Hindawi Publishing Corporation
Computational Intelligence and Neuroscience
Volume 2015, Article ID 947098, 14 pages
http://dx.doi.org/10.1155/2015/947098

Research Article

Training Spiking Neural Models Using Artificial Bee Colony

Roberto A. Vazquez[1] and Beatriz A. Garro[2]

[1]*Intelligent Systems Group, Faculty of Engineering, La Salle University, Benjamín Franklin 47, Colonia Condesa, 06140 Mexico City, DF, Mexico*
[2]*Instituto en Investigaciones en Matemáticas Aplicadas y en Sistemas, Universidad Nacional Autónoma de México, Ciudad Universitaria, 04510 Mexico City, DF, Mexico*

Correspondence should be addressed to Roberto A. Vazquez; ravem@lasallistas.org.mx

Received 18 October 2014; Accepted 6 January 2015

Academic Editor: Jianwei Shuai

Spiking neurons are models designed to simulate, in a realistic manner, the behavior of biological neurons. Recently, it has been proven that this type of neurons can be applied to solve pattern recognition problems with great efficiency. However, the lack of learning strategies for training these models do not allow to use them in several pattern recognition problems. On the other hand, several bioinspired algorithms have been proposed in the last years for solving a broad range of optimization problems, including those related to the field of artificial neural networks (ANNs). Artificial bee colony (ABC) is a novel algorithm based on the behavior of bees in the task of exploring their environment to find a food source. In this paper, we describe how the ABC algorithm can be used as a learning strategy to train a spiking neuron aiming to solve pattern recognition problems. Finally, the proposed approach is tested on several pattern recognition problems. It is important to remark that to realize the powerfulness of this type of model only one neuron will be used. In addition, we analyze how the performance of these models is improved using this kind of learning strategy.

1. Introduction

Artificial neural networks (ANNs) are applied in a broad range of problems. Among the most popular tasks using ANN, we could mention pattern recognition, forecasting, and regression problems. However, the accuracy of these models could drastically diminish if the topology is not well-designed and if the training algorithm is not selected carefully. One interesting alternative to designing the topology and training and exploiting the capabilities of an ANN is to adopt a learning strategy based on evolutionary and swarm intelligence algorithms. It is well-known that designing and training tasks can be stated as optimization problems; for that reason, it is possible to apply different types of evolutionary and swarm intelligence algorithms. For example, particle swarm optimization [1] and differential evolution [2] have been used to design and train ANNs automatically.

Several swarm intelligence algorithms based on the collective behavior of self-organizing systems have been proposed in the last years. Among the most popular, we could mention ant colony system (ACO) [3] and particle swarm optimization (PSO) [1] and artificial bee colony (ABC) [4]. Most of the studies related to honey bee swarm are focused on the dance and communication, task allocation, collective decision, nest site selection, mating, marriage, reproduction, foraging, floral and pheromone laying, and navigation behaviours of the swarm [5]. ABC is a novel algorithm that tries to mimic the behavior of the bees in nature, which tasks consist in exploring their environment to find a food source.

The ABC algorithm has been used in a broad range of optimization problems. This algorithm is relatively simple, and its implementation is straightforward for solving optimization problems, being able to produce acceptable results at a low computational cost. The different studies performed in the literature compare its efficiency against other traditional strategies such as genetic algorithm (GA), differential evolution (DE), particle swarm optimization

(PSO), ant colony optimization (ACO), and their variants. The efficiency obtained on numerical problems using numerical test functions, multivariable functions, constrained and unconstrained optimization problems, and even multiobjective problems, suggests that the ABC algorithm is a serious candidate for training ANN [5].

Some of the first works that use the ABC algorithm to adjust the synaptic weight of an ANN are described in [6, 7]. In [8] the authors train a feed-forward ANN using ABC applied to solve the XOR, 3-bit parity, and 4-bit encoder-decoder problem, and some signal processing applications. In [9], the ANN is trained using ABC algorithm to solve a medical pattern classification problem. In [10], the authors train an ANN to classify different dataset utilized in the machine learning community. In [11], the ANN is trained for the classification of the acoustic emission signal to their respective source. Another interesting paper for designing and training an ANN is presented in [12], where the authors described a methodology for maximizing its accuracy and minimizing its connections by evolving the weights, the architecture, and the transfer function of each neuron. In the context of forecasting, [13] used ABC to train an ANN for bottom hole pressure prediction in underbalanced drilling. Whereas in [14] the authors train a recurrent ANN for stock price forecasting, in [15] the author uses ABC for training an ANN for earthquake time series data. Reference [16] presents an ANN trained with ABC for S-models of biomedical networks approximation. From all these papers the authors conclude that ABC algorithm is capable of training ANN with an acceptable accuracy and, in some cases, the results are better than those obtained with other traditional techniques.

Moreover, there exist many algorithms based on the bees' behavior such as bee algorithm, honey-bee mating algorithm, and bee colony optimization (BCO), among others [17]. Hence, there are investigations about training ANNs that use different kinds of algorithms related to the ABC. For example, in [18] the authors apply the bee colony algorithm to train an ANN, which later is applied to the wood's defect problem. In [19], the authors estimate the state variables in distribution networks, including distributed generators using a honey-bee mating algorithm. Furthermore, in [20], the authors use this algorithm in combination with a self-organizing map (SOM) in the market segmentation problem.

Swarm intelligence algorithms have contributed and gained popularity in the field of ANN as a learning strategy. However, the intrinsic limitations of ANN do not allow applying them in complex pattern recognition problems, even using different learning strategies. These limitations motivate to explore other alternatives to model and generate neural models to make possible their application in several pattern recognition problems.

Although ANNs were inspired by the behavior of the human brain, the fact is that they do not mimic the behavior of a biological neuron. In that sense, the development and application of more realistic neural models could improve the accuracy of an ANN during several pattern recognition tasks.

Spiking neuron models are called the 3rd generation of artificial neural networks [21]. These neurons increase the level of realism in a neural simulation and integrate the concept of time. These types of models have been used in a broad range of areas, mainly from the field of computational neurosciences [22], brain region modeling [23], auditory processing [24, 25], visual processing [26–28], robotics [29, 30], and so on.

Several spiking models have been proposed in the last years. One of the most realistic and complex models was proposed in [31]. Nonetheless, there are simplified versions of this model that reduce its computational complexity. Among these models, we could mention the well-known integrate-and-fire model [32], Izhikevich model [33], FitzHugh-Nagumo model [34], and Hindmarsh-Rose [35].

Theoretically, these types of models could simulate the behavior of any perceptron type neural network. However, their application in computer vision and pattern recognition has not been widely explored. Although there are some works related to image segmentation [36–38] and pattern recognition [39–43], there still are several issues to research related to the learning process, design, and implementation.

The process of learning of these models is conducted with different techniques. In [44], the authors present the Spike-Prop, an adaptation of the well-known backpropagation algorithm to train a spiking neural model. Furthermore, several variants to improve the efficiency of spike-prop have been proposed [45–47]. However, these algorithms require a careful tuning of the network to obtain acceptable results.

Another approach to train these models is based on probabilistic models [47–49], information bottleneck learning [50–52], and reinforcement learning [53–55].

On the other hand, nongradient based methods like evolutionary strategies (such as GA, PSO, and DE) have emerged as an alternative to traditional methods for training spiking neural models. Although this approach is computationally more expensive compared with traditional methods, it has several advantages that make possible its application in real pattern recognition problems [42, 56, 57].

Recently, it has been proven that only one spiking neuron model can solve nonlinear pattern recognition problems, showing a clear advantage against the traditional perceptron [58–60]. One alternative to simulate the learning process of this type of model is to use swarm intelligence algorithms. For example, in [58] the authors describe an approach to applying a leaky-integrate-and-fire spiking neuron in various linear and nonlinear pattern recognition problems. In that work, the authors use the differential evolution algorithm as a learning strategy. In other researches, the authors use the Izhikevich spiking model to the same set of problems using a differential evolution strategy [60]. In [61, 62], the authors use the Izhikevich spiking model to the same set of problem using cuckoo search and particle swarm optimization algorithms, respectively. In general, the methodology described in those papers can be stated as follows: given a set of input patterns belonging to K classes, first of all, each input pattern is transformed into an input signal. Then the spiking neuron is stimulated during T ms and the firing rate is computed. After adjusting the synaptic weights of the spiking model by means of a swarm intelligence algorithm, we expect that input patterns from the same class produce similar firing rates,

and input patterns from different classes generate firing rates different enough to discriminate among the categories.

