

Spiking ink drop spread clustering algorithm and its memristor crossbar conceptual hardware design

Iman Esmaili Paen Afrakoti¹, Vahdat Nazerian¹, Tole Sutikno^{2,3}

¹Department of Electrical Engineering, Faculty of Engineering and Technology, University of Mazandaran, Babolsar, Iran

²Master Program of Electrical Engineering, Faculty of Industrial Technology, Universitas Ahmad Dahlan, Yogyakarta, Indonesia

³Embedded System and Power Electronics Research Group, Yogyakarta, Indonesia

Article Info

Article history:

Received Nov 21, 2020

Revised Aug 20, 2023

Accepted Sep 19, 2023

Keywords:

Active learning method

Ink drop spread

Memristor

Neuro-fuzzy clustering

Spiking neural network

ABSTRACT

In this study, a new clustering algorithm that combines neural networks and fuzzy logic properties is proposed based on spiking neural network and ink drop spread (IDS) concepts. The proposed structure is a single-layer artificial neural network with leaky integrate and fire (LIF) neurons. The structure implements the IDS algorithm as a fuzzy concept. Each training data will result in firing the corresponding input neuron and its neighboring neurons. A synchronous time coding algorithm is used to manage input and output neurons firing time. For an input data, one or several output neurons of the network will fire; confidence degree of the network to outputs is defined as the relative delay of the firing times with respect to the synchronous pulse. A memristor crossbar-based hardware is introduced for implementation of the proposed algorithm as a processing hardware. The simulation result corroborates that the proposed algorithm can be used as a neuro-fuzzy clustering and vector quantization algorithm.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Vahdat Nazerian

Department of Electrical Engineering, Faculty of Engineering and Technology, University of Mazandaran

Pasdaran Street, Babolsar, Iran

Email: v.nazerian@umz.ac.ir

1. INTRODUCTION

Soft computing systems were originally developed to simulate human inference, and now these techniques find extensive application in diverse scientific domains, including but not limited to system control, modeling, decision-making, and knowledge extraction. Drawing inspiration from the biological processes of the brain, spiking neural networks (SNNs) are commonly regarded as the 3rd generation of artificial neural networks. The appeal of SNNs lies in their ability to precisely model the synaptic connections between neurons, which involves considering the timing of spike firing. Currently, the primary objective is to develop effective learning mechanisms that can leverage the unique characteristics of SNNs while preserving the fundamental properties of conventional connectionist models. Conventional connectionist models are general-purpose and easy-to-use models, further there were available simulators for them [1].

Besides SNNs, fuzzy systems have achieved great results. One of the most effective algorithms in fuzzy context is active learning method (ALM), which was proposed by Shouraki *et al.* [2]. Inspired by human problem-solving behavior, the ALM aims to approximate a multiple input-single output (MISO) system using several single input-single output (SISO) subsystems, each exhibiting output behavior with respect to a single input. This kind of behavior is depicted on a plane called ink drop spread (IDS) plane or IDS unit. The IDS method applies a two-dimensional membership function to each data point, diffusing information across the problem space, and combining the diffused data to obtain a narrow path on each IDS

plane. In contrast to instance-based learning approaches, the IDS algorithm does not retain data during the learning process [3], but there is a need for a short constant memory to store estimated narrow path representation data.

Creation of memristor in Hewlett Packard (HP) labs sparked interest in its utilization to implement various applications. In the domain of SNNs, memristor crossbars have been used to develop fully asynchronous spike-timing-dependent plasticity (STDP) learning algorithm of neuromorphic architectures [4]–[6]. Moreover, some others have drawn an analogy between memristor crossbar and IDS planes to implement fuzzy relations and algorithms [7]–[10]. The proposed networks have been depicted in Figures 1(a) and 1(b). The similarity of these architectures and their behavior, which are implementing brain computation procedures, was an encouragement to propose the current model. This model is a mixture of these two depicted architectures. In our neuro-fuzzy architecture, pulses do the computations, while input-output variables are fuzzy.

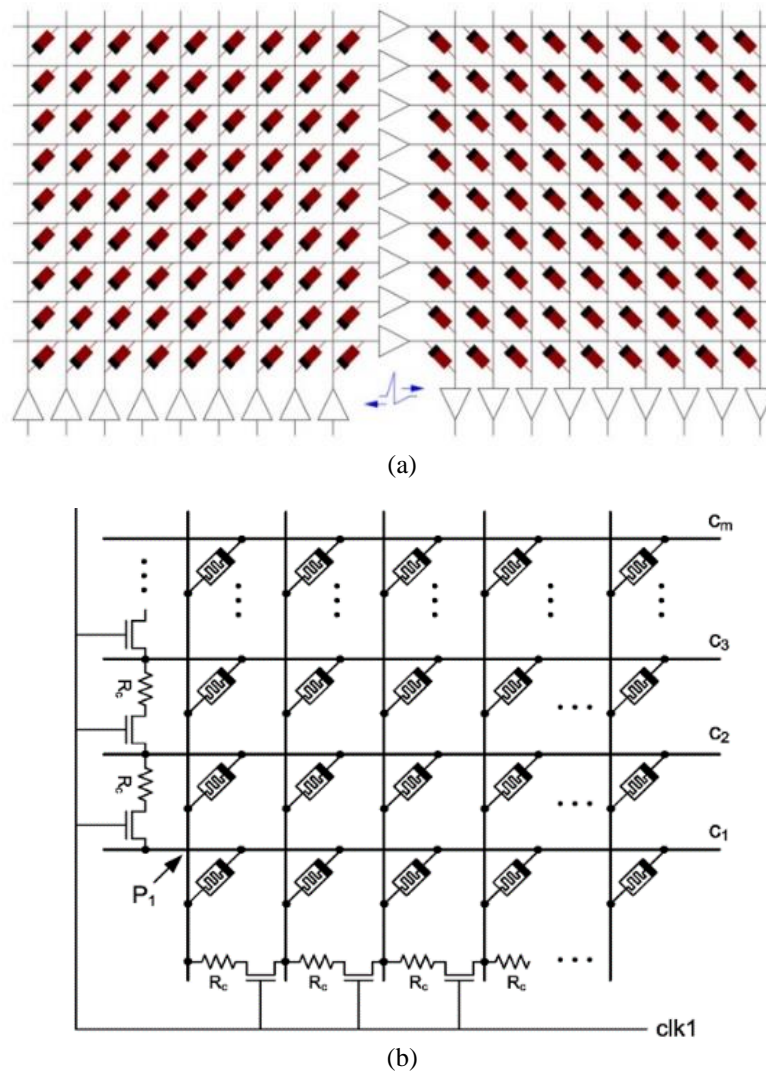


Figure 1. Proposed structure for (a) implementation of asynchronous STDP [4] and (b) IDS plane implementation [7]

The subsequent sections of the paper are structured in the following manner. Section 2 provides a comprehensive examination of the SNN models and the ink drop spread (IDS) algorithm. The introduction of the memristor-crossbar structure is discussed in section 3. The network structure under consideration is elaborated upon in section 4, while section 5 presents and thoroughly discusses the simulation results. Ultimately, section 6 presents the derived conclusions.

