ISSN: 0067-2904

The Effects of Conductance on Metastable Switches in Memristive Devices Based on Anti-Hebbian and Hebbian (AHaH) Learning Rules

Amenah D. Abbood[1*], Ammar A. Hasan[2], Baraa A. Attea[1]
[1]Department of Computer Science, University of Baghdad, Baghdad, Iraq
[2] Department of Computer Engineering, University of Baghdad, Baghdad, Iraq

Received: 21/12/2020 Accepted: 5/4/2021

Abstract

In the last few years, the literature conferred a great interest in studying the feasibility of using memristive devices for computing. Memristive devices are important in structure, dynamics, as well as functionalities of artificial neural networks (ANNs) because of their resemblance to biological learning in synapses and neurons regarding switching characteristics of their resistance. Memristive architecture consists of a number of metastable switches (MSSs). Although the literature covered a variety of memristive applications for general purpose computations, the effect of low or high conductance of each MSS was unclear. This paper focuses on finding a potential criterion to calculate the conductance of each MMS rather than the whole conductance as reported in the literature. Anti-Hebbian and Hebbian (AHaH) learning rules are used to mimic the changes in memristance of the memristors. This research will concentrate on the effect of conductance on an individual MSS to simulate the nanotechnology devices of the memristors. A single synapse is presented by a couple of memristors to mimic its resistance switching. The learning circuit of artificial synapses could be used in many applications, such as image processing and neural networks, for pattern classification of synapses, represented by a map of the memeristors. These synapses are essential elements for data processing and information storage in both real and artificial neural systems.

Keywords: Memristive, machine learning, Anti-Hebbian and Hebbian learning, nano-device and simulation, artificial synapses.

تأثير التوصيلية على المفاتيح الثابتة في Memristive المستندة على قواعد التعلم (Hebbian AHaH و Anti−Hebbian)

امنة داحم عبود[1] * , عمار عادل حسن[2] , براء علي عطية[1]
[1] قسم علوم الحاسبات، جامعة بغداد، بغداد، العراق
[2] قسم هندسة الحاسبات، جامعة بغداد، بغداد، العراق

الخلاصة

في السنوات القليلة الماضية ، قدمت البحوث فائدة كبيرة في دراسة استخدام الاجهزة الحديثة للحسابات. تعد أجهزة Memristive مهمة في البنية والديناميكيات والعمليات الكهربائية للشبكات العصبية الاصطناعية

*Email: amenah.abbood@sc.uobaghdad.edu.iq

(ANNs) بسبب تشابهها مع المشابك البيولوجية والخلايا العصبية فيما يتعلق بخصائص التبديل لمقاومتهم. يتكون البناء المعماري لـ Memristive من عدد من المفاتيح الثابتة (MSS). على الرغم من أن البحوث تغطي مجموعة متنوعة من تطبيقات لحسابات الأغراض العامة ، فإن تأثير الموصلية المنخفضة أو العالية غير واضح لكل (MSS). تم استخدام طريقة التعلم AHaH لتعديل مقاومة الجهاز. يركز هذا البحث على تأثير الموصلية لكل مفتاح MSS بدلاً من التوصيل الكامل كما تم استخدامه في البحوث الموجودة .يتم بناء مشبك واحد من قبل اثنين من memristors لتقليد تبديل المقاومة. هذا البحث يركز أن كل MSS تتأثر بقيمة مقدار التوصيل لمحاكاة الجهاز المتناهي في الصغر الذي ممكن ان يستخدم في كثير من التطبيقات مثلا معالجة صور والشبكات العصبية والمشابك يتم تمثيلها بالجهاز. تعتبر هذه المشابك عناصر أساسية لمعالجة البيانات وتخزين المعلومات في كل من الأنظمة العصبية الحقيقية والاصطناعية.

1. Introduction

Recently, memristor based applications have been developed rapidly through research in computation and human brain emulation, which is simulated as synapse in artificial neural networks. The artificial synapse needs to recall its previous history. In conjunction with pre-synaptic and post-synaptic neurons, memory-resistor characteristics, which is a main behaviour in the memristors, can do that in an efficient method, realizing all the synaptic properties needed. In reality, the term memristor is an abbreviation of "memory" and "resistor". It is an electrical device combining the behaviours of both resister and memory. The resistance of the memristor is not constant but depends on the number of electrical charges that flow through it, which can determine the amount of the current flowing in the memristor. The relation between voltage (v) across the memristor and current (i) flowing through it at any time (t) can be expressed by the following Ohm's law:

$$v(t) = M\big(q(t)\big)i(t) \tag{1}$$

where is the charge-dependent resistance or memristance (in Ohms) at time *t*. At any given time, the integral time of the current through the memristor determines the value of incremental memristance, as follows [1, 2]:

$$\int_{-\infty}^{t} i\,(t)d(t) \tag{2}$$

The relation between voltage and current is shown in Figure 1. The simulation of this circuit element is of much interest in the field of computation and synapse representation in real and artificial neural systems. For example, arithmetic operations can be implemented based on memristance rather than on the circuit based on voltage or current. This is important in hardware implementation for synapse simulation [3].