Despite the results presented in those papers, it is still necessary to explore and develop strategies that allow these models to learn from their environment and improve their accuracy solving complex pattern recognition problems. Due to the capabilities of producing acceptable results at a low computational cost in a broad range of optimization problems, including ANN field, the ABC algorithm could be an excellent tool for simulating the learning process of a spiking neural model.

In this paper, we proposed to use the ABC algorithm as a learning strategy to training a spiking neuron model aiming to perform various linear and nonlinear pattern recognition problems. Based on the methodology described in [60, 61], we present a comparison of the results presented in [58, 60, 61] to determine which learning strategy provides the best accuracy and how it affect the accuracy of the spiking neuron. In order to test the accuracy of the learning strategy combined with the spiking neuron model, we perform several experiments using different pattern recognition problems, including an odor recognition problem and cancer classification based on DNA microarrays.

The outline of this paper is divided into five sections. A brief introduction to the ABC algorithm is presented in Section 2. The concepts related to the third generation of neural networks knowing as spiking neural networks are presented in Section 3. Section 4 presents the proposed methodology for training spiking neural networks using the ABC algorithm. The experimental results, as well as the discussion of the results, are presented in Section 5. Finally, the conclusions of this work are presented in Section 6.

2. Basics on Artificial Bee Colony

The artificial bee colony (ABC) algorithm is a novel approach in the area of the swarm optimization proposed by Karaboga and Akay [6]. The ABC algorithm is based on the behavior of bees in nature, whose task consists of exploring their environment to find a food source, picking up the flower's nectar and returning to the hive in order to evaluate the quality and the amount of the food, and then call the other bees of the community to fly towards the food source. Communication among bees is done by a particular dance.

This algorithm can find the optimum values in the search space of a given optimization problem. A global optimization problem can be defined as finding the parameter vector $\mathbf{x}$ that minimizes the objective function $f(\mathbf{x})$:

$$\text{minimize } f(\mathbf{x}), \quad \mathbf{x} = (x_1, x_2, \ldots, x_i, \ldots, x_{n-1}, x_n) \in \mathbb{R}^n \tag{1}$$

which is constrained by the following inequalities and/or equalities:

$$l_i \leq x_i \leq u_i, \quad i = 1, \ldots, n. \tag{2}$$

$f(\mathbf{x})$ is defined on a search space, S, which is n dimensional rectangle in $\mathbb{R}^n$. The variable domains are limited by their lower and upper bounds (2).

This algorithm represents the solutions of a given problem by means of the position of different food sources visited by a bee. Furthermore, it works with three kinds of bees in order to explore and exploit the search space: employed, onlookers, and scouts bees.

Following the work described in [7], the employed bee has to modify the position (solution) in its memory based on the local information (visual information) and test the nectar amount (fitness value) of the new source (new solution). If the quantity of nectar in the new position is better than the old one, the bee memorizes it and forgets the old one. Contrarily, it keeps the previous one in its memory. After the entire employed bees complete the search process, they share the nectar information about the food sources and their position, with the onlooker bees in the dance area.

An onlooker bee checks the nectar information obtained from all employed bees and selects a food source with a probability in terms of its nectar amount. The employed bee modifies the position in its memory and checks the quantity of nectar obtained from the candidate source. If its nectar is higher than that of the previous one, then the bee memorizes the new position and forgets the old one.

An artificial onlooker bee chooses a food source depending on the probability p_i associated with that food source. This probability is calculated with the following expression:

$$p_i = \frac{\text{fit}_i}{\sum_{n=1}^{\text{SN}} \text{fit}_n}, \tag{3}$$

where fit_i is the fitness value of the solution i of dimension D which is proportional to the nectar amount of the food source in the position i and SN is the number of food sources that is equal to the number of employed bees.

In order to produce a candidate food position from the old one in memory, the ABC algorithm uses the following expression:

$$v_{ij} = x_{ij} + \phi_{ij}\left(x_{ij} - x_{kj}\right), \tag{4}$$

where $k \in \{1, 2, \ldots, \text{SN}\}$ and $j \in \{1, 2, \ldots, D\}$ are indexes randomly chosen. Although k is determined randomly, it has to be different from i. ϕ_{ij} is a random number between $[-1, 1]$ that controls the production of neighbor food sources around x_{ij}.

The food source whose nectar is discarded by the bees is changed with a new food source by the scouts. In the ABC algorithm, this is simulated by producing a random position and replacing it with the abandoned one. If that position cannot be enhanced after a number of trials, then the solution is discarded. This parameter is called the "limit" for abandonment. Assume that the abandoned source is x_i and $j \in \{1, 2, \ldots, D\}$; then the scout bee discovers a new food source to be replaced with x_i. This operation can be defined as

$$x_{ij} = l_j + \text{rand}(0, 1)\left(u_j - l_j\right), \tag{5}$$

where l_j and u_j are the lower and upper bounds of the parameter x_{ij}, respectively.

(1) Initialize the population of solutions x_i $\forall i = 1, \ldots, \mathrm{SN}$
(2) Evaluate the population x_i $\forall i = 1, \ldots, \mathrm{SN}$
(3) **for** *cycle* = 1 to MCN **do**
(4) Produce new solutions v_i for the employed bees by using (4), verify boundaries and evaluate them.
(5) Apply the greedy selection process.
(6) Calculate the probability values p_i for the solutions x_i by (3).
(7) Produce the new solutions v_i for the onlookers from the solutions x_i selected depending on p_i, verify boundaries and evaluate them.
(8) Apply the greedy selection process.
(9) Determine the abandoned solution for the scout, if exist, and replace it with a new randomly produced solution x_i by (5).
(10) Memorize the best solution achieved so far.
(11) $cycle = cycle + 1$
(12) **end for**

Pseudocode 1

After each candidate source position v_{ij} is generated and then evaluated by the artificial bee, its performance is compared to the performance associated with the previous position. If the new food source has equal or better nectar than the old one, then it is substituted with a new one in the memory. Otherwise, the old one is retained in the memory.

The pseudocode of the ABC algorithm is shown in Pseudocode 1.

The ABC algorithm randomly initializes a population of solutions that represent the position of food sources within the lower and upper bounds. The size of the population is a parameter that corresponds to the number of food sources that is equal to the number of employed or onlooker bees. The stop criterion adopted in this algorithm is the maximum cycle number. The solutions are limited by their lower and upper bounds (2). If they are out of the boundaries, they are set to the lower or upper bounds.

One advantage of this algorithm is that only three control parameters are needed: population size (SN), the maximum cycle number (MCN), and the value of the "limit."

3. Spiking Neural Models

The main distinctive elements that compose a typical spiking neuron are: dendrites (*input device*), soma (*central processing unit*), and axon (*output device*). The dendrites collect signals from other neurons and transmit them to the soma. The soma performs an important nonlinear processing step where an output signal is generated if the total input exceeds a certain threshold. Finally, the output signal is taken over by the axon, which delivers the signal (short electrical pulses called action potentials or spikes) to other neurons. Typically, the spikes have an amplitude of about 100 mV and a duration of 1-2 ms [63]. Although the same elements exist in a linear perceptron, the main difference between a linear perceptron and a spiking model is the action potential generated during the stimulation time. Furthermore, the activation function used in spiking models is a differential equation that tries to model the dynamic properties of a biological neuron in terms of spikes.

According to [63], a spike train is a sequence of stereotyped events generated at regular or irregular intervals. The form of the spike does not carry any information, and what is important is the number and the timing of spikes. The shortest distance between two spikes defines the absolute refractory period of the neuron that is followed by a phase of relative refractoriness where it is difficult to generate a spike.

Several spiking models have been proposed in the last century aiming to model different neurodynamic properties of neurons [64]. Among these models, we could mention the well-known integrate-and-fire model, resonate-and-fire, Izhikevich model, FitzHugh-Nagumo model, and Hodgkin-Huxley model.

One of the most simple and versatile models is the one proposed by Izhikevich. This model has only nine dimensionless parameters, and it is described with the next equation:

$$\begin{aligned} C\dot{v} &= k\left(v - v_r\right)\left(v - v_t\right) - u + I \quad \text{if } v \geq v_{\text{peak}} \text{ then} \\ \dot{u} &= a\left\{b\left(v - v_r\right) - u\right\} \quad v \longleftarrow c,\ u \longleftarrow u + d. \end{aligned} \tag{6}$$

Depending on the values of a and b, the spiking model can act as an integrator or a resonator. Whereas a is the recovery time constant, the sign of b force to u behaves as an amplifying ($b < 0$) or a resonant ($b > 0$) variable. The parameters c and d consider that the action of high-threshold voltage-gated currents activated during the spike affects only the after-spike transient behavior. In addition, c define the voltage reset value, and d describes the total amount of outward minus inward currents activated during the spike. On the other hand, the variable v represents the membrane potential, u is the recovery current, C is the membrane capacitance, v_r is the resting membrane potential, v_t is the instantaneous threshold potential, and v_{peak} is the spike cutoff value [65].

