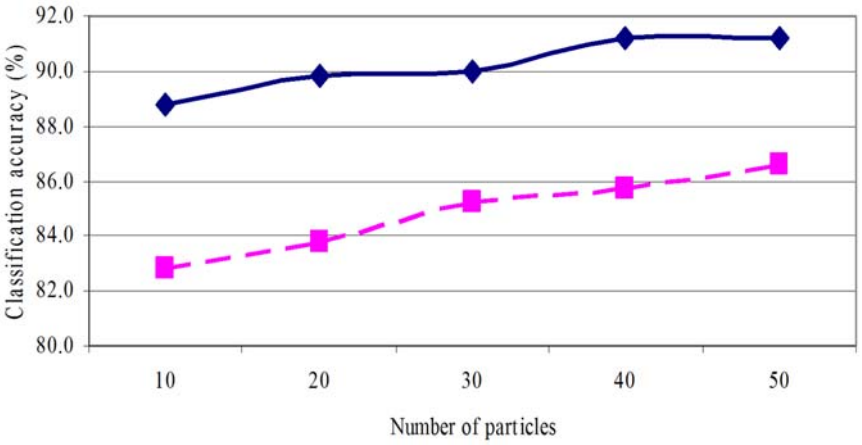

(a) Performance result for various particles number vs maximum generation = 10



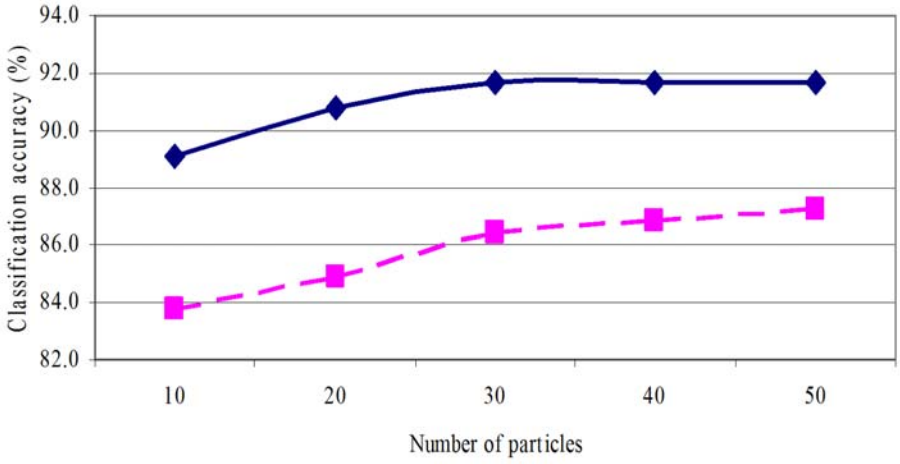

(b) Performance result for various particles number vs maximum generation = 20




(c) Performance result for various particles number vs maximum generation = 30




(d) Performance result for various particles number vs maximum generation = 40




(e) Performance result for various particles number vs maximum generation = 50

FIGURE 7. Performance results for swarm classifiers with respect to different numbers of particles against five maximum generations: (a) $\max Gen = 10$, (b) $\max Gen = 20$, (c) $\max Gen = 30$, (d) $\max Gen = 40$, and (e) $\max Gen = 50$. (——— SSO, – – – PSO)

SSO for each particle in each generation. In addition, each particle in PSO requires the use of more than two equations, and generates three random numbers, five multiplications, and three additions to update the position of any particle [29].

5. **Conclusions.** An efficient audio classification algorithm has been presented in this paper. Simplified Swarm Optimization (SSO) is a new data mining algorithm derived from the traditional Particle Swarm Optimization (PSO), and it mimics the biological behavior of swarm intelligence. New $LowerBound$ and $UpperBound$ thresholds have been introduced for each particle encoding scheme to improve the audio classification accuracy. To evaluate the effectiveness of the proposed method, experiments were carried out on a database with audio sounds of five different musical instruments: piano, trumpet, violin, guitar, and accordion. The comparison results demonstrate the difference in performance between two algorithms based on a biologically inspired paradigm, and one algorithm based on statistical learning theory. Our results show that the SSO classifier can increase the classification accuracy by 3.2% and 5.3% over SVM and PSO, respectively. SSO has demonstrated superior performance to both of these algorithms for audio classification based on decision-rule generation.