2. OVERVIEW OF SNNs

In addition to neuronal and synaptic state, SNNs integrate the concept of time into their model. Numerous models of spiking neurons have been proposed such as Hodgkin-Huxley [11], integrate and fire, leaky integrate and fire [12], theta neuron [13] and Izhikevich's neuron [14]. A comprehensive review on spiking neuron models was done in [1].

Leaky integrate and fire (LIF) is derived from Hodgkin-Huxley neuron model. The most important simplification in the LIF neuron is neglecting the shape of the action potentials. Thus, every spike is considered to be a uniform event, and defined only by its appearance time. The electrical model of LIF neuron consists of a capacitor C in parallel with a resistor R that is driven by an input current $I(t)$, as shown in Figure 2. In this model, the dynamics of the membrane potential in the LIF neuron are described by a single first-order linear differential as (1):

$$\tau_m \frac{du}{dt} = u_{rest} - u(t) + RI(t) \quad (1)$$

where $\tau_m = RC$ is the time constant of the voltage leakage of the neuron membrane model. Additionally, the firing time $t^{(f)}$ of the neuron is defined by a threshold crossing equation $u(t^{(f)}) = v$, under the condition $u'(t^{(f)}) > 0$. Immediately after $t^{(f)}$, the potential is reset to a given value of u_{rest} , $u_{rest} = 0$ is a common assumption. An absolute refractory period can be modeled by forcing the neuron's voltage to a value $u = u_{abs}$ during the time interval Δ_{abs} after a spike emission, and then restarting the integration by the initial value of u_{abs} .

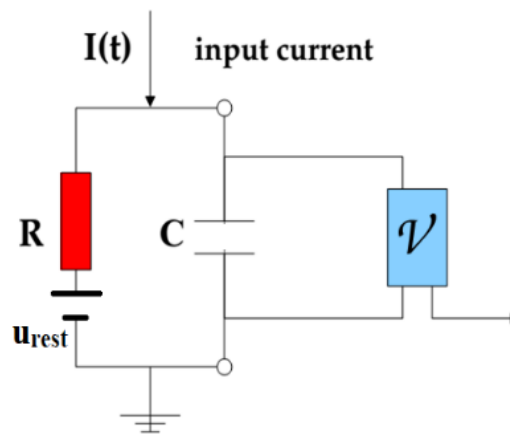


Figure 2. The integrate-and-fire neuron model (I&F) [1]

ALM is inspired by human problem-solving behavior and aims to approximate a MISO system using several SISO subsystems. Each subsystem represents the output behavior with respect to a single input, which is obtained from the pattern generated on a two-dimensional plane using the IDS operator. The IDS operator applies a two-dimensional membership function to each data point, diffusing information across the problem space and combining the diffused data to obtain a smooth pattern representing the functional behavior of a single variable. ALM has demonstrated its effectiveness in various engineering applications, such as modeling, control, and pattern recognition [15]–[18].

The pattern on a plane is constructed by projecting all $(x_i; y)$ data on the plane. Diffusion of information has the main role in this process; common sense implies that each sample data not only has information in its exact point but also is valid in neighboring points with less confidence degree. By getting away from the point of sample data, the confidence degree will decrease. It is similar to the notion of fuzzy membership functions. The diffusion of information can thus be implemented by considering a three-dimensional pyramid-shape fuzzy membership function centering at each data point as shown in Figure 3. Figure 3(a) depicts two such membership functions overlapped and aggregated. An IDS plane after applying IDS operator to the five data samples is shown in Figure 3(b). For the sake of simulation and hardware conformity, the range of input variables should be quantized to n levels, giving rise to $n \times n$ pixel IDS planes.

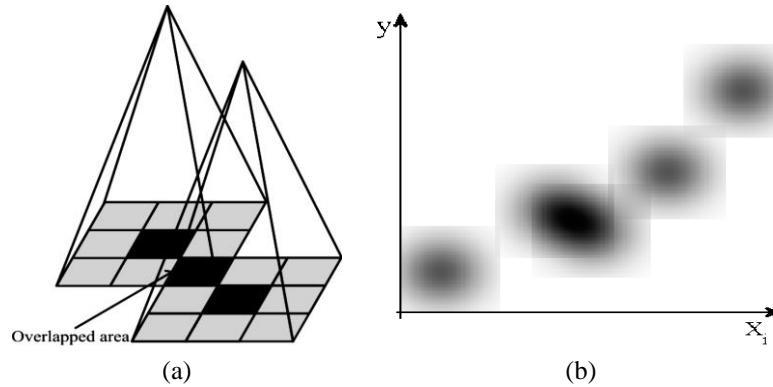


Figure 3. The diffusion of information is implemented by fuzzy membership function (a) ink stains with pyramid shape and (b) five Inks are diffused on the plane

3. MEMRISTOR AND MEMRISTOR CROSSBAR

A memristor is an analog variable resistor where its resistance can be adjusted by changing the direction and the duration of the applied voltage. Therefore, memristor can be used as a storage device in which analog values can be stored in it as memristance instead of voltage or charge [19]. In May 1, 2008, a paper, which was published in Nature by Hewlett Packard (HP) Labs [20], announced the first experimental realization of the memristor. This paper has caused an increased interest in this passive circuit element. It has become clear that memristor can have many potential applications such as non-volatile memory construction [21], creation of analog neural network and emulation of human learning [21]–[23], building programmable analog circuits [24], [25], implementing digital circuits [26] and signal processing algorithms [27], [28].