According to [33], the spiking model can reproduce various intrinsic firing patterns based on different values of the parameters. Regular spiking (RS) neurons are the most typical neurons in the cortex; see Figure 1. For the intrinsically bursting (IB) behavior, the neurons fire a stereotypical burst of spikes followed by repetitive single spikes. Whereas the neurons with a chattering (CH) behavior can fire stereotypical bursts of closely spaced spikes, neurons with a fast-spiking (FS) behavior can fire periodic trains of action

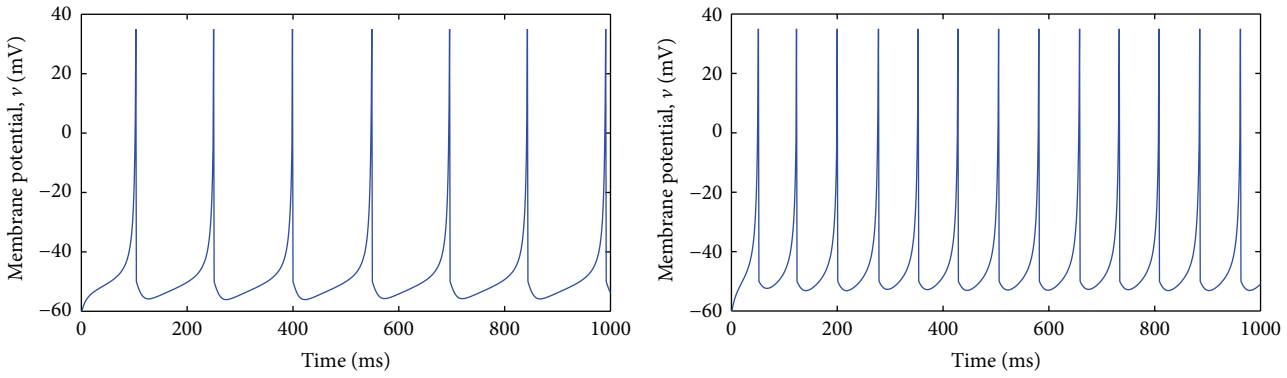

Figure 1: Simulation of the Izhikevich neuron model $100\dot{v} = 0.7(v + 60)(v + 40) - u + I$, $\dot{u} = 0.03\{-2(v + 60) - u\}$, if $v \geq 35$ then $v \leftarrow -50$ and $u \leftarrow u + 100$. (a) Injection of the step of DC current $I = 70$ pA. (b) Injection of the step of DC current $I = 100$ pA.

potentials with extremely high frequency practically without any adaptation (slowing down). Finally, for the low-threshold spiking (LTS), the neurons can also fire high-frequency spike trains but with a notable spike frequency adaptation.

A profound description of the Izhikevich model can be found in [65]. In this paper, we will concentrate on the parameters that produce regular spiking patterns. However, we do not discard the use in the near future of other firing patterns, useful for solving pattern recognition tasks. The Izhikevich model can produce a class 1 neural excitability behavior: action potentials can be generated with arbitrary low firing rate, depending on the strength of the applied current [65]. The response of the Izhikevich neuron changes if the input current changes generating different firing rates; see Figure 1.

The neuron is stimulated during T ms with an input signal, and it fires when its membrane potential reaches a specific value and thus generating an action potential (spike) or a train of spikes.

4. Proposed Method

Based on the hypothesis "patterns from the same class produce similar firing rates in the output of the spiking neuron and patterns from other classes produce firing rates different enough to discriminate among the classes," the authors in [58, 60] proposed a methodology which describes the way that a spiking neuron can be applied to solve pattern recognition problems.

Following the same approach, let $D = \{\mathbf{x}^i, k\}_{i=1}^{p}$ be a set of associations composed of p input patterns, where $k = 1, \ldots, K$ is the class to which $\mathbf{x}^i \in \mathbb{R}^n$ belongs. The learning process adjusts the synaptic values of the spiking model in such way that the output generates a different firing rate for each class k, reproducing the behavior described in the hypothesis.

4.1. Classifying with Firing Rates. This subsection describes how a spiking neural model can be applied in a pattern classification task based on [58, 60].

In order to use a spiking neural model to solve a pattern classification problem, it is necessary to compute the input current I that stimulates the model. In other words, the spiking neuron model is not directly stimulated with the input pattern $\mathbf{x}^i \in \mathbb{R}^n$ but with the input current I. If we assume that each feature of the input pattern $\mathbf{x}^i$ corresponds to the presynaptic potential of different receptive fields, then we can calculate the input current I that stimulates the spiking neuron as

$$I = \mathbf{x} \cdot \mathbf{w}, \tag{7}$$

where $\mathbf{w}^i \in \mathbb{R}^n$ is the set of synaptic weights of the neuron model. This input current is used in the methodology to stimulate the spiking model during T ms.

Instead of using the spike train generated by the spiking model to perform the pattern classification tasks, we compute the firing rate of the neuron defined as

$$\text{fr} = \frac{N_{\text{sp}}}{T}, \tag{8}$$

where N_{sp} is the number of spikes that occur within the time window of lenght T.

It is necessary to calculate the average firing rate $\text{AFR} \in \mathbb{R}^K$ of each class, by using the firing rates produced by each input pattern. In this sense, the learning process consists of finding the synaptic values of the spiking model in such way that it generates a different average firing rate for each class k.

Suppose that the spiking neuron is already trained using a learning strategy. To determine the class to which an unknown input pattern $\mathbf{x}$ belongs, it is necessary to compute the firing rate generated by the trained spiking neuron. After that, the firing rate is compared against the average firing rate of each class. The minimum difference between the firing rate and the average firing rates determines the class of an unknown pattern. This is expressed with the following equation:

$$\text{cl} = \arg \min_{k=1}^{K} \left(\left| \text{AFR}_k - \text{fr} \right| \right), \tag{9}$$

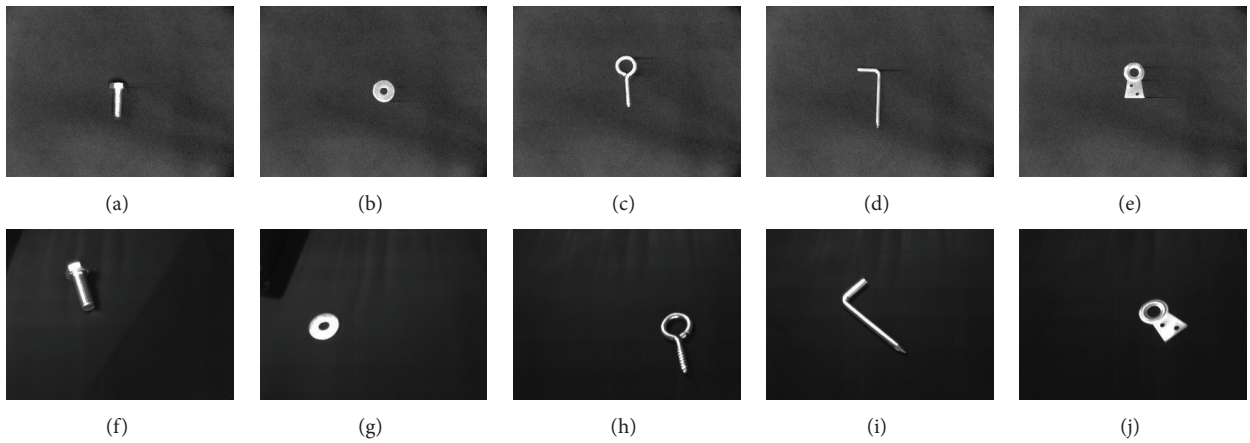

Figure 2: (a)–(d) Some of the images used to train the proposed method. (f)–(i) Some of the images were used to test the proposed method.

where fr is the firing rate generated by the neuron model stimulated with the unknown input pattern $\tilde{\mathbf{x}}$.

4.2. Adjusting Synapses of the Spiking Neuron Model. In order to achieve the desired behavior at the output of the spiking neuron, it is necessary to adjust its synaptic weights. This step corresponds to the learning (training) phase of the spiking neuron. Several swarm intelligence algorithms can be used in the learning phase, but in this research we focus on the artificial bee colony (ABC) algorithm as a learning strategy.

