USING META-HEURISTIC ALGORITHM IN SPIKING NEURAL NETWORK FOR PATTERN RECOGNITION TASKS

REGINA ESI TURKSON[1,2], SICHAO LIU[1], EDWARD Y. BAAGYERE[1], MOSES J. EGHAN[2]

[1]School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, 611731, China.
[2]Department of Computer Science and Information Technology, University of Cape Coast, P.M.B University Post Office. Cape Coast, Ghana.
E-MAIL: regina_turkson@yahoo.com, 201821081124@std.uestc.edu, ybaagyere@uds.edu.gh, meghan@ucc.edu.gh

Abstract:

The Bat Algorithm (BA) is a meta-heuristic algorithm based on echolocation behavior of microbats. The authors propose BA based Spiking Neural Network (SNN) model, where the advantages of BA and efficiency of SNN are exploited for classification tasks using some benchmark datasets. The advantages of the BA have been well exploited in the Artificial Neural Networks (ANN) domain particularly with the adjustment of weights. We therefore, leveraged on the BA as a learning strategy to train an SNN using the Leaky Integrate and Fire (LIF) and Izhikevich models to solve non-linear pattern classification tasks. In order to successfully discriminate between the various classes, the models are trained to fire at the same or similar firing rate for inputs from the same class, and inputs patterns from different classes to also spike or fire at different rate. To justify how efficient and how powerful the proposed model is, only one neuron is used. Finally, the model is tested on different non-linear pattern recognition tasks and comparison is made between our model and other similar existing models and our proposed model outperformed some of the state-of-the-art-models. To the best of our knowledge, this is the first work to implement BA in SNN.

Keywords:

Spiking Neural Network; Bat Algorithm; Meta-heuristic; Pattern Recognition

1. Introduction

Spiking Neural Networks (SNNs), have several proposed models [1] and their application in classification task [2] and computer vision has not been adequately explored, making SNN a nature candidate to be applied in ANN. Several meta-heuristic algorithms such as Particle Swarm Optimization (PSO) [3], Differential Evolution (DE) [4], Artificial Bee Colony (ABC) algorithm [5], Cuckoo Search (CS) Algorithm [6] have been proposed in recent years for fine-tuning the synaptic weight of ANNs, for designing topology and selecting transfer function of neurons. These meta-heuristic methods are typically based on local search, population methods, and cooperative co-evolutionary models. Meta-heuristic algorithms provide optimal solution even in a complex search space and have the ability of escaping from the problem of local minima or maxima [7] and have the potential of producing highly accurate and robust solutions in the shortest time. Bat Algorithm [7] is one of such meta-heuristic, nature-inspired algorithm and has been applied in classification tasks [8], optimization problems [9], ANN problems [10] and others. BA is a combination of the major advantages of GA, PSO and harmony search algorithms. Earlier works reveal that BA outperforms PSO and GA in providing solution to unconstrained optimization problems [11].

The work in [12] presented an evolving SNNs of artificial creatures using GA. In [13], a quantum PSO was used to train a string pattern recognition in SNN. The authors in [14] and [15] presented an SNN trained with PSO algorithm. ABC algorithm was also used as learning strategy to train SNN to perform various linear and non-linear pattern recognition problems in [16]. The work in [17] used CS algorithm to fine-tune the synaptic weights of neuron to solve non-linear and a real object recognition problem. In [18-19], DE was used as a learning strategy to demonstrate how only LIF or Izhikevich can be applied to solve different pattern recognition problems. Regardless of the results presented in these literatures, it is still prudent to explore and develop strategies that allow these models to learn from their environment. We proposed a new concept of applying a different meta-heuristic algorithm in SNN for classification tasks. Implementing BA to simulate the learning process in SNN will be appropriate since BA has the competency of producing satisfactory results at low computational cost in optimization problems and ANN.