A memristor crossbar is a two-dimensional grid in which analog values can be stored in the crosspoints using the memristance of the memristors, as illustrated in Figure 1. Therefore, the memristor crossbar appears to be an ideal structure for implementing the two-dimensional planes of the Ink Drop Spread (IDS) units. In this case, the vertical and horizontal wires of the crossbar represent different discrete values of the variables x_i and y , respectively, while the memristors in the crosspoints act as the pixels of the IDS plane. Additionally, memristor crossbars have been demonstrated to be useful as synapses for implementing STDP learning rule [29]. STDP is an improvement of Hebb's rule that considers the precise relative timing of individual pre- and post- synaptic spikes, rather than their average rates over time. The shape of STDP function ξ can be interpolated from experimental data, such as that obtained by Bi and Poo [30], as shown in Figure 4(a). The resulting function $\Delta w(\Delta t)$, computed using the memristor model through (2), is illustrated in Figure 4(b). This result shows that a memristive type of mechanism could be behind the biological STDP phenomenon.

$$\Delta w(\Delta t) = \int f(V_{MR}(t, \Delta t)) dt = \xi(\Delta t) \quad (2)$$

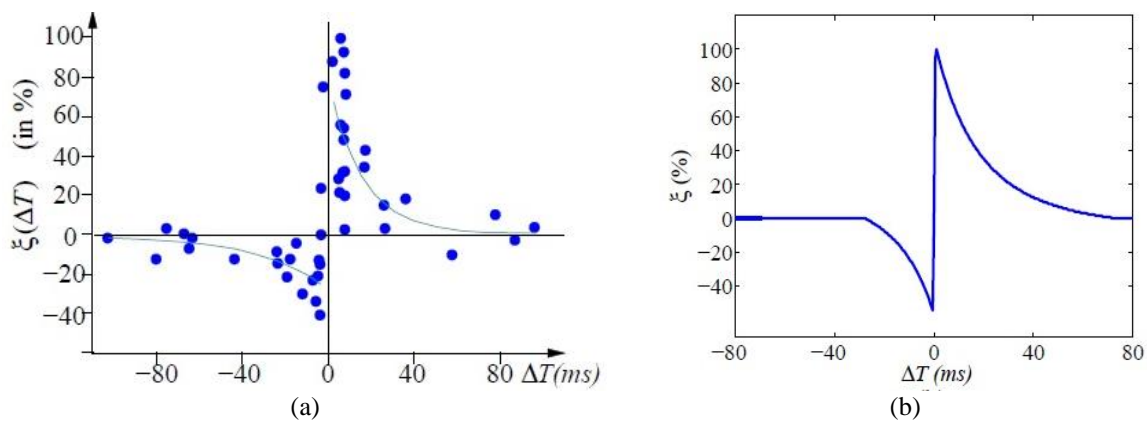


Figure 4. STDP (a) experimentally measured STDP function $\xi(\Delta t)$ on biological synapses (data from Bi and Poo) and (b) ideal STDP update function used in computational models of STDP learning [31]

4. PROPOSED NETWORK STRUCTURE

The proposed structure is a one layer, fully connected network, which uses the LIF type neurons. Each variable, input or output is partitioned into quantized levels. One neuron is used for each quantized level. Therefore, if the system has p inputs and one output, and each one is quantized into n levels, $p \times n$ input neurons and n output neurons would be implemented as depicted in Figure 5.

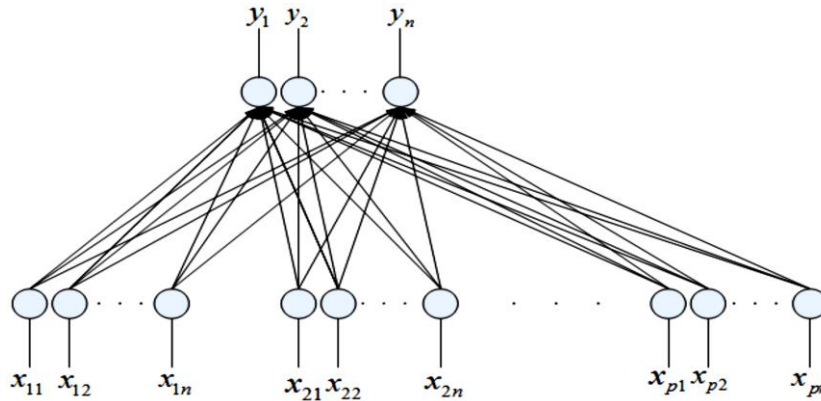


Figure 5. Proposed network scheme with n quantized level for each variable

Analogy between the above structure and IDS planes can be drawn. In each IDS plane there are n pixels for each input. It will be shown in the proposed inference algorithm that each pixel represents the confidence degree of IDS plane to its corresponding input-output value. This scheme is equivalent to the above proposed network. Each pixel in IDS plane represents a synaptic coefficient. As this value increases, the confidence degree of this plane to the desired output increases. Similarly, as the network synaptic coefficient increases, the corresponding output neuron is more likely to fire.

4.1. Information coding

As mentioned in the last section, the incoming data make the input neurons fire, corresponding to their quantized level. For each input both corresponding and neighboring neurons will fire, although neurons that are farther from the main neuron will have a lower confidence degree. It has been assumed that the main neuron is fired with confidence degree value equal to one.

In Figure 6, information coding has been depicted. In the first interval, a pulse in t_0 is fired, and in the second time interval, a pulse in t_1 is fired. As it can be seen in this figure, confidence degree for each pulse is $\mu(t_0)$ and $\mu(t_1)$ respectively. Note that every descending function can be used as confidence degree function.

Confidence degree of neighboring neurons can be determined using Bohte’s algorithm [32]. Figure 7 illustrates the procedure. On each input’s quantized level, a receptive field such as trigonometric, Gaussian or any convex function can be used. For an input quantized level, a vertical line is drawn. Every neuron that crosses the vertical bar has to fire related to the confidence degree which can be computed using the horizontal bar. Therefore, as the convex functions become wider, more neighbor neurons fire in the learning phase. Hence, ink drops radius has a direct relationship with wideness of convex function.