The synapses of the neuron model $\mathbf{w}$ are adjusted by means of the ABC algorithm. The food source in the ABC algorithm represents the synaptic weights of the spiking model. In order maximize the accuracy of the spiking neuron model during a pattern recognition task, the best set of synaptic weights must be found using the ABC algorithm. However, if we state the problem as a minimization problem, the classification error must be minimized. The fitness function that uses the classification error to find the set of synaptic weights is defined as

$$f(\mathbf{w}, D) = 1 - \text{performance}(\mathbf{w}, D), \tag{10}$$

where $\mathbf{w}$ are the synapses of the spiking model, D is the set of input patterns, and performance($\mathbf{w}$, D) is a function which computes the classification accuracy, in terms of (9), given by

$$\text{performance}(\mathbf{w}, D) = \frac{P_{\text{cc}}}{P_t}, \tag{11}$$

where P_{cc} denotes the number of patterns correctly classified and P_t denotes the number of tested patterns.

5. Experimental Results

In this section, we analyze and discuss the results generated with the proposed methodology. In order to evaluate the accuracy of the proposed methodology, we perform several experiments using different datasets. The section is divided into three subsections. The first section presents a comparison of the proposed methodology against different swarm intelligence algorithms. In the last two subsections, we present preliminary results applying spiking neural models using the proposed methodology for solving an odor classification problem and cancer classification based on DNA microarrays.

5.1. Analysis and Comparison of Experimental Results. In order to evaluate the classification accuracy, when the spiking neuron is trained using the ABC algorithm, six sets of experiments were performed. Each set of experiments was performed with a dataset that corresponds to a specific pattern recognition problem. The iris plant, glass, diabetes, liver-bupa, and wine datasets were taken from the UCI machine learning benchmark repository [66].

The iris plant dataset is a three-class problem composed of input patterns with four features. The wine dataset also is a three-class problem composed input patterns with 13 features. The glass dataset is a six-class problem composed of patterns with nine features; however, in this experiment we used the two most representative classes. The diabetes dataset is a two-class problem composed of input patterns with eight features. Finally, the liver dataset also is a two-class problem composed of input patterns with six features.

Furthermore, an object recognition problem, composed of five different objects, was used to evaluate the accuracy of the proposal; see Figure 2 [67]. A dataset was generated from a set of 100 images in different positions, rotations, and scale changes. Instead of recognizing objects directly from their images, an invariant description of each object was calculated applying the next process over each image: a standard threshold [68] was applied to get its binary version; small spurious regions were eliminated from each image by means of a size filter [69]; finally, the seven well-known Hu moments invariant to translations, rotations, and scale changes [70] were computed over each image. As a result of this process, we obtained a five-class dataset problem composed of patterns with seven features.

To reproduce the regular spiking behavior, the parameters for the Izhikevich neuron model were set as $C = 100$, $v_r = -60$, $v_t = -40$, $v_{\text{peak}} = 35$, $k = 0.7$, $a = 0.03$, $b = -2$, $c = -50$,

Table 1: Statistical results obtained with the proposed methodology.

Dataset	Mean		Confidence intervals	
	Tr. cr.	Te. cr.	TR	TE
Wine	0.963 ± 0.015	0.878 ± 0.038	[0.957–0.968]	[0.864–0.892]
Iris plant	0.996 ± 0.006	0.957 ± 0.025	[0.994–0.999]	[0.948–0.967]
Glass	0.832 ± 0.029	0.703 ± 0.064	[0.822–0.843]	[0.679–0.727]
Diabetes	0.800 ± 0.016	0.743 ± 0.024	[0.794–0.806]	[0.734–0.752]
Liver	0.749 ± 0.024	0.688 ± 0.034	[0.740–0.757]	[0.675–0.700]
Object recognition	1.000 ± 0.000	0.990 ± 0.021	[1.000-1.000]	[0.982–0.998]

Tr. cr = training classification rate, Te. cr. = testing classification rate.

and $d = 100$. To solve the differential equation, Euler method with $dt = 1$ with a simulation time of $T = 1000$. Finally, the parameters for the artificial bee colony algorithm were set to SN = 40, MAXGEN = 1000, LIM = 100, and the boundaries for the solution to $XMAX = 10$ and $XMIN = -10$.

The classification rate of the model was computed in terms of the number of input patterns correctly classified divided by the total number of tested input patterns. 30 experiments over each dataset were performed to validate the accuracy of the spiking neuron when the ABC algorithm is used as a learning strategy. In order to train the spiking neural model, 80% of the samples (training subset) were randomly selected to do so, the remaining samples (testing subset) were used during the testing phase.

At the beginning of the evolutionary learning process, we observed that the learning error rapidly converges to a stable error and after a certain number of generations the learning error changes at a slower rate. Although this behavior was observed in whole experiments, the learning error achieved with the ABC algorithm was not good enough for all the problems. For some problems, the achieved learning error was highly acceptable. In Figure 3 is shown how the learning error evolves through each generation of the ABC algorithm.

After the spiking neuron was trained, we proceed to evaluate the accuracy of the neuron using the remaining 20% samples. From these experiments, we can observe that the performance of the spiking neuron trained with the ABC algorithm was highly acceptable for the six pattern recognition problems. The average percentage of classification computed from all experiments is shown in Table 1. Something that should be remarked is that these methods only used one Izhikevich neuron model. Furthermore, we present the results using the t-distribution test to construct the confidence intervals around the mean.

We observed that the classification rate achieved for the iris, wine, and object recognition problems using the spiking neuron trained with the ABC algorithm during the training phase was greater than 96%. However, the classification rate achieved during the testing phase most of the times diminished. Nonetheless, the classification rate was higher than 87%.

On the contrary, for the case of the glass and diabetes problems, we observed that whereas the classification rate achieved using the spiking neuron trained with the ABC algorithm during the training phase was not greater than 84%, the classification rate for the liver problem was not greater than 75%. As a consequence, the accuracy achieved with the spiking neuron drastically diminished during the testing phase. Nonetheless, the results obtained with the spiking neuron trained with the ABC algorithm were acceptable.

Figures 4(a), 4(b), and 4(c) show the experimental results obtained with the wine, iris plant, and object recognition datasets. Each dot represents the time when the neuron produces a spike. As the reader can observe, the synaptic weights found with the ABC algorithm provoke that the Izhikevich neuron generates almost the same firing rate when it is stimulated with patterns from the same class. Furthermore, the Izhikevich neuron generates firing rates different enough to discriminate patterns that belong to other classes.

On the other hand, Figures 4(d), 4(e), and 4(f) show the experimental results obtained with the glass, diabetes, and liver datasets, respectively. For this pattern recognition problems, the set of synaptic weights found with the ABC algorithm during the training phase were not good enough for the spiking neuron to generate a similar firing rate when it is stimulated with patterns from the same class; as a result, some patterns belonging to the same class were classified in a different class. Although the Izhikevich neuron does not generate firing rates different enough to discriminate among patterns from different classes, the results achieved with the proposed method were quite acceptable.

In addition, in Table 2, we compare the behavior of the methodology using three different swarm intelligence algorithms as a learning strategy: differential evolution (DE) [58, 60], cuckoo search (CS) [61], and particle swarm optimization (PSO) [62].

Furthermore, we perform a comparison against the well-known feedforward neural network (FNN). Two different topologies (1 and 2 hidden layers) and two different training algorithms (descendant gradient and Levenberg-Marquardt) [71] were used to design the FNN. The parameters for training algorithms were set as 0.1 for the learning rate, 2000 epochs for the stop criterion. For each combination (topology-learning algorithm), 30 experiments were done to prove the obtained results statistically. For each experiment, we randomly select the 70% and 10% of the samples for both, training and validation phases. The remaining 20% of the samples were chosen for the testing phase.