In this work we propose a new concept of using BA as a learning strategy to fine-tune the synaptic weights of SNN

978-1-7281-4242-5/19/$31.00 ©2019 IEEE

by exploring the behavior of BA on only one neuron model for classification tasks. Using similar methodology as in [18] and [19], we apply BA to both LIF and Izhikevich models and compare results of similar works in [15-19] to our proposed model. The results demonstrated the capabilities of BA for solving non-linear pattern recognition problems. The efficiency of this work is verified on benchmark classification problems. To the best of our knowledge, this is the first work to implement BA in SNN.

2. The BAT Algorithm

Xin-She Yang was the first to propose a novel BA [7]. Many variations of BA [20-21] have since been proposed, the basic principle that, through echolocation the bat keeps updating its velocity, position and frequency to catch its prey in shortest span remains unchanged. BA possesses the potentials of automatic zooming and automatic switching from explorative moves to local intensive exploitation, allowing BA to have a quick convergence particularly at the early stages of iteration. The work in [22] gives detailed description of the BA.

3. The Spiking Neural Model

The spiking nature of biological neurons has led to studies of the computation power and more biological plausibility of the brain to uncover the mystery of how the brain works and to design human-like intelligent system [23, 24]. We adopt two SNN models, namely the LIF [18] and the Izhikevich [19] models in our work.

3.1. The Leaky-Integrate-and-Fire Model

The Leaky-Integrate-and Fire (LIF) model is one of the most widely used neurons in computational neuroscience due to it simple nature and easier implementation. We adopt a variant of the LIF model as presented in [25] in our work and is defined in Equation 1;

$$v' = I + a - bv \quad \text{if} \ v \geq v_{thresh} \quad \text{then} \ v \leftarrow c \tag{1}$$

where I denotes the input current of the neuron, v_{thresh} is the threshold for firing the spike and c is the reset voltage.

3.2. The Izhikevich Model

The Izhikevich model is one of the most simple and versatile models in SNN and has nine (9) dimensionless parameters. The model is presented as shown in Equation 2:

$$C\overset{g}{v} = k(v - v_r)(v - v_t) - u + I \quad \text{if} \ v \geq v_{peak} \quad \text{then}$$

$$\overset{g}{u} = a\{b(v - v_r) - u\} \quad v \leftarrow c, u \leftarrow u + d \tag{2}$$

A detailed description of the model can be found in [19].

4. Our Proposed Method

The proposed method is inspired by the methodology described by authors in [18, 19] based on the hypothesis that patterns belonging to the same class generates same or similar firing rates in the output of the SNN and patterns belonging to other classes generate firing rates different enough to distinguish among the classes. Changes in an input current signal, results in changes in the reply of the neuron thereby generating dissimilar firing rates. A firing rate is computed by dividing the number of spikes generated in an interval of duration Tms by the length of the time window T. The neuron is stimulated during Tms with an input signal and spikes when its membrane potential crosses a specific threshold generating an action potential or a train of spikes.

Let $\{x^i, k\}_{i=1}^{p}$ be a set of p inputs patterns with $k = 1, 2..., K$ denote the class to which $x^i \in \mathbb{R}^n$ belongs. Each input pattern is first altered into an input signal I, the spiking neuron then stimulates I during Tms and computes the firing rate of the neuron afterwards. Once the firing rate has been obtained, the average firing rate $AFR \in \mathbb{R}^K$ of each class can then be computed. We propose the adoption of BA as the learning strategy to fine-tune the synaptic weight to generate the spiking neurons. After training, the firing rates generated by each of the input pattern are used to define the class to which an unknown pattern $\tilde{x}$ belongs presented in Equation 3 as:

$$cl = \arg\min_{k=1}^{K}(|AFR_k - fr|) \tag{3}$$

where fr denotes the firing rate produced by the neuron.