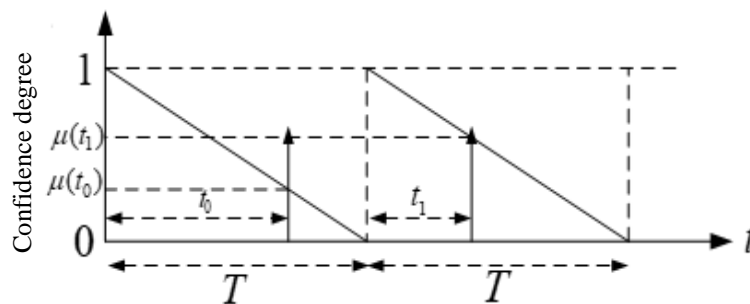


Figure 6. Confidence degree of each pulse relative to its starting time

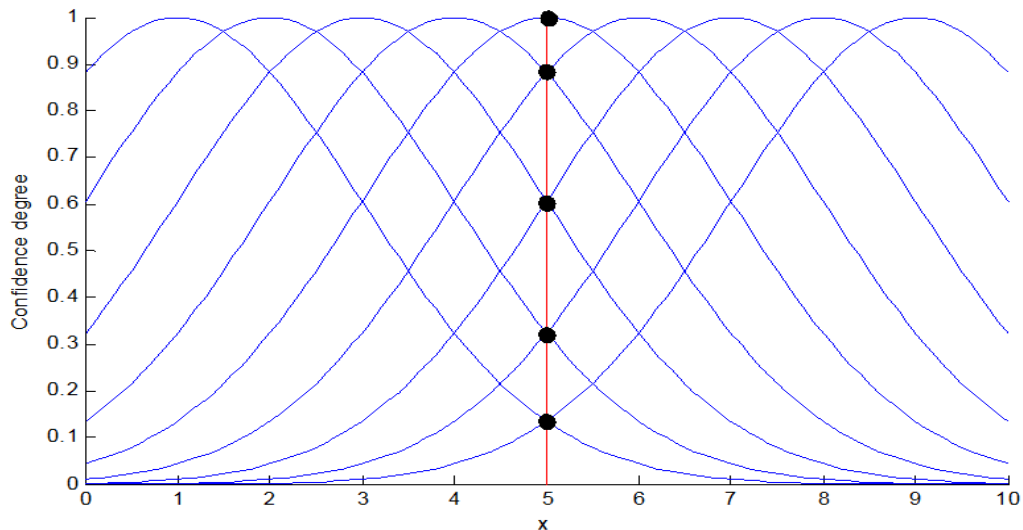


Figure 7. Confidence degree calculation using Bohte's algorithm for input value of $x=5$

Network learning uses winner takes all (WTA) mechanism, which needs a control block for implementation. The threshold voltage of output neurons is an issue which has to be considered. If this value is too small, only the neurons with very high confidence degrees will have a role in the firing of output neurons. It is also possible that output neuron fire just after the main neuron and before neighboring neurons. In this case, neighboring neurons have no role in choosing the winner. Albeit, this problem does not pose an algorithmic problem and the solution will be achieved, but ink drop spread loses its concept. The ink drop spread learning algorithm presumes that by occurring data in input space not only do we have information about that pixel but also, we have information of its surroundings. Hence, learning algorithms use both information codes: the information of the data itself and the information of its surroundings. On the other hand, high threshold value probably causes no winner neuron or requires a long time until a neuron fires. It can cause late convergence of algorithm, which is a weakness of implementing memristor crossbar networks.

In the clustering phase, the network receives data, which causes output neurons to fire. As the firing time of an output neuron gets closer to beginning of the period, the confidence degree of network to that neuron becomes higher. Therefore, the time between output neuron firings and beginning of the pulse is relative to confidence of network to each output. Thus, the input data can be in different clusters with different confidence degrees.

4.2. Hardware implementation of the proposed structure using memristor crossbar

The overall structure of network hardware for a system with two inputs and one output is shown in Figure 8. Four neurons are dedicated for each input and output variable, which means each variable is quantized to four levels. The neurons in this structure are implemented using the circuit proposed by Bernabe, which is depicted in the Figure 9 [31].

Considering consistency of network learning with learning algorithm introduced in the last section, pulse shape and firing type have to be specified. Two pulse types have been chosen for network hardware learning. The first type is the one used by input neurons which is applied to network, and then its inverse returns by the output winner neuron. Voltage amplitude of this pulse is lower than memristor's threshold voltage v_{th} and cannot cause any change in memristance. Furthermore, voltage difference between signal and returned pulse is more than threshold voltage in which memristance increases in proportion to it. The winner output neuron generates the second type pulse which is passed to input neurons. This pulse decreases synaptic coefficients of input neurons which have not fired. Its amplitude is more than memristors threshold voltage. As this amplitude increases, the memristance decreases. In addition, it is in opposite polarity compared to returned pulse.

As mentioned before, implementing learning algorithm has two phases. In the first phase, the main input neuron and its neighboring fire pulses and other input neurons reach a high impedance state. Subsequently, output neurons compete, and the winner neuron returns a pulse to its input. When the potential difference of pulses is applied to the memristor, its memristance decreases; consequently, the synaptic coefficient increases. In the second phase of training, the firing neurons transition to high impedance state

and the rest of neurons are connected to the ground potential. Therefore, the output winner neuron fires the second pulse. This pulse has opposite polarity to the first pulse; consequently, it decreases the synaptic coefficient.

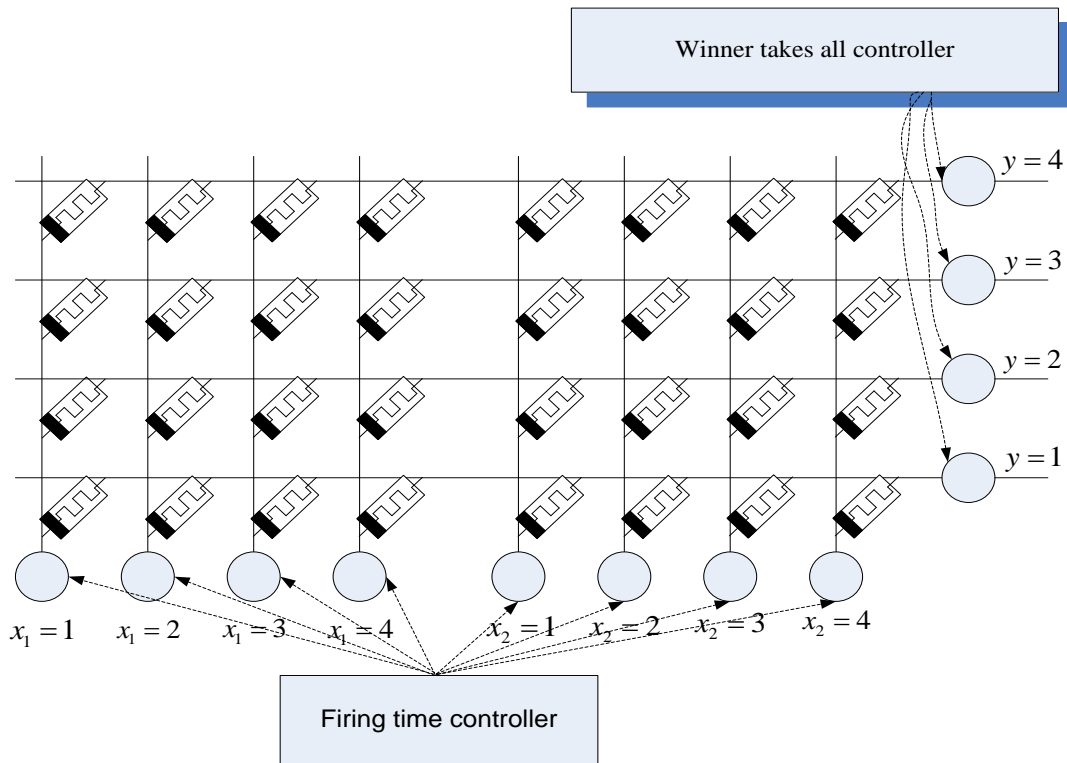


Figure 8. High level hardware structure of network with two input and four quantized level