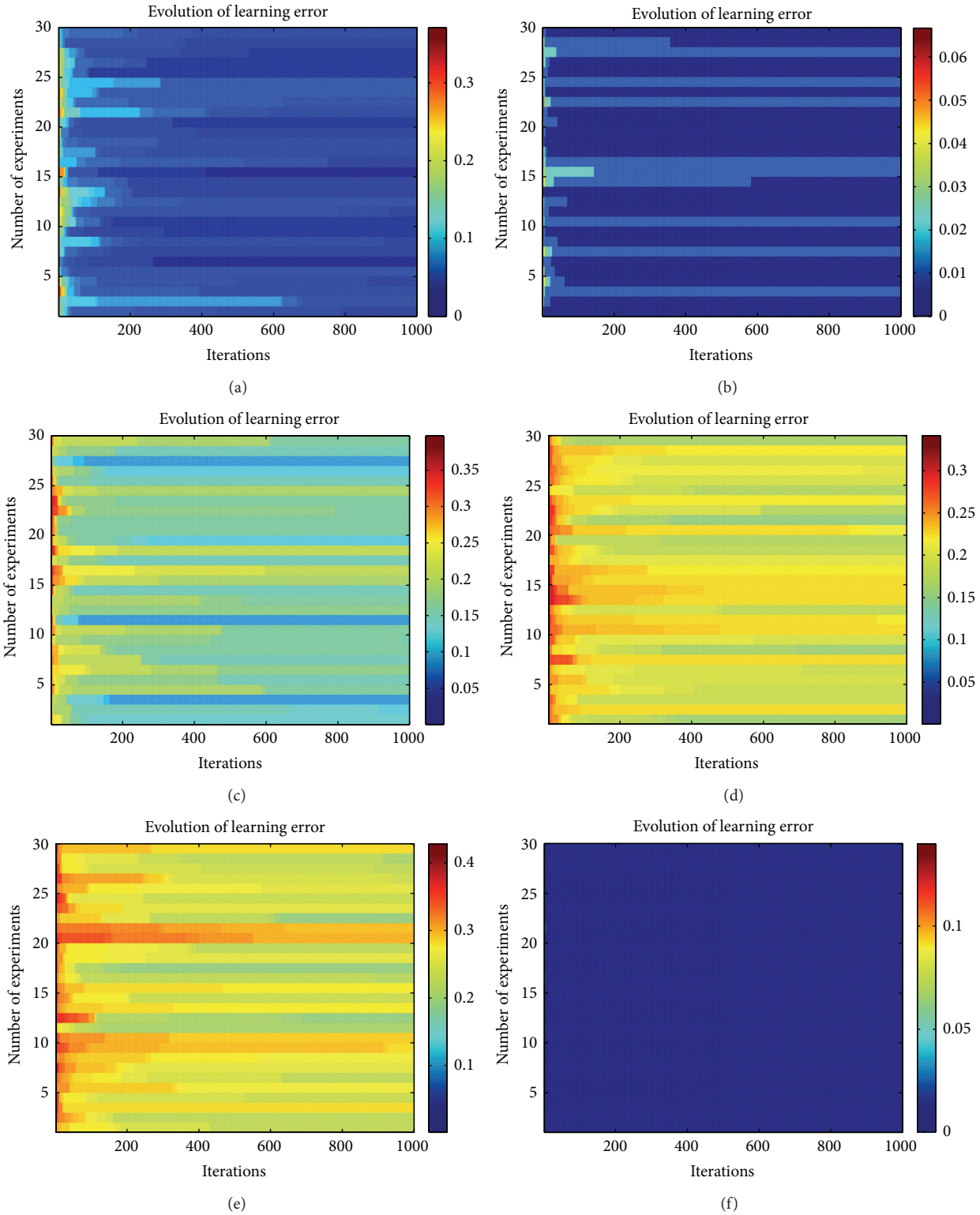

Figure 3: Training error achieved by the spiking model during the learning phase using ABC algorithm. (a) 30 experiments using the wine dataset are shown. (b) 30 experiments using the iris dataset are shown. (c) 30 experiments using the glass dataset are shown. (d) 30 experiments using the diabetes dataset are shown. (e) 30 experiments using the liver dataset are shown. (f) 30 experiments using the object recognition dataset are shown.

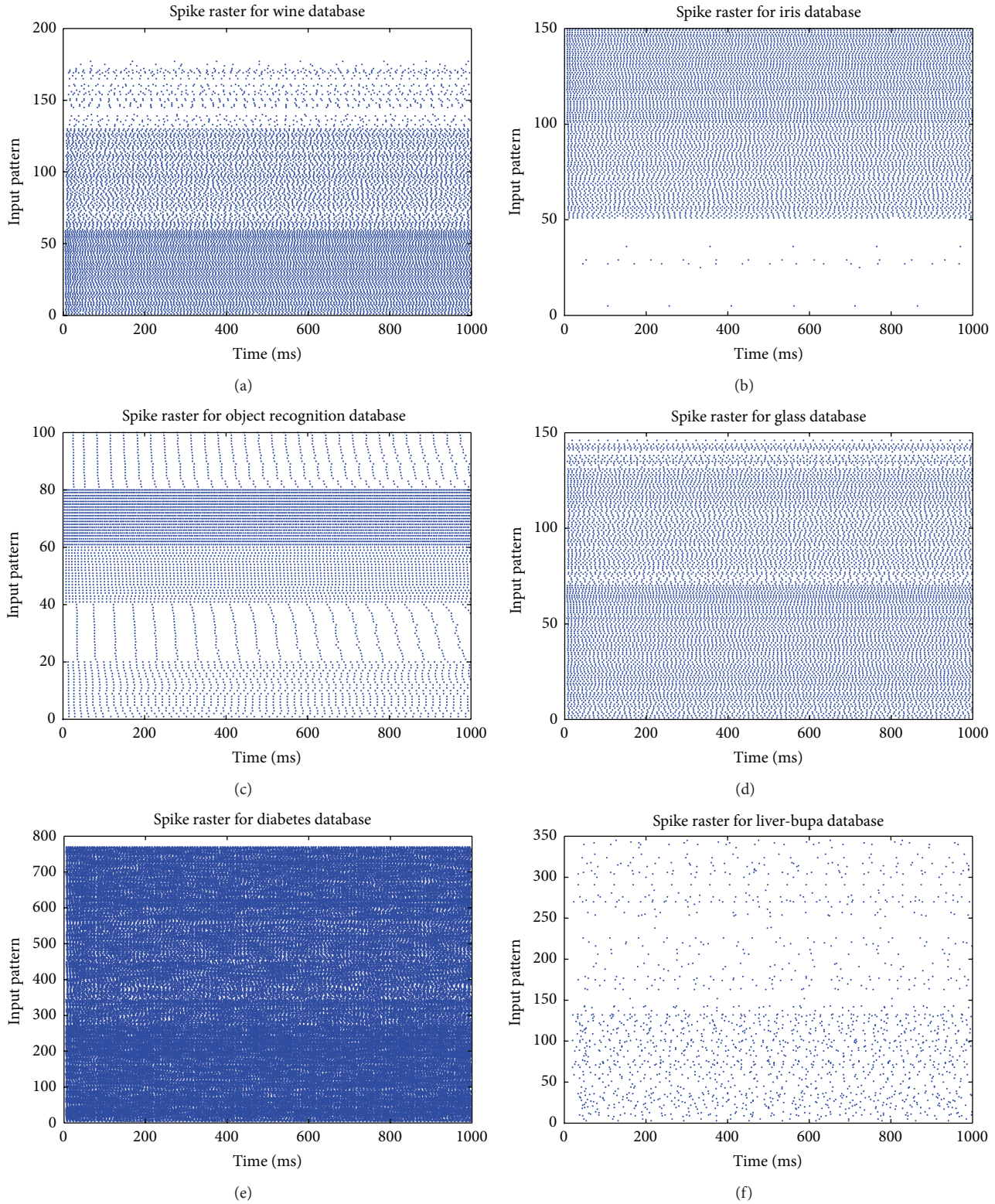

Figure 4: Experimental results obtained with proposed methodology for different datasets. Notice that different firing rates which correspond to different classes can be observed. Each dot represents the time that the neuron generate an spike. (a) Wine dataset. (b) Iris plant dataset. (c) Object recognition dataset. (d) Glass dataset. (e) Diabetes dataset. (f) Liver dataset.

Table 2: Average accuracy provided by the methods using different databases.

Dataset	Method using DE		Method using PSO		Method using CS	
	Tr. cr.	Te. cr.	Tr. cr.	Te. cr.	Tr. cr.	Te. cr.
Wine	0.9796	0.8744	0.9782	0.8879	0.9831	0.9078
Iris plant	0.9933	0.9833	0.9933	0.97	0.9942	0.9467
Glass	0.8158	0.7411	0.8178	0.7457	0.8080	0.7646
Diabetes	0.8038	0.7371	0.7990	0.7619	0.8051	0.7477
Liver	0.7620	0.6870	0.7591	0.6754	0.7609	0.6536
Object recognition	1	0.9850	1	0.9950	1	1

Tr. cr = training classification rate, Te. cr. = testing classification rate.

Table 3: Average accuracy provided by the FNN using different databases.