4.1. Classifying Firing Rate

SNN models cannot directly stimulate the input pattern $x \in \mathbb{R}^n$, rather, they stimulate an injected current I calculated from the input pattern. Since the synaptic weights are directly connected to the input pattern $x \in \mathbb{R}^n$, the computation of the I on the input pattern is presented as:

$$I = \mathrm{x} \cdot \mathrm{w} \cdot \theta \tag{4}$$

where $w \in \mathbb{R}^n$ denotes the set of synaptic weights and θ is a gain factor that assists the neuron to fire. This transformation could incite that one or more input patterns should be transformed into same or similar current, aiding the neurons to generate similar firing rates.

4.2. Adjusting `Synapses of Neuron Models

The fitness function to find the set of synaptic weights to maximize the accuracy of the models is defined as:

$$f(\vec{w}, D) = 1 - performance(\vec{w}, D) \quad (5)$$

where $\vec{w}$ defines the synapses of the model, D is the set of input patterns and $performance(\vec{w}, D)$ represents the function which computes the classification accuracy as the number of patterns correctly classified divided by the number of tested patterns.

5. Experimental Results

This section presents the analysis and discussion of results obtained. We present experiments on eight different benchmarked datasets taken from the UCI machine learning repository [26]. Table 1 shows the description of the datasets.

To validate the accuracy of the spiking neuron, ten (10) experiments over each dataset were performed. For every experiment two subsets were randomly generated from each dataset; 80% of the samples comprise the training subset, and the remaining 20%, the testing subset.

This section has three subsections. The first and second sections present comparison of results of the BA using LIF and Izhikevich models to other similar works respectively. The last subsection presents the results of applying the proposed methodology for solving a Breast Cancer, Hepatitis and Haberman classification problem. The parameters used in the experiments are shown in table 2 and table 4.

5.1. Analysis and Comparison of Results of the LIF Model

Fig.1 (a) and (b) show how the learning error evolves through each generation of the BA using the LIF model for iris plant and wine datasets. We noticed that the learning error rapidly decreases at the beginning of the evolutionary learning process and changes at a slower rate when a certain number of generations is reached. Nonetheless, the learning error attain with BA is good for the two datasets.

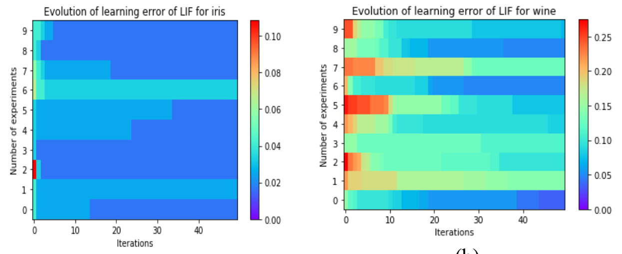
Fig.2 (a) and (b) shows the experimental results of iris plant and wine datasets for the spiking time when the neuron is stimulated using the LIF model. Each dot represents the time the neuron spikes. The synaptic weights found with the BA incite the LIF neuron to generate same firing rate when stimulated with patterns from same class and different firing rates to distinguish patterns from other classes.

Table 1 Description of Datasets

Dataset Name	No. of Classes	No. of Features
Iris Plant	3	4
Wine	3	13
Glass	6	9
Liver-Bupa	2	6
Diabetes	2	8
Breast Cancer	2	30
Hepatitis	2	19
Haberman	2	3

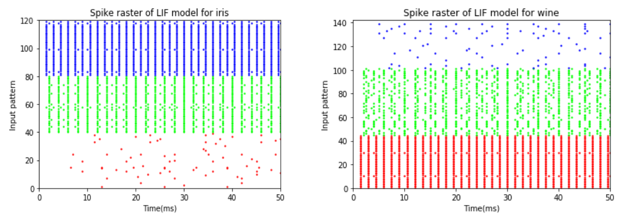
Table 2 Parameter values for the LIF model

Parameter	Value	Parameter	Value	Parameter	Value
a	0.5	V_r	-60	$N_{\_pop}$	30
b	-0.001	V_{peak}	35	$N_{\_gene}$	50
c	-50	θ	10	Q_{max}	5
dt	1	T	200	Q_{min}	1
e	0.9				



(a) (b)

Fig.1 Evolution of training error for LIF model. (a) Iris(b) Wine.