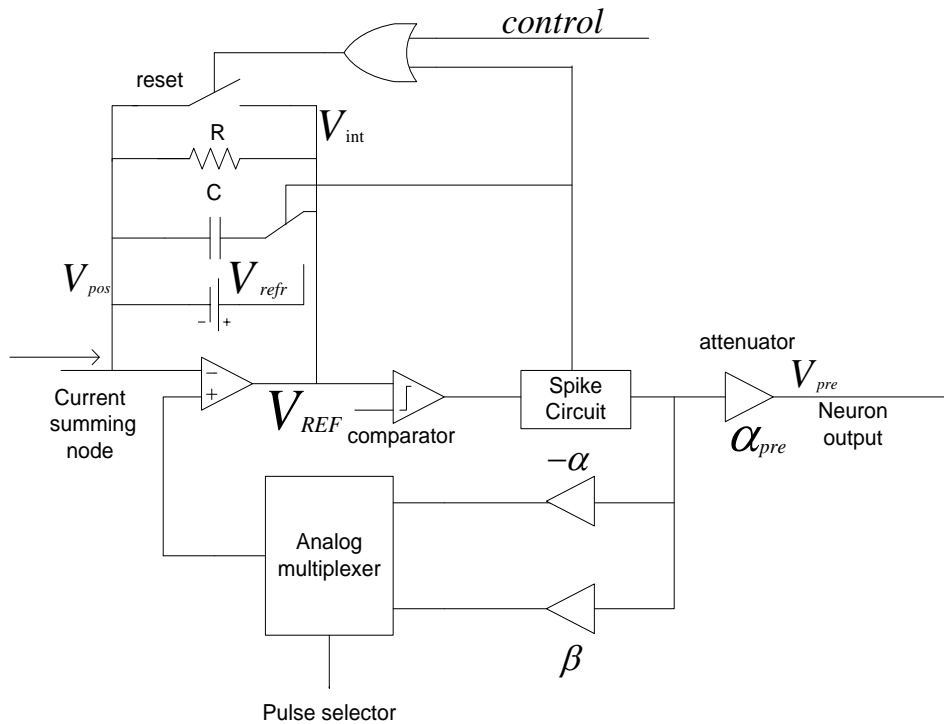


Figure 9. Hardware realization of LIF neuron [31]

In Figure 10, T is Learning time. In $T/2$ the process for increasing synaptic coefficient takes place. The confidence degree for firing each input neuron sets the starting time of firing in the interval of 0 to $T/4$. Then the firing pulse remains active till $T/2$. This coding has been chosen in order to maximize the possible number of neurons to influence firing of the output winner neuron. In this period, other input neurons reach high impedance. In the time interval $[T/2, T]$, the winner neuron fires another pulse with reverse polarity compared with the first pulse. In this state, input firing neurons in the last period go to high impedance, and other neurons connect to ground. Hence, this pulse increases synaptic coefficient in these neurons. The problem that must be considered here is that the increase in synaptic coefficients of the grounded neurons is greater than firing neurons.

In the Figure 10, v_1 is the fired pulse by an input neuron, v_2 is the fired pulse by an output winner neuron whose difference ($v_1 - v_2$) increases the value of synaptic coefficient. $v_{decrease}$ is the winner neuron pulse in the time interval $[T/2, T]$ which is used to decrease the synaptic coefficient. In Figure 11 an increase in synaptic coefficient of input neurons, which have been fired according to their firing time and firing time of winner neuron, has been depicted.

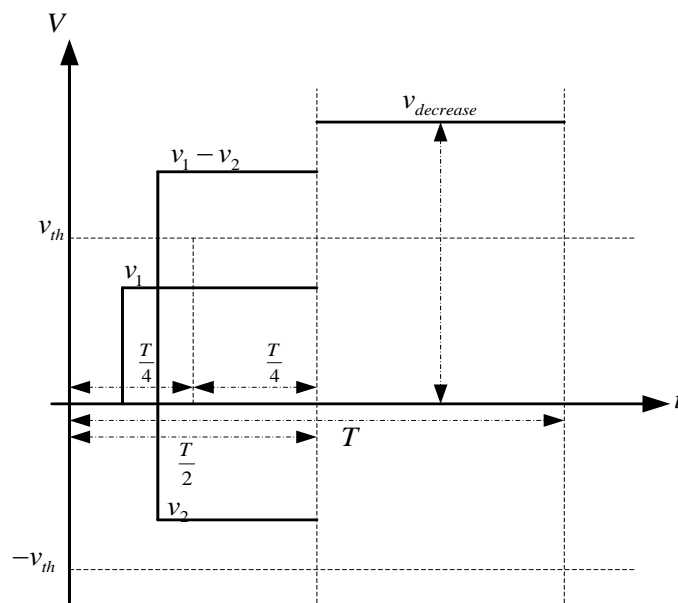


Figure 10. Learning pulses which are used in the process

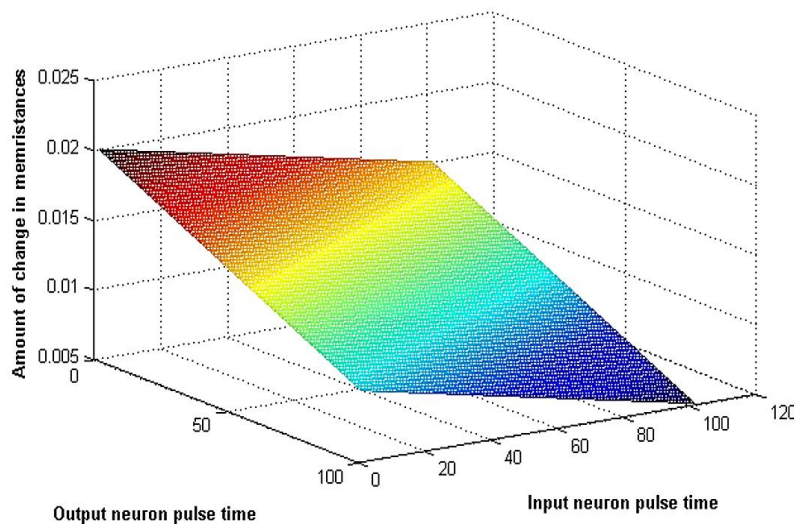


Figure 11. Changing of the synaptic weights versus firing time of input and output neuron

5. SIMULATION RESULTS

For simulation and evaluation of the proposed algorithm, three classes are randomly generated with two inputs, with a uniform distribution, means $(-1.5, -1.5)$, $(0, 0)$, and $(1.5, 1.5)$, and variance 1, as depicted in Figure 12. Figure 12(a) shows the training data; Figure 12(b) shows the clustering with a radius of 40; and Figure 12(c) shows the clustering with a radius of 20. These classes are fully separated. In this simulation, the resolution of input variables is 128, and 8 rows have been used for a maximum of 8 classes. Clustering has been simulated with radii of 20 and 40. As seen in Figure 12, when the radius is 40, test data are clustered in three separate classes, but when the radius of the ink drop is decreased, the number of clusters increases.