Dataset	FNN DG-1		FNN DG-2		FNN LM-1		FNN LM-2	
	Tr. cr.	Te. cr.	Tr. cr.	Te. cr.	Tr. cr.	Te. cr.	Tr. cr.	Te. cr.
Wine	0.9867	0.9800	0.9497	0.9171	0.9986	0.9714	0.9797	0.9629
Iris plant	0.9133	0.8867	0.6625	0.6367	0.9942	0.9767	0.7667	0.7733
Glass	0.6907	0.6738	0.6390	0.6310	0.8453	0.7714	0.7413	0.7119
Diabetes	0.7663	0.7732	0.7153	0.7144	0.7993	0.7320	0.7850	0.7588
Liver	0.5924	0.6145	0.5928	0.5623	0.7243	0.6652	0.7181	0.6812
Object recognition	0.7413	0.6800	0.7413	0.6800	0.7413	0.6800	0.7413	0.6800

Tr. cr = training classification rate, Te. cr. = testing classification rate.

The number of neurons that compose the hidden layers depends on the classification problem. The next two topologies were used in combination with each dataset.

(1) Wine dataset: 1 HL composes 8 neurons. 2 HL compose 5 and 3 neurons, respectively.

(2) Iris plant dataset: 1 HL composes 4 neurons. 2 HL compose 2 and 2 neurons, respectively.

(3) Glass dataset: 1 HL composes 8 neurons. 2 HL compose 5 and 3 neurons, respectively.

(4) Diabetes dataset: 1 HL composes 5 neurons. 2 HL compose 3 and 2 neurons, respectively.

(5) Object recognition dataset: 1 HL composes 6 neurons. 2 HL compose 4 and 2 neurons, respectively.

Table 3 shows the classification efficiencies using feedforward neural networks (FNN). Each column contains the obtained results using a combination of topology and learning algorithms during training and testing phases. Label DG-1 means that the descendant gradient method and one hidden layer (HL) were used to design the FNN, and label LM-2 indicates that Levenberg-Marquardt method and two hidden layers were used to create the FNN, and so on.

The results obtained with the spiking neuron model trained with the ABC algorithm are similar to those obtained in [58, 60–62]. Although with some dataset the ABC algorithm provides better results than DE and PSO algorithm and vice versa, we could not say that one learning strategy is better than the other.

In fact, swarm intelligence algorithms can be considered as a learning strategy to adjust the synaptic weights of a third-generation neural model. After comparing the different algorithms, we note that they present the same behavior and are serious candidates to be considered as a learning strategy. These results suggest that swarm intelligence algorithms combined with spiking neurons can be regarded as an alternative way to perform various pattern recognition tasks. After several experiments, we observed that if we want to improve the efficiency of the methodology, it is not enough to change the learning strategies because they provide similar efficiencies. Nonetheless, the results obtained are highly acceptable.

On the other hand, the reader can observe in Table 2 that the proposed methodology provides better results than traditional FNN composed of several neurons. These results, obtained with the described methodology, show a visible advantage of a spiking model against FNN.

We can also conjecture that if only one neuron is capable of solving pattern recognition problems, perhaps several spiking neurons working together can improve the experimental results obtained in this research.

5.2. Application of the Proposed Methodology in a Odor Recognition Problem. In this subsection, we present some preliminary results obtained with the proposed methodology in a problem related to odor recognition. The dataset used was built in Bermak and Martinez [72] where the authors developed an automated gas delivery experimental setup for extracting volatile compounds at given concentrations from liquids, composed of two pumps, two mass flow controllers

Table 4: Statistical results obtained with the proposed methodology.

Dataset	Mean		Confidence intervals	
	Tr. cr.	Te. cr.	TR	TE
Odor	0.952 ± 0.024	0.898 ± 0.064	[0.943–0.961]	[0.874–0.922]

Tr. cr = training classification rate, Te. cr. = testing classification rate.

Table 5: Statistical results obtained with the proposed methodology.

Dataset	Mean		Confidence intervals	
	Tr. cr.	Te. cr.	TR	TE
Cancer DNA	0.883 ± 0.024	0.625 ± 0.097	[0.875–0.892]	[0.588–0.661]

Tr. cr = training classification rate, Te. cr. = testing classification rate.

(MFCs), one bubbler, a gas chamber, and a data acquisition system. This dataset contains patterns obtained from ethanol and butanol vapors injected into the gas chamber at a flow rate determined by the mass flow controllers. The authors also used sensor arrays composed of five commercial TGS Figaro gas sensors (TGS 2600, 2602, 2610, 2611, and 2620). The potential differences across the sensor resistances were measured using a voltage divider with 2.2 k load resistors while keeping the heating voltage constant to 5 V. Finally, the sensors output voltages were sampled at a rate of 10 Hz and quantized with an 11 bit analog to digital converter to build a dataset composed of 124 patterns with 5 features each pattern.

The parameters for the Izhikevich neuron and for the artificial bee colony algorithm and the number of experiments as well as the samples used for training and testing phases were set as equal as in the previous subsection.

Table 4 presents the results achieved with the proposed methodology applied to an odor recognition problem. According to the results presented in Table 4, the reader can observe that the accuracy of the spiking neuron is highly acceptable.

These results obtained with the proposed methodology confirm that spiking neurons can be considered as a pattern recognition technique useful in a wide range of applications, including odor recognition.

5.3. Application of the Proposed Methodology in Cancer Classification Based on DNA Microarrays. In this subsection, we present some preliminary results obtained with the proposed methodology for a problem related to cancer classification based on DNA microarrays.

DNA microarrays are a powerful technique in genetic science due to the possibility to analyze the gene expression level of millions of genes at the same time. However, the main problem that arises when DNA microarrays are analyzed with computational intelligent techniques is that the number of genes is too big, and the samples are too few [73].

In that sense, we evaluate the capabilities of a spiking neuron for predict a disease based on the DNA microarrays dataset. It is important to mention that in order to capture the real behavior of the spiking neuron, we do not apply any dimensional reduction technique over the dataset.

The dataset used in this paper was built in [74] where the authors described a generic approach to cancer classification based on gene expression monitoring by DNA microarrays that discover the distinction between acute myeloid leukemia (AML) and acute lymphoblastic leukemia (ALL). The dataset consists of 38 bone marrow samples for training (27 ALL and 11 AML), over 7129 probes from 6817 human genes. Also, 34 samples testing data are provided, with 20 ALL and 14 AML.

The parameters for the Izhikevich neuron and for the artificial bee colony algorithm and the number of experiments were set as equal as in the previous subsection.

Table 5 present the results achieved with the proposed methodology applied to a cancer recognition problem using DNA microarrays. According to the results presented in Table 5, the reader can observe that the accuracy of the spiking neuron is highly acceptable during training phases; however during the testing phase, the accuracy drastically diminished.

These results obtained with the proposed methodology are preliminary and suggest that only one spiking neuron is not enough to solve this problem. Nonetheless, if we combine several spiking neurons to build a network or apply a dimensional reduction stage before training the spiking model, the accuracy could increases. It is important to remark that few highly dimensional samples were used to train the spiking neural model.

6. Conclusion

In this paper, we described how the artificial bee colony algorithm can be used as a learning strategy to train a spiking neural model. The results obtained with this approach suggest that ABC algorithm is a useful alternative to adjusting the synaptic weights of the spiking neuron model. Furthermore, we observed that the spiking neuron model provides acceptable results during the pattern recognition task, regardless of the swarm intelligence algorithms used as a learning strategy, the spiking neuron model provides acceptable results during the pattern recognition task.

Although we found that the spiking neuron did not behave as good as we desired with the six datasets, we could say that, in general, the behavior achieved with the spiking neuron provokes that patterns belonging to the same class generate almost the same firing rate in the output of the spiking neuron, and patterns belonging to different classes produce firing rates different enough to discriminate among the different classes. Thanks to the swarm intelligence algorithm used during the evolutionary learning process, the spiking model behaves like we previously described in our starting hypothesis.

Furthermore, a comparison among the ABC, DE, CS, and PSO algorithms was performed. In general, we observe that the accuracy obtained with the spiking neuron model trained with the ABC algorithm is comparable to the accuracy obtained with the methods described in [58, 60–62]. We could not say that one learning strategy is better than the other because, with some datasets, the ABC algorithm

provides better results than DE and PSO algorithm and vice versa.

This research provides a clear idea of how powerful a spiking neuron model is in a pattern classification task. Furthermore, the results obtained with the odor recognition and cancer classification problem based on DNA microarrays were highly encouraged.