(a) (b)

Fig.2 Some experimental results obtained for using LIF. (a) Iris (b) Wine

In table 3, the average classification rate computed for iris plant and wine datasets for ten experiments are compared with the results in [18]. The result with LIF trained with BA shows a significant margin of improvement as compared to [18] when applied on non-linear problem.

Table 3 AVERAGE ACCURACY OF BA USING LIF

DATASETS	DE		BA	
	Tr. Clr.	*Te. Clr.*	*Tr. Clr.*	*Te. Clr.*
IRIS	0.9853	0.9560	0.9825	0.980
WINE	0.9528	0.8101	0.9408	0.9083

Tr. Clr. = Training Classification Rate, *Te. Clr.* = Testing Classification Rate

5.2. Analysis and Comparison of Results of the Izhikevich Model

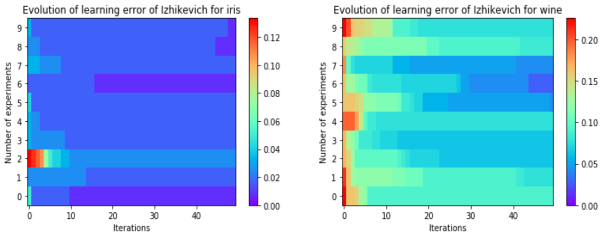
At the beginning of the evolutionary learning process we observed that independently of the initial value, the error converges speedily and we detected that the learning error changes at a slower rate at certain number of generation. This behavior was observed in all experiments. The error achieved with the BA was not good enough for the glass, liver and diabetes datasets because their error did not converge to an acceptable value. fig.3 (a) and (b) shows some of the error evolution results of the Izhikevich model.

Table 4 Parameter values for the Izhikevich model

Parameter	Value	Parameter	Value	Parameter	Value
C	100	a	0.03	N_{pop}	30
v_r	-60	b	-2	N_{gene}	50
v_t	-40	c	-65	Q_{max}	5
v_{peak}	35	d	100	Q_{min}	1
k	0.7	θ	200	T	200

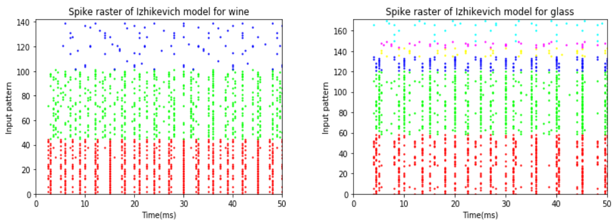
Fig.4 (a) and (b) present some results of the spiking time when the neuron is stimulated with patterns from wine and glass datasets. Each dot represents the time the neuron generates a spike. Unlike in [16] and [17] where the set of synaptic weights found during the training phase with the Izhikevich model was not good enough for the spiking neuron to produce similar firing rate when stimulated with patterns from the same class for glass, diabetes and liver datasets, our proposed method with the Izhikevich neuron generated firing rates different enough to differentiate among patterns from different classes and similar firing rates for patterns from same class for all the datasets. These patterns are ordered by class. The results achieved with the BA were highly acceptable for all the datasets.

Table 5 shows the average classification rate for the experimental results using the Izhikevich model for iris plant, wine, glass, liver and diabetes datasets. The results of BA were compared with the results in [15-17][19]. The results from the glass dataset was not so good because during the evolution learning, the error didn't converge to an acceptable value. However, the result obtained for the proposed model is acceptable since this is the first work to use BA to fine-tune the synaptic weight of SNN. From table 3 and table 5, we noticed that all the learning strategies performed differently since they were all inspired by different phenomena and they all generated slightly different results in this kind of problems at the end.