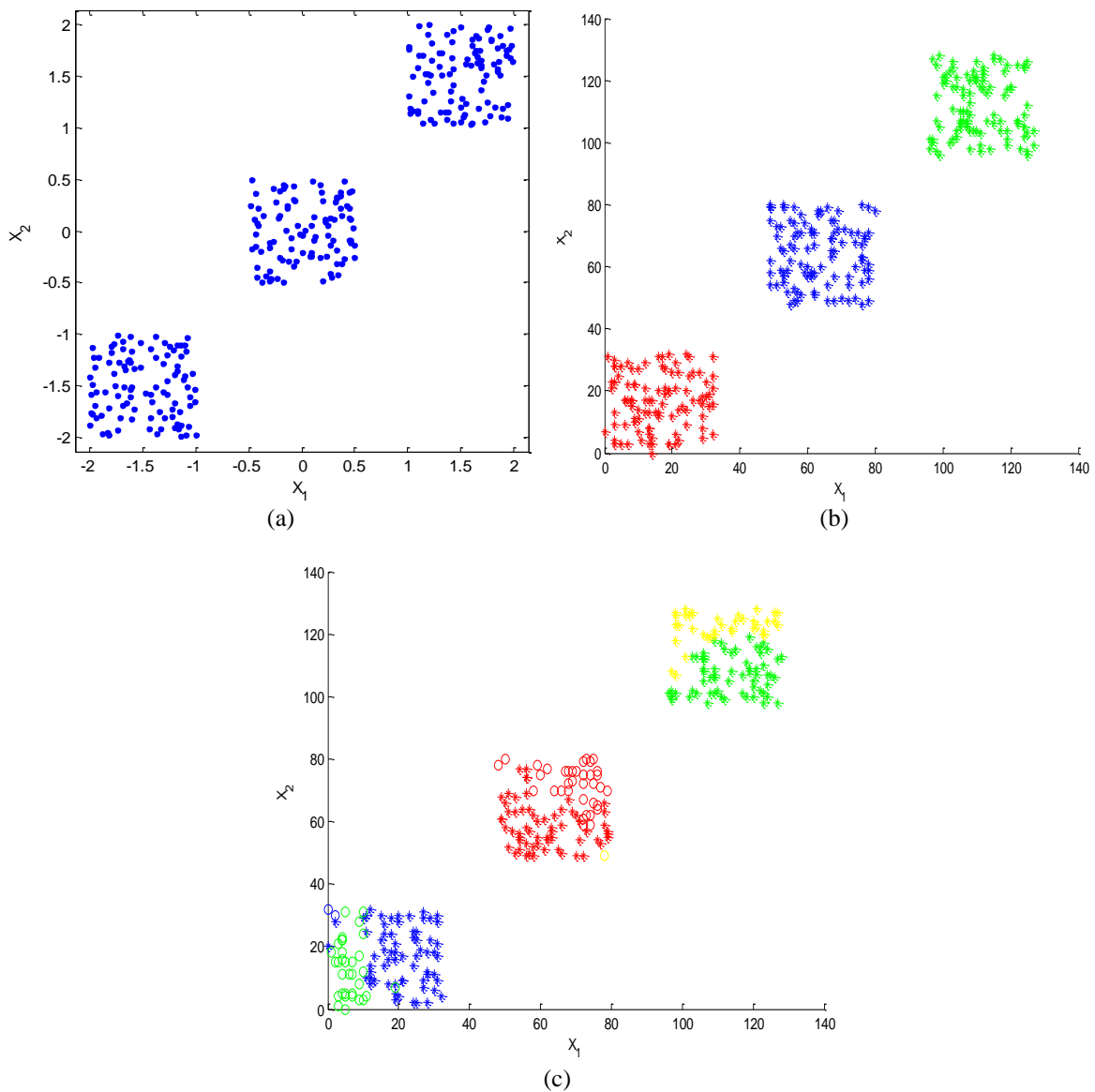


Figure 12. Simulation and evaluation of the proposed algorithm (a) training data, (b) clustering with radius=40, and (c) clustering with radius=20

For further evaluation, class means and variances are changed. So, classes become very close to each other. As shown in Figure 13, a similar behavior to the previous example has happened here. Figure 13(a) shows the training data; Figure 13(b) shows the clustering with a radius of 40; Figure 13(c) shows the clustering with a radius of 20, and Figure 13(d) shows the clustering with a radius of 10. Decrease in the radius of ink drop has increased the number of clusters. This characteristic of the proposed algorithm can be used for vector quantization. The smaller the radius is, the more the detected vectors are and vice versa.