Nowadays, we are testing different spiking neuron models. Despite the encouraging results achieved with this methodology, we are developing a new method to evolve the synaptic weights at the same time not only for one spiking neuron, but for several spiking neurons as well as to adjust the parameters of the models.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The authors would like to thank Universidad La Salle for the economic support under Grant no. ULSA I-061/12. Beatriz Garro thanks CONACYT for the posdoctoral scholarship.

References

[1] B. A. Garro, H. Sossa, and R. A. Vazquez, "Design of artificial neural networks using a modified particle swarm optimization algorithm," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN '09)*, pp. 938–945, June 2009.

[2] B. A. Garro, H. Sossa, and R. A. Vázquez, "Design of artificial neural networks using differential evolution algorithm," in *Proceedings of the 17th International Conference on Neural Information Processing: Models and Applications—Part II (ICONIP '10)*, vol. 6444 of *Lecture Notes in Computer Science*, pp. 201–208, Springer, Berlin, Germany, 2010.

[3] M. Dorigo and C. Blum, "Ant colony optimization theory: a survey," *Theoretical Computer Science*, vol. 344, no. 2-3, pp. 243–278, 2005.

[4] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm," *Journal of Global Optimization*, vol. 39, no. 3, pp. 459–471, 2007.

[5] D. Karaboga, B. Gorkemli, C. Ozturk, and N. Karaboga, "A comprehensive survey: artificial bee colony (ABC) algorithm and applications," *Artificial Intelligence Review*, vol. 42, no. 1, pp. 21–57, 2014.

[6] D. Karaboga and B. Akay, "Artificial Bee Colony (ABC) algorithm on training artificial neural networks," in *Proceedings of the IEEE 15th Signal Processing and Communications Applications (SIU '07)*, pp. 1–4, IEEE, Eskişehir, Turkey, June 2007.

[7] D. Karaboga and B. Basturk, "Artificial bee colony (abc) optimization algorithm for solving constrained optimization problems," in *Proceedings of the 12th International Fuzzy Systems Association World Congress on Foundations of Fuzzy Logic and Soft Computing (IFSA '07)*, pp. 789–798, Springer, Cancun, Mexico, 2007.

[8] D. Karaboga, B. Akay, and C. Ozturk, "Artificial bee colony (ABC) optimization algorithm for training feed-forward neural networks," in *Proceedings of the 4th International Conference on Modeling Decisions for Artificial Intelligence (MDAI '07)*, pp. 318–329, Springer, 2007.

[9] D. Karaboga, C. Ozturk, and B. Akay, "Training neural networks with abc optimization algorithm on medical pattern classification," in *Proceedings of the International Conference on Multivariate Statistical Modelling and High Dimensional Data Mining*, 2008.

[10] D. Karaboga and C. Ozturk, "Neural networks training by artificial bee colony algorithm on pattern classification," *Neural Network World*, vol. 19, no. 3, pp. 21–57, 2009.

[11] S. N. Omkar and J. Senthilnath, "Artificial bee colony for classification of acoustic emission signal source," *International Journal of Aerospace Innovations*, vol. 1, no. 3, pp. 129–143, 2009.

[12] B. A. Garro, H. Sossa, and R. A. Vazquez, "Artificial neural network synthesis by means of artificial bee colony (ABC) algorithm," in *Proceedings of the IEEE Congress of Evolutionary Computation (CEC '11)*, pp. 331–338, New Orleans, La, USA, June 2011.

[13] R. Irani and R. Nasimi, "Application of artificial bee colony-based neural network in bottom hole pressure prediction in underbalanced drilling," *Journal of Petroleum Science and Engineering*, vol. 78, no. 1, pp. 6–12, 2011.

[14] T. J. Hsieh, H. F. Hsiao, and W. C. Yeh, "Forecasting stock markets using wavelet transforms and recurrent neural networks: an integrated system based on artificial bee colony algorithm," *Applied Soft Computing*, vol. 11, no. 2, pp. 2510–2525, 2011.

[15] H. Shah, R. Ghazali, and N. M. Nawi, "Using artificial bee colony algorithm for MLP training on earthquake time series data prediction," *Journal of Computing*, vol. 3, no. 6, pp. 135–142, 2011.

[16] W.-C. Yeh and T.-J. Hsieh, "Artificial bee colony algorithm-neural networks for S-system models of biochemical networks approximation," *Neural Computing and Applications*, vol. 21, no. 2, pp. 365–375, 2012.

[17] D. Karaboga and B. Akay, "A survey: algorithms simulating bee swarm intelligence," *Artificial Intelligence Review*, vol. 31, no. 1–4, pp. 61–85, 2009.

[18] D. T. Pham, A. J. Soroka, A. Ghanbarzadeh, E. Koc, S. Otri, and M. Packianather, "Optimising neural networks for identification of wood defects using the bees algorithm," in *Proceedings of the IEEE International Conference on Industrial Informatics (INDIN '06)*, pp. 1346–1351, IEEE, Singapore, August 2006.

[19] T. Niknam, "Application of honey-bee mating optimization on state estimation of a power distribution system including distributed generators," *Journal of Zhejiang University: Science A*, vol. 9, no. 12, pp. 1753–1764, 2008.

[20] B. Amiri and M. Fathian, "Integration of self organizing feature maps and honey bee mating optimization algorithm for market segmentation," *Journal of Theoretical and Applied Information Technology*, vol. 3, pp. 70–86, 2007.

[21] W. Maass, "Networks of spiking neurons: the third generation of neural network models," *Neural Networks*, vol. 10, no. 9, pp. 1659–1671, 1997.

[22] F. Rieke, D. Warland, Robert, and W. Bialek, *Spikes—Exploring the Neural Code*, 1997.

[23] M. E. Hasselmo, C. Bodelón, and B. P. Wyble, "A proposed function for hippocampal theta rhythm: separate phases of encoding and retrieval enhance reversal of prior learning," *Neural Computation*, vol. 14, no. 4, pp. 793–817, 2002.

[24] J. J. Hopfield and C. D. Brody, "What is a moment? 'Cortical' sensory integration over a brief interval," *Proceedings of the*

National Academy of Sciences of the United States of America, vol. 97, no. 25, pp. 13919–13924, 2000.

[25] S. Loiselle, J. Rouat, D. Pressnitzer, and S. Thorpe, "Exploration of rank order coding with spiking neural networks for speech recognition," in *Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN '05)*, vol. 4, pp. 2076–2080, Québec, Canada, July-August 2005.

[26] H. Azhar, K. Iftekharuddin, and R. Kozma, "A chaos synchronization-based dynamic vision model for image segmentation," in *Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN '05)*, vol. 5, pp. 3075–3080, 2005.

[27] S. J. Thorpe, R. Guyonneau, N. Guilbaud, J.-M. Allegraud, and R. VanRullen, "SpikeNet: real-time visual processing with one spike per neuron," *Neurocomputing*, vol. 58–60, pp. 857–864, 2004.

[28] R. A. Vazquez, B. Girau, and J. C. Quinton, "Visual attention using spiking neural maps," in *Proceedings of the International Joint Conference on Neural Network (IJCNN '11)*, pp. 2164–2171, August 2011.

[29] E. A. di Paolo, "Spike-timing dependent plasticity for evolved robots," *Adaptive Behavior*, vol. 10, no. 3-4, pp. 243–263, 2003.

[30] D. Floreano, J.-C. Zufferey, and J.-D. Nicoud, "From wheels to wings with evolutionary spiking circuits," *Artificial Life*, vol. 11, no. 1-2, pp. 121–138, 2005.

[31] A. L. Hodgkin, "The local electric changes associated with repetitive action in a non-medullated axon," *The Journal of Physiology*, vol. 107, no. 2, pp. 165–181, 1948.

[32] L. F. Abbott, "Lapicque's introduction of the integrate-and-fire model neuron (1907)," *Brain Research Bulletin*, vol. 50, no. 5-6, pp. 303–304, 1999.

[33] E. M. Izhikevich, "Simple model of spiking neurons," *IEEE Transactions on Neural Networks*, vol. 14, no. 6, pp. 1569–1572, 2003.

[34] R. FitzHugh, "Impulses and physiological states in theoretical models of nerve membrane," *Biophysical Journal*, vol. 1, no. 6, pp. 445–466, 1961.

[35] R. M. Rose and J. L. Hindmarsh, "The assembly of ionic currents in a thalamic neuron. I. The three-dimensional model," *Proceedings of the Royal Society B: Biological Sciences*, vol. 237, no. 1288, pp. 267–288, 1989.

[36] P. Rowcliffe, J. Feng, and H. Buxton, "Clustering within integrate-and-fire neu rons for image segmentation," in *Artificial Neural Networks—ICANN 2002*, J. Dorronsoro, Ed., vol. 2415 of *Lecture Notes in Computer Science*, pp. 69–74, Springer, Berlin, Germany, 2002.