To validate the competitiveness of SSO with PSO, five different numbers of particles and maximum generations have been performed in each experiment. The results show that SSO requires fewer particles to achieve higher accuracy. To the best of our knowledge, this is the first time the SSO and PSO algorithms have been applied to the audio classification problem.

The results show that SSO is a promising approach for automated multimedia data classification. The results show in particular that SSO can classify music according to the instrument which is playing, which is very useful for music searching and indexing. The SSO algorithm is still in its early stages, and further improvements in results can be expected with more investigation. Further research will focus on improving the search strategy in SSO and investigating other audio feature sets.

**Acknowledgment.** This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (No. 2010-0028631).

## REFERENCES

[1] W. Pan, Y. Yao, Z. Liu et al., Audio classification in a weighted SVM, *Proc. of 2007 International Symposium on Communications and Information Technologies*, pp.468-472, 2007.

[2] C. Lim and J. Chang, Adaptive kernel function of SVM for improving speech/music classification of 3GPP2, *ETRI Journal*, vol.33, no.6, pp.871-879, 2011.

[3] S. Yoon, C. Lyuh, I. Chun et al., A new support vector compression method based on singular value decomposition, *ETRI Journal*, vol.33, no.4, pp.652-655, 2011.

[4] S. Xu, J. Lam, P. Shi, E. K. Boukas and Y. Zou, Guaranteed cost control for uncertain neutral stochastic systems via dynamic output feedback controllers, *J. of Optimization Theory and Applications*, vol.143, no.1, pp.207-223, 2009.

[5] J. Wang, J. Wang, W. Yuan and P. Shi, Gain-scheduled guaranteed cost control of LPV systems with time-varying state and input delays, *International Journal of Innovative Computing Information and Control*, vol.5, no.10(B), pp.3377-3389, 2009.

[6] J. Zhang, P. Shi and J. Qiu, Non-fragile guaranteed cost control for uncertain stochastic nonlinear time-delay systems, *J. of the Franklin Institute*, vol.346, pp.676-690, 2009.

[7] P. Shi, X. Luan and F. Liu, $H_\infty$ filtering for discrete-time systems with stochastic incomplete measurement and mixed delays, *IEEE Trans. on Industrial Electronics*, vol.59, no.6, pp.2732-2739, 2012.

[8] J. Kennedy and R. Eberhart, Particle swarm optimization, *Proc. of the IEEE International Conference on Neural Networks*, Perth, Australia, pp.1942-1948, 1995.

[9] D. Karaboga, An idea based on honeybee swarm for numerical optimization, *Technical Report TR06*, Erciyes University, 2005.

[10] M. Dorigo and G. D. Carok, Ant colony optimization: A new meta-heuristic, *Proc. of Congress of Evolutionary Computing*, pp.1470-1477, 1999.

[11] L. N. D. Castro and F. J. V. Zuben, Artificial immune system: Part 1 – Basic theory and applications, *Technical Report RtDca 01/99, Feec/Unicamp*, Brazil, 1999.

[12] P. Acharjee and S. K. Goswami, Expert algorithm based on adaptive particle swarm optimization for power flow analysis, *Expert Systems with Applications*, vol.36, no.3, pp.5151-5156, 2009.

[13] J. Zhang, C. Zhang and S. Liang, The circular discrete particle swarm optimization algorithm for flow shop scheduling problem, *Expert Systems with Applications*, vol.37, no.8, pp.5827-5834, 2010.

[14] T. Sousa, A. Silva and A. Neves, Particle swarm based data mining algorithms for classification tasks, *Parallel Computing*, vol.30, no.5-6, pp.767-783, 2004.

[15] F. Du, W. Shi, L. Chen et al., Infrared image segmentation with 2-d maximum entropy method based on particle swarm optimization (PSO), *Pat. Recog. Letters*, vol.26, no.5, pp.597-603, 2005.

[16] Z. Lian, X. Gu and B. Jiao, A similar particle swarm optimization algorithm for permutation flowshop scheduling to minimize makespan, *Applied Mathematics and Computation*, vol.175, no.1, pp.773-785, 2006.

[17] S. H. Zahiri and S. A. Seyedin, Swarm intelligence based classifiers, *Journal of the Franklin Institute*, vol.344, no.5, pp.362-376, 2007.