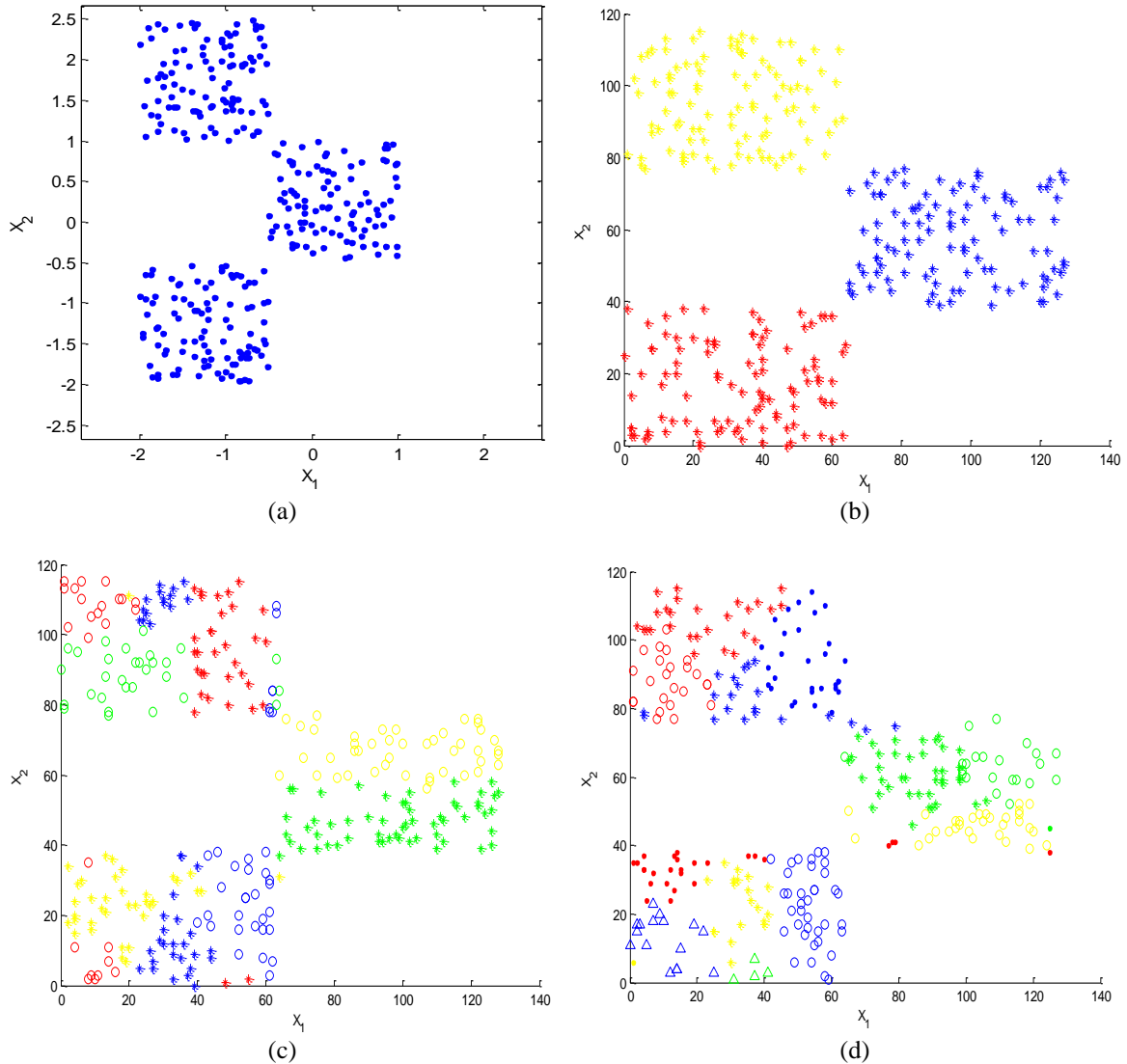


Figure 13. Other simulation and evaluation of the proposed algorithm (a) training data, (b) clustering with radius=40, (c) clustering with radius=20, and (d) clustering with radius=10

6. CONCLUSION

Combining the fuzzy concept and spiking neural network algorithm as a third artificial neural network generation is what this paper is about. Therefore, the border between these two branches of the artificial intelligence and soft-computing family is faded out in this algorithm. Although the proposed algorithm is not fully mature yet, the simulation result shows that it can be useful in clustering tasks. The proposed memristor-crossbar hardware structure of the algorithm can be a proof that this algorithm can be a new guideline for brain simulation domain.




REFERENCES

- [1] H. Paugam-Moisy and S. Bohte, "Computing with spiking neuron networks," in *Handbook of Natural Computing*, vol. 1–4, Springer Berlin Heidelberg, 2012, pp. 335–376.
- [2] S. B. Shouraki and N. Honda, "Recursive fuzzy modeling based on fuzzy interpolation," *Journal of Advanced Computational Intelligence and Intelligent Informatics*, vol. 3, no. 2, pp. 114–125, Apr. 1999, doi: 10.20965/jaciii.1999.p0114.
- [3] S. B. Shouraki and N. Honda, "Outlines of a soft computer for brain simulation," in *International Conference on Soft Computing Information/Intelligence Systems (IZUKA'98)*, Iizuka, Japan 1998.
- [4] G. S. Snider, "Spike-timing-dependent learning in memristive nanodevices," in *2008 IEEE International Symposium on Nanoscale Architectures*, Jun. 2008, pp. 85–92, doi: 10.1109/NANOARCH.2008.4585796.
- [5] X. Zhu, Y. Sun, H. Liu, Q. Li, and H. Xu, "Simulation of the spiking neural network based on practical memristor," *MATEC Web of Conferences*, vol. 173, Jun. 2018, doi: 10.1051/mateconf/201817301025.