(a) (b)

Fig.3 Evolution of training error of Izhikevich for some datasets



(a) (b)

Fig.4 Experimental results obtained for Izhikevich. (a) Wine (b) Glass

5.3. Applicatio`n on Other Classification Problems using the LIF and Izhikevich Models

This subsection presents some results obtained with the proposed methodology in a problem related to Breast Cancer, Hepatitis and Haberman classification problems. The parameters for the LIF and the Izhikevich models, the BA, as well as the samples used for training and testing phases were set as same as in table 2 and table 4.

Fig.5 shows some of the experimental results for the models. Each dot represents the time the neuron fires a spike. Our proposed method demonstrated capabilities of producing firing rates different enough to distinguish among patterns from different classes and similar firing rates for patterns from same class for both models. Due to limited space, we present the error evolution for only breast cancer and hepatitis datasets for the two models (See fig.6). We noticed that independently of the initial value, the error

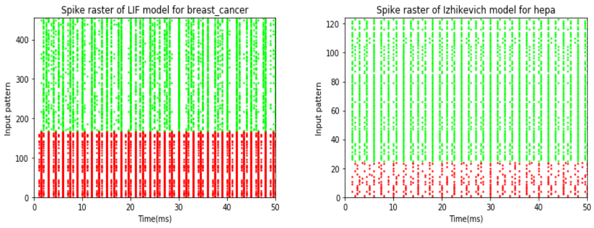


Fig.5 some experimental results for LIF and Izhikevich models

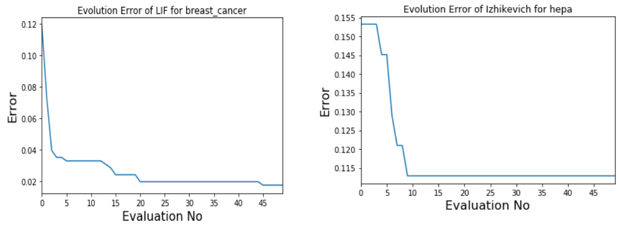


Fig.6 Evolution of training error for LIF and Izhikevich models for breast cancer and hepatitis datasets

Table 5 AVERAGE ACCURACY PROVIDED BY BA USING IZHIKEVICH MODEL

DATASETS	**DE**		**PSO**		**CUCKOO**		**ABC**		**BA**	
	Tr. Clr.	***Te. Clr.***	***Tr. Clr.***	***Te. Clr.***	***Tr. Clr.***	***Te. Clr.***	***Tr. Clr.***	***Te. Clr.***	***Tr. Clr.***	***Te. Clr.***
IRIS	0.9993	0.9833	0.9933	0.97	0.9942	0.9467	0.996	0.957	0.9858	0.9933
WINE	0.9796	0.8744	0.9782	0.8879	0.9831	0.9078	0.963	0.878	0.9479	0.9056
GLASS	0.8158	0.7411	0.8178	0.7457	0.8080	0.7646	0.832	0.703	0.6193	0.6674
DIABETES	0.8038	0.7371	0.7990	0.7619	0.8051	0.7477	0.800	0.743	0.7595	0.7253
LIVER	0.7620	0.6870	0.7591	0.6754	0.7609	0.6536	0.749	0.688	0.7072	0.713

Tr. Clr. = Training Classification Rate, *Te. Clr* = Testing Classification Rate

converges rapidly through the learning evolution process. As the error approached its stable state, the improvement rate increases greatly.

Once the spiking neuron was trained, we assess the accuracy using the testing subsets. For breast cancer datasets, we observed that the percentage of classification achieved with the BA using the training and testing subsets were >95% for both models (See fig.7). On the contrary, the percentage of classification attained for the training subset was >85% for hepatitis dataset and >77%, however, the testing subsets were mostly lower than that of the training subsets. This is because during the training the proposed method did not achieve an acceptable error rate. Nonetheless, the results achieved with the models trained with BA were highly acceptable.