[37] B. Meftah, O. Lezoray, and A. Benyettou, "Segmentation and edge detection based on spiking neural network model," *Neural Processing Letters*, vol. 32, no. 2, pp. 131–146, 2010.

[38] Q. X. Wu, T. M. McGinnity, L. P. Maguire, A. Belatreche, and B. Glackin, "Processing visual stimuli using hierarchical spiking neural networks," *Neurocomputing*, vol. 71, no. 10–12, pp. 2055–2068, 2008.

[39] A. Belatreche, L. P. Maguire, and T. M. McGinnity, "Pattern recognition with spiking neural networks and dynamic synapse," in *Proceedings of the International FLINS Conference on Applied computational intelligence*, pp. 205–210, 2004.

[40] J. J. Wade, L. J. McDaid, J. A. Santos, and H. M. Sayers, "SWAT: a spiking neural network training algorithm for classification problems," *IEEE Transactions on Neural Networks*, vol. 21, no. 11, pp. 1817–1830, 2010.

[41] Q. Yu, K. Tan, and H. Tang, "Pattern recognition computation in a spiking neural network with temporal encoding and learning," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN '12)*, pp. 1–7, IEEE, Brisbane, Australia, June 2012.

[42] H. N. Abdull Hamed, N. Kasabov, Z. Michlovský, and S. M. Shamsuddin, "String pattern recognition using evolving spiking neural networks and quantum inspired particle swarm optimization," in *Neural Information Processing: 16th International Conference, ICONIP 2009, Bangkok, Thailand, December 1–5, 2009, Proceedings, Part II*, vol. 5864 of *Lecture Notes in Computer Science*, pp. 611–619, Springer, Berlin, Germany, 2009.

[43] N. Kasabov, "Evolving spiking neural networks for spatio-and spectro-temporal pattern recognition," in *Proceedings of the 6th IEEE International Conference Intelligent Systems (IS '12)*, pp. 27–32, September 2012.

[44] S. M. Bohte, J. N. Kok, and H. la Poutré, "Error-backpropagation in temporally encoded networks of spiking neurons," *Neurocomputing*, vol. 48, no. 1–4, pp. 17–37, 2002.

[45] B. Schrauwen and J. van Campenhout, "Extending SpikeProp," in *Proceedings of the IEEE International Joint Conference on Neural Networks*, vol. 1, pp. 471–475, July 2004.

[46] J. Xin and M. J. Embrechts, "Supervised learning with spiking neural networks," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN '01)*, vol. 3, pp. 1772–1777, July 2001.

[47] R. P. N. Rao, "Hierarchical bayesian infere nce in networks of spiking neurons," in *Advances in Neural Information Processing Systems*, vol. 17, MIT Press, 2005.

[48] S. Deneve, "Bayesian spiking neurons. I: inference," *Neural Computation*, vol. 20, no. 1, pp. 91–117, 2008.

[49] N. Kasabov, "To spike or not to spike: a probabilistic spiking neuron model," *Neural Networks*, vol. 23, no. 1, pp. 16–19, 2010.

[50] D. Barber, "Learning in spiking neural assemblies," in *Proceedings of the 16th Annual Neural Information Processing Systems Conference (NIPS '02)*, December 2002.

[51] G. Chechik, "Spike-timing-dependent plasticity and relevant mutual information maximization," *Neural Computation*, vol. 15, no. 7, pp. 1481–1510, 2003.

[52] J. P. Pfister, T. Toyoizumi, D. Barber, and W. Gerstner, "Optimal spike-timing-dependent plasticity for precise action potential firing in supervised learning," *Neural Computation*, vol. 18, no. 6, pp. 1318–1348, 2006.

[53] H. S. Seung, "Learning in spiking neural networks by reinforcement of stochastic synaptic transmission," *Neuron*, vol. 40, no. 6, pp. 1063–1073, 2003.

[54] E. M. Izhikevich, "Solving the distal reward problem through linkage of STDP and dopamine signaling," *Cerebral Cortex*, vol. 17, no. 10, pp. 2443–2452, 2007.

[55] R. Legenstein, D. Pecevski, and W. Maass, "A learning theory for reward-modulated spike-timing-dependent plasticity with application to biofeedback," *PLoS Computational Biology*, vol. 4, no. 10, Article ID e1000180, 2008.

[56] S. Schliebs, M. Defoin-Platel, S. Worner, and N. Kasabov, "Integrated feature and parameter optimization for an evolving spiking neural network: exploring heterogeneous probabilistic models," *Neural Networks*, vol. 22, no. 56, pp. 623–632, 2009, Advances in Neural Networks Research: International Joint Conference on Neural Networks (IJCNN'09).

[57] A. Cachón and R. A. Vázquez, "Tuning the parameters of an integrate and fire neuron via a genetic algorithm for solving

pattern recognition problems," *Neurocomputing*, vol. 148, pp. 187–197, 2015.

[58] R. A. Vazquez and A. Cachón, "Integrate and Fire neurons and their application in pattern recognition," in *Proceedings of the 7th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE '10)*, pp. 424–428, September 2010.

[59] R. Vazquez, "Izhikevich neuron model and its application in pattern recognition," *Australian Journal of Intelligent Information Processing Systems*, vol. 11, no. 1, pp. 35–40, 2010.

[60] R. A. Vázquez, "Pattern recognition using spiking neurons and firing rates," in *Advances in Artificial Intelligence—IBERAMIA 2010: Proceedings of the 12th Ibero-American Conference on AI, Bahía Blanca, Argentina, November 1–5, 2010*, vol. 6433 of *Lecture Notes in Computer Science*, pp. 423–432, Springer, Berlin, Germany, 2010.

[61] R. A. Vazquez, "Training spiking neural models using cuckoo search algorithm," in *Proceedings of the IEEE Congress of Evolutionary Computation (CEC '11)*, pp. 679–686, June 2011.

[62] R. A. Vázquez and B. A. Garro, "Training spiking neurons by means of particle swarm optimization," in *Advances in Swarm Intelligence: 1st International Conference, ICSI 2010, Beijing, China, June 12–15, 2010, Proceedings*, vol. 6728 of *Lecture Notes in Computer Science*, Part I, pp. 242–249, Springer, Berlin, Germany, 2011.

[63] W. Gerstner and W. M. Kistler, *Spiking Neuron Models: Single Neurons, Populations, Plasticity*, Cambridge University Press, Cambridge, UK, 2002.

[64] E. M. Izhikevich, "Which model to use for cortical spiking neurons?" *IEEE Transactions on Neural Networks*, vol. 15, no. 5, pp. 1063–1070, 2004.

[65] E. M. Izhikevich, *Dynamical Systems in Neuroscience: the Geometry of Excitability and Bursting*, Computational Neuroscience, MIT Press, 2007.

[66] P. M. Murphy and D. W. Aha, "UCI repository of machine learning databases," Tech. Rep., Department of Information and Computer Science, University of California, Irvine, Calif, USA, 1994.

[67] R. A. V. E. de Los Monteros and J. H. S. Azuela, "A new associative model with dynamical synapses," *Neural Processing Letters*, vol. 28, no. 3, pp. 189–207, 2008.

[68] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[69] R. Jain and R. K. B. G. Schunck, *Machine Vision*, McGraw-Hill, 1995.

[70] M.-K. Hu, "Visual pattern recognition by moment invariants," *IEEE Transactions on Information Theory*, vol. 8, no. 2, pp. 179–187, 1962.

[71] M. I. A. Lourakis, *A Brief Description of the Levenberg-Marquardt Algorithm Implemented by Levmar*, Foundation for Research and Technology-Hellas, Vassilika Vouton, Crete, GREECE, 2005.

[72] A. Bermak and D. Martinez, "A compact 3D VLSI classifier using bagging threshold network ensembles," *IEEE Transactions on Neural Networks*, vol. 14, no. 5, pp. 1097–1109, 2003.

[73] B. A. Garro, R. A. Vazquez, and K. Rodríguez, "Classification of DNA microarrays using artificial bee colony (ABC) algorithm," in *Advances in Swarm Intelligence*, Y. Tan, Y. Shi, and C. Coello, Eds., vol. 8794 of *Lecture Notes in Computer Science*, pp. 207–214, Springer, 2014.

[74] T. R. Golub, D. K. Slonim, P. Tamayo et al., "Molecular classification of cancer: class discovery and class prediction by gene expression monitoring," *Science*, vol. 286, no. 5439, pp. 531–527, 1999.