- [6] J.-X. Shen, D.-S. Shang, Y.-S. Chai, S.-G. Wang, B.-G. Shen, and Y. Sun, "Mimicking synaptic plasticity and neural network using memtransistors," *Advanced Materials*, vol. 30, no. 12, Mar. 2018, doi: 10.1002/adma.201706717.
- [7] F. Merrikh-Bayat, S. B. Shouraki, and A. Rohani, "Memristor crossbar-based hardware implementation of the IDS method," *IEEE Transactions on Fuzzy Systems*, vol. 19, no. 6, pp. 1083–1096, Dec. 2011, doi: 10.1109/TFUZZ.2011.2160024.
- [8] I. E. P. Afrakoti, S. B. Shouraki, and B. Haghghat, "An optimal hardware implementation for active learning method based on memristor crossbar structures," *IEEE Systems Journal*, vol. 8, no. 4, pp. 1190–1199, Dec. 2014, doi: 10.1109/JSYST.2013.2295963.
- [9] I. E. P. Afrakoti, S. B. Shouraki, F. M. Bayat, and M. Gholami, "Using a memristor crossbar structure to implement a novel adaptive real-time fuzzy modeling algorithm," *Fuzzy Sets and Systems*, vol. 307, pp. 115–128, Jan. 2017, doi: 10.1016/j.fss.2016.02.016.
- [10] S. H. Klidbary, S. B. Shouraki, and I. E. P. Afrakoti, "An adaptive efficient memristive ink drop spread (IDS) computing system," *Neural Computing and Applications*, vol. 31, no. 11, pp. 7733–7754, Jul. 2019, doi: 10.1007/s00521-018-3604-0.
- [11] A. L. Hodgkin and A. F. Huxley, "A quantitative description of membrane current and its application to conduction and excitation in nerve," *The Journal of Physiology*, vol. 117, no. 4, pp. 500–544, Aug. 1952, doi: 10.1113/jphysiol.1952.sp004764.
- [12] R. B. Stein, "A theoretical analysis of neuronal variability," *Biophysical Journal*, vol. 5, no. 2, pp. 173–194, Mar. 1965, doi: 10.1016/S0006-3495(65)86709-1.
- [13] G. B. Ermentrout and N. Kopell, "Parabolic bursting in an excitable system coupled with a slow oscillation," *SIAM Journal on Applied Mathematics*, vol. 46, no. 2, pp. 233–253, Apr. 1986, doi: 10.1137/0146017.
- [14] E. M. Izhikevich, "Simple model of spiking neurons," *IEEE Transactions on Neural Networks*, vol. 14, no. 6, pp. 1569–1572, Nov. 2003, doi: 10.1109/TNN.2003.820440.
- [15] M. Murakami and N. Honda, "A study on the modeling ability of the IDS method: a soft computing technique using pattern-based information processing," *International Journal of Approximate Reasoning*, vol. 45, no. 3, pp. 470–487, Aug. 2007, doi: 10.1016/j.ijar.2006.06.022.
- [16] S. A. Shahdi and S. B. Shouraki, "Supervised active learning method as an intelligent linguistic controller and its hardware implementation," in *2nd IASTED International Conference on Artificial Intelligence and Applications (AIA '02)*, 2002, pp. 453–458.
- [17] M. Firouzi, S. B. Shouraki, and I. E. P. Afrakoti, "Pattern analysis by active learning method classifier," *Journal of Intelligent and Fuzzy Systems*, vol. 26, no. 1, pp. 49–62, 2014, doi: 10.3233/IFS-120714.
- [18] H. Sagha, I. E. P. Afrakoti, and S. Bagherishouraki, "Actor-critic-based ink drop spread as an intelligent controller," *Turkish Journal of Electrical Engineering and Computer Sciences*, vol. 21, no. 4, pp. 1015–1034, 2013, doi: 10.3906/elk-1106-40.
- [19] D. B. Strukov, G. S. Snider, D. R. Stewart, and R. S. Williams, "The missing memristor found," *Nature*, vol. 453, no. 7191, pp. 80–83, May 2008, doi: 10.1038/nature06932.
- [20] L. O. Chua, "Memristor-the missing circuit element," *IEEE Transactions on Circuit Theory*, vol. 18, no. 5, pp. 507–519, 1971, doi: 10.1109/TCT.1971.1083337.
- [21] R. Waser and M. Aono, "Nanoionics-based resistive switching memories," *Nature Materials*, vol. 6, no. 11, pp. 833–840, Nov. 2007, doi: 10.1038/nmat2023.
- [22] Y. V. Pershin, S. La Fontaine, and M. Di Ventra, "Memristive model of amoeba learning," *Physical Review E*, vol. 80, no. 2, Jul. 2009, doi: 10.1103/PhysRevE.82.019904.
- [23] S. Shin, K. Kim, and S.-M. Kang, "Memristor-based fine resolution programmable resistance and its applications," in *2009 International Conference on Communications, Circuits and Systems*, Jul. 2009, pp. 948–951, doi: 10.1109/ICCCAS.2009.5250376.
- [24] Y. V. Pershin and M. Di Ventra, "Practical approach to programmable analog circuits with memristors," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 57, no. 8, pp. 1857–1864, Aug. 2010, doi: 10.1109/TCSI.2009.2038539.
- [25] P. Kuekes, "Material implication: digital logic with memristors," in *Memristor and memristive systems symposium*, 2008, vol. 21.
- [26] B. Mouttet, "Proposal for memristors in signal processing," in *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering*, vol. 3, Springer Berlin Heidelberg, 2009, pp. 11–13.
- [27] F. Merrikh-Bayat and S. Bagheri-Shouraki, "Mixed analog-digital crossbar-based hardware implementation of sign-sign LMS adaptive filter," *Analog Integrated Circuits and Signal Processing*, vol. 66, no. 1, pp. 41–48, Aug. 2011, doi: 10.1007/s10470-010-9523-3.
- [28] F. Merrikh-Bayat and S. B. Shouraki, "Memristive neuro-fuzzy system," *IEEE Transactions on Cybernetics*, vol. 43, no. 1, pp. 269–285, Feb. 2013, doi: 10.1109/TSMCB.2012.2205676.
- [29] T. Serrano-Gotarredona, T. Masquelier, T. Prodromakis, G. Indiveri, and B. Linares-Barranco, "STDP and STDP variations with memristors for spiking neuromorphic learning systems," *Frontiers in Neuroscience*, vol. 7, 2013, doi: 10.3389/fnins.2013.00002.
- [30] G. Bi and M. Poo, "Synaptic modification by correlated activity: Hebb's postulate revisited," *Annual Review of Neuroscience*, vol. 24, no. 1, pp. 139–166, Mar. 2001, doi: 10.1146/annurev.neuro.24.1.139.
- [31] C. Zamarreño-Ramos, L. A. Camuñas-Mesa, J. A. Pérez-Carrasco, T. Masquelier, T. Serrano-Gotarredona, and B. Linares-Barranco, "On spike-timing-dependent-plasticity, memristive devices, and building a self-learning visual cortex," *Frontiers in Neuroscience*, vol. 5, 2011, doi: 10.3389/fnins.2011.00026.
- [32] S. M. Bohte, J. N. Kok, and H. La Poutre, "Error-backpropagation in temporally encoded networks of spiking neurons," *Neurocomputing*, vol. 48, no. 1–4, pp. 17–37, Oct. 2002, doi: 10.1016/S0925-2312(01)00658-0.

BIOGRAPHIES OF AUTHORS






Iman Esmaili Paeen Afrakoti    received his B.Sc. in Electrical Engineering in 2005 from Khaje Nasir Toosi University of Technology, Tehran, Iran, M.Sc. and Ph.D. degrees in Digital Electronic from Department of Electrical Engineering, Sharif University of Technology, Iran in 2007 and 2014, respectively. He joined soon to Electrical Engineering Department of University of Mazandaran as a faculty member. His research interests include memristor, neuromorphic systems implementation based on memristor crossbar structures, Chaos, fuzzy systems, artificial neural networks, soft-computing and pattern recognition. He can be contacted at email: i.esmaili.p@umz.ac.ir.



Vahdat Nazerian    received his B.Sc. degree from Iran University of Science and Technology, Tehran, Iran in 2003, M.Sc. and Ph.D. degrees from Khaje Nasir Toosi University of Technology, Tehran, Iran in 2005 and 2011, respectively, all in Electronics. He joined the Electrical Engineering Department of the University of Mazandaran in 2014 as an assistant professor. His research and teaching concern semiconductor devices, analysis and fabrication. His research presently focuses on soft-computing, intelligent network, nanotechnology and application, renewable energy, and optimization algorithms. He can be contacted at email: v.nazerian@umz.ac.ir.



Tole Sutikno    is a lecturer, and serves as the head of the Master Program of Electrical Engineering, Faculty of Industrial Technology, Universitas Ahmad Dahlan, Indonesia. He received his B.Eng., M.Eng. and Ph.D. degrees in Electrical Engineering from Universitas Diponegoro, Universitas Gadjah Mada and Universiti Teknologi Malaysia, in 1999, 2004 and 2016, respectively. He has been an Associate Professor in UAD, Yogyakarta, Indonesia since 2008. He is among the top 2% of researchers named by Stanford University and Elsevier BV as the most influential scientists in the world for 2021–present. He is currently an Editor-in-Chief of the TELKOMNIKA and the Head of the Embedded Systems and Power Electronics Research Group (ESPERG). His research interests include the field of digital design, industrial applications, industrial electronics, industrial informatics, power electronics, motor drives, renewable energy, FPGA applications, embedded system, artificial intelligence, intelligent control, information technology and digital library. He can be contacted at email: tole@te.uad.ac.id.