6. Conclusion

In this work, we used BA as a learning strategy to train both LIF and Izhikevich models to solve non-linear pattern classification tasks. The experimental results obtained with the two models confirmed that BA can be used as an alternative to fine-tune the synaptic weight of a third generational neural model. Also, the BA learning strategy provides an acceptable result for performing discriminating task of generating same firing rate for patterns belonging to

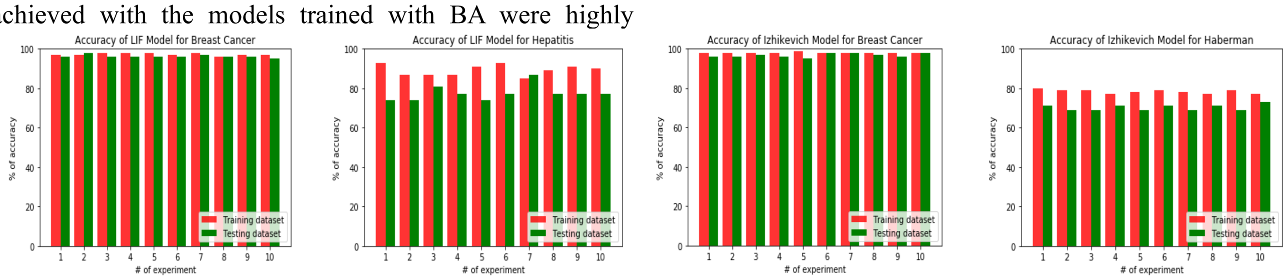


Fig.7 Accuracy of the proposed method during training and testing phase for breast cancer, hepatitis and haberman datasets

the same class in the output of the spiking neuron, and patterns belonging to different classes generate firing rates different enough to distinguish among the different classes. Moreover, the percentage of recognition obtained for the various datasets with the two models were acceptable.

Also, this research provides a clear idea of how powerful SNN is in pattern classification task. It should be remarked that this work used only one Izhikevich neuron model and one LIF neuron model for the classification tasks. The results suggest that only one spiking neuron is effective enough to perform different pattern recognition tasks and thus BA based SNN is an alternative way of solving pattern recognition tasks. If only one neuron is capable of solving pattern recognition problems, then perhaps several spiking neurons working together can improve the experimental results obtained in this research. And this will be explored in our future work by combining several spiking neurons to solving more complex pattern recognition tasks.

References

[1] R. A. V´azquez, "Pattern Recognition Using Spiking Neurons and Firing Rates," in *IBERAMIA*, 2010.

[2] M. Zhang, H. Qu, Y. C. Ammar Belatreche and Z. Yi, "A Highly Effective and Robust Membrane Potential-Driven Supervised Learning Method for Spiking Neurons," *IEEE Transactions On Neural Networks and Learning Systems,* vol.30,pp.123-13, 2019.

[3] A. G. Beatriz, S. Humberto and R. A. Vazquez, "Design of Artificial Neural Networks using a Modified Particle Swarm Optimization Algorithm," in *Proceedings of International Joint Conference on Neural Networks*, , USA, June 14-19, 2009.

[4] A. G. Beatriz, S. Humberto and R. A. Vázquez, "Design of Artificial Neural Networks Using Differential Evolution Algorithm.," in *ICONIP* , Berlin Heidelberg , 2010.

[5] K. Dervis and B. Bahriye, "Artificial Bee Colony (ABC) Optimization Algorithm for Solving Constrained Optimization Problems," in *IFSA*, Berlin Heidelberg , 2007.

[6] X.-S. Yang and S. Deb, "Cuckoo Search via L´evy Flights. Publications," in *Proc. of NaBIC 2009*, USA, December 2009.

[7] X.-S. Yang, "A New Metaheuristic Bat-Inspired Algorithm, In Nature-inspired cooperative strategies for optimization," in NICSO 2010, 2010.