[18] Z. Chang and W. W. Ping, An improved PSO-based rule extraction algorithm for intrusion detection, *Proc. of the 2009 International Conference on Computational Intelligence and Natural Computing*, Wuhan, China, pp.56-58, 2009.

[19] I. D. Falco, A. D. Cioppa and E. Tarantino, Facing classification problems with particle swarm optimization, *Applied Soft Computing*, vol.7, no.3, pp.652-658, 2007.

[20] Z. Wang, X. Sun and D. Zhang, Classification rule mining based on particle swarm optimization *LNCS*, vol.4062, pp.436-441, 2006.

[21] H. Liu, A. Abraham and W. Zhang, A fuzzy adaptive turbulent particle swarm optimization, *International Journal of Innovative Computing and Applications*, vol.1, no.1, pp.39-47, 2007.

[22] S. Stevens, J. Volkman and E. Newman, A scale for the measurement of the physiological magnitude pitch, *Journal of Acoustical Society of America*, vol.8, no.3, pp.185-190, 1937.

[23] M. Xu, L. Duan, J. Cai et al., HMM-based audio keyword generation, *Lecture Notes in Computer Science*, vol.3333, pp.566-574, 2004.

[24] M. Sahidullah and G. Saha, Design, analysis and experimental evaluation of block based transformation in MFCC computation for speaker recognition, *Speech Communication*, vol.54, no.4, pp.543-565, 2012.

[25] C. Bae, Y. Y. Chung, M. A. M. Shukran et al., An intelligent classification algorithm for life log multimedia applications, *Proc. of the IEEE International Workshop on Multimedia Signal Processing, MMSP 2008*, Cairns, Queensland, Australia, pp.558-562, 2008.

[26] E. Keogh and C. A. Ratanamahatanat, Exact indexing of dynamic time warping, *Knowledge and Information Systems*, vol.7, no.3, pp.358-386, 2004.

[27] R. C. Eberhart and Y. Shi, Particle swarm optimization: Developments, applications and resources, *Proc. of the 2001 Congress on Evolutionary Computation*, Seoul, South Korea, pp.81-86, 2001.

[28] W. C. Yeh, A two-stage discrete particle swarm optimization for the problem of multiple multi-level redundancy allocation in series systems, *Expert System with Applications*, vol.36, no.5, pp.9192-9200, 2009.

[29] W. C. Yeh, W. W. Chang and Y. Y. Chung, A new hybrid approach for mining breast cancer pattern using discrete particle swarm optimization and statistical method, *Expert System with Applications*, vol.36, no.4, pp.8204-8211, 2009.

[30] C. Bae, W.-C. Yeh, N. Wahid, Y. Y. Chung and Y. Liu, A new simplified swarm optimization (SSO) using exchange local search scheme, *International Journal of Innovative Computing, Information and Control*, vol.8, no.6, pp.4391-4406, 2012.

[31] J. H. Ang, K. C. Tan and A. A. Mamun, An evolutionary memetic algorithm for rule extraction, *Expert System with Applications*, vol.37, no.2, pp.1302-1315, 2009.

[32] E. S. Correa, A. A. Freitas and C. G. Johnson, A new discrete particle swarm algorithm applied to attribute selection in a bioinformatics data set, *Proc. of GECCO'06*, Seattle, Washington, USA, pp.35-42, 2006.

[33] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*, 2nd Edition, Morgan Kaufmann, San Francisco, 2005.

[34] R. S. Parpinelli, H. S. Lopes and A. A. Freitas, Data mining with an ant colony optimization algorithm, *IEEE Transactions on Evolutionary Computation*, vol.6, no.4, pp.321-332, 2002.

[35] A. P. Schmidt and T. K. M. Stone, *Music Classification and Identification System*, www.trevors tone.org/school/MusicRecognitionDatabase.pdf, 2010.

[36] X. He, L. Guo, X. Zhou et al., Hybrid support vector machine and general model approach for audio classification, *LNCS*, vol.4493, pp.434-440, 2007.

[37] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines*, Cambridge University Press, 2000.

[38] J. Platt, *Fast Training of Support Vector Machines Using Sequential Minimal Optimization*, MIT Press, Cambridge, MA, 1988.