[8] S. Mishra, K. Shaw and D. Mishra, "A New Metaheuristic Bat Inspired Classification Approach for Microarray Data.," in Procedia Technology, 2012.

[9] X.-S. Yang, "Bat Algorithm for Multi-objective Optimisation," International Journal of Bio-Inspired Computation, vol. 3, no. 5, pp. 267–274, 2011.

[10] H. G. Amir, Y. Xin-She, A. H. Alavi and T. Siamak, "Bat Algorithm for Constrained Optimization Tasks," in Neural Comput & Applic (2013), London, 2013.

[11] G.-Q. Huang, W. Zhao, Q.-Q. & Lu, "Bat Algorithm with Global Convergence for Solving Large-Scale Optimization Problem.," Application Research of Computers,vol. 30, no.3, pp.1-10, 2013.

[12] E. Elahe, A. Arash, G. Shaghayegh, A. Majid and S. Mehrdad, "Evolving Spiking Neural Networks of Artificial Creatures Using Genetic Algorithm," in IJCNN, 2016.

[13] H. N. A. Hamed, N. Kasabov, Z. Michlovský and S. M. Shamsuddin, "String Pattern Recognition Using Evolving Spiking Neural Networks and Quantum Inspired Particle Swarm Optimization," Springer-Verlag, Berlin Heidelberg, 2009.

[14] S. Hong, L. Ning, L. Xiaoping and W. Qian, "A Cooperative Method for Supervised Learning in Spiking Neural Networks," in Proceedings of the 2010 14th International Conference on Computer Supported Cooperative Work in Design, 2010.

[15] R. A. V´azquez and B. A. Garro, "Training Spiking Neurons by Means of Particle Swarm Optimization," in ICSI 2011, Berlin.

[16] R. A. Vázquez and A. G. Beatriz, "Training Spiking Neural Models Using Artificial Bee Colony," in Computational Intelligence and Neuroscience, February 2015.

[17] R. A. Vazquez, "Training Spiking Neural Models using Cuckoo Search Algorithm," in CEC 2011, 2011.

[18] R. A. Vazquez and A. Cachón, "Integrate and Fire Neurons and their Application in Pattern Recognition," in CCE 2010, México.

[19] A. V. Roberto, "Izhikevich Neuron Model and its Application in Pattern Recognition," Australian Journal of Intelligent Information Processing Systems, vol. 11, no. 1, February 2016.

[20] S. Mirjalili, S. M. Mirjalili and X.-S. Yang, "Binary Bat Algorithm," in Neural Computing and Applications, 2014.

[21] X.-S. Yang, "Bat Algorithm for Multi-Objective Optimization," International Journal of Bio-Inspired Computation, vol. 3, no. 5, pp267–274, 2011.

[22] X.-S. Yang, "Bat Algorithm and Cuckoo Search: A Tutorial," in Artif. Intell., Evol. Comput. and Metaheuristics, 2013.

[23] M. Zhang, H. Qu, A. Belatreche and X. Xie, "EMPD: An Efficient Membrane Potential Driven Supervised Learning Algorithm for Spiking Neurons," IEEE Transactions on Cognitive and Developmental Systems,

vol. 10, no. 2, pp. 151-162, June 2018.

[24] M. Zhang, J. Wu, Y. Chua, X. Luo, Z. Pan, D. Liu and H. Li, "MPD-AL: An Efficient Membrane Potential Driven Aggregate-Label Learning Algorithm for Spiking Neurons," AAAI-19, 2019, pp. 1327-1334.

[25] E. M. Izhikevich, "Which Model to Use for Cortical Spiking Neurons," IEEE Transactions on Neural Networks, vol. 15, no. 5, pp. 1063-1070, September 2004.

[26] UCI, "https://archive.ics.uci.edu/ml/datasets," UCI Repository of Machine Learning Databases (USA), Tech. Rep.,Department Inf. Comput. Sci. Univ. California, Irvine, 2014. [Online]