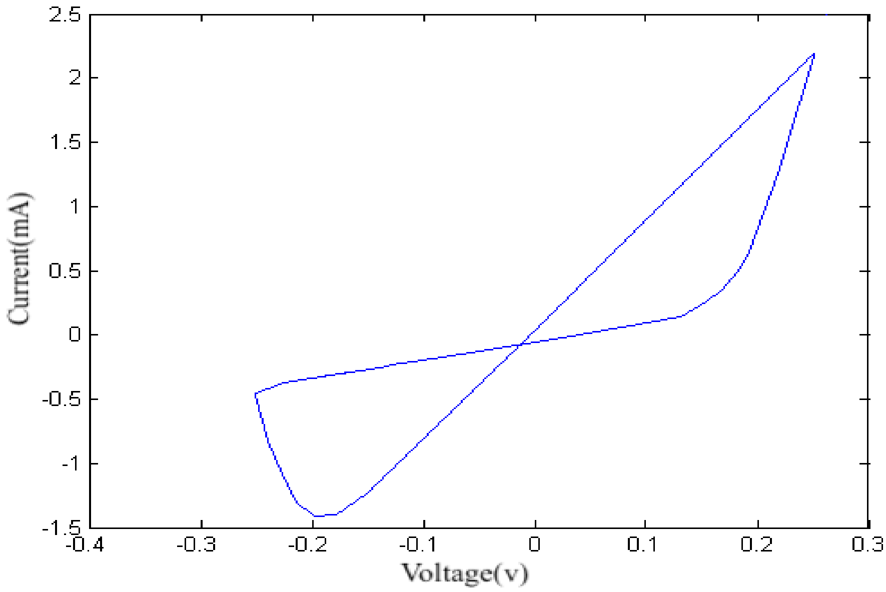

Figure 1-Sinusoidal input voltage applied through the memristor reflects the relation between the voltage and current.

The simulation of memristive nano-devices as synapses has an important role in both computation and information storage [4]. In this case, the memristor can provide the same features of a synapse depending on the learning rule. This paper shows the effect of conductance on each MSS to find the whole conductance of the memristive. AHaH learning rule could be used to change the memristance of the memristor, then enabling the synapse to update its weights depending upon the occurrence of spikes between the input and output.

The rest of the paper is structured as follows. In Section 2, we describe the required background and the relevant work. The generation of the memristive model and simulation are described in Sections 3 and 4. We introduce our results in Section 5. The conclusions are drawn in Section 6.

2. Related works

Several emerging technologies have been suggested to build hardware-based ANNs, such as memristive devices, phase change memories (PCMs) [5], and magnetic random access memories (MRAMs) [6]. Memristor-based artificial neurons have recently been created, but with restricted interaction within an interconnected network with artificial synapses. The resistance of memristor is not fixed, but changes depending on the amount and the direction of the charges flowing through it [1]. When the power goes out, the resistance will remain the same; therefore, it is non-volatile memory.

It is well known in electronics that there exist only three fundamental passive circuit elements: the resistor, capacitor, and inductor. However, more than 50 years ago, a fourth fundamental circuit element, the memristor, was discovered [2]. In the earliest concept for the memristor proposed in 1971, Chua predicted that the memristor would be a two-terminal electrical device that represents the missing circuit element [2]. This device provided a relationship between flux and charge, but this study was only theoretical [2, 7].

In 2008, the memristor was highlighted by Hewlett-Packard (HP) Labs when a group of scientists realized the first memristor electrical device, which was a nano-size device based on TiO_2 [8], following the theory which was predicated in 1971 by Chua. In the same year, HP combined the memristive with CMOS neurons to produce neuromorphic circuits [9-10]. In this study, they overlapped the Spike Time Dependent Plasticity learning law, which is an updated biological synaptic rule, with memristance. Since then, memristors have witnessed dramatic developments and research interest. In 2009, a nonlinear memristive grid, rather than nonlinear resistive grids, was used for edge smoothing in two-dimensional image processing. The result showed that the memristive grid was more efficient than the resistive grid in noise elimination and edge precision [11]. In addition, it was efficient in neuromorphic computing [12].

The next stage in research development was to simulate the biological synaptic process. Two, rather than one, memristors are required, connected serially with different polarities and can do arithmetic operations based on analogy circuit, rather than voltage based circuit [3-4]. In 2014, Nugent and Timothy [13] introduced a general memristive device model based on AHaH learning rules, where synaptic weights are represented using two memristors connected serially with the same polarities. These memristors are needed to simulate AHaH circuit nodes. There are many memristive device models which model specific devices, but this type of memristor modelled a large number of devices [13-14].

Modelling memristive devices is believed to be very helpful for general purpose computing and machine learning applications, such as image processing, neural networks, communication systems, and neuromorphic-in-memory applications [15-21]. In this research, we will concentrate on simulated models of memristive devices, rather than physical devices, in terms of the effect of conductance of MSS on the performance of the synaptic weights. This simulation improves the efficiency of learning process in terms of time and space.

3. Memristive model generation

Memristive neuromorphic device is a successful candidate for the development of artificial brain. The memristive model is needed for simulating AHaH node circuits [13]. The memristor consists of a number of MSSs (*N*), some are in the high conductance state (N_B) and the remaining are in the low-conductance state ($N_A = N - N_B$). All these parameters must be determined to find the model. In addition, the conductance for each MSS must be determined in both cases; in other words, determination of the conductance of the switch in state A and state B.

The conductance (G_m) of the memristor for all MSSs is computed from the following equation:

$$G_m = \left(N_B \times \left((G_B \div N) - (G_A \div N)\right)\right) + N \times G_A \tag{3}$$

where N_B is computed depending on *N* and N_A, $N_A = R_a \times N,$ where the initial value for the resistance is R_a, a random value between 0 and 1 represents the lowest and the highest resistance, respectively. Therefore, $N_B = (1 - R_a) \times N$.

4. Simulation of single memristor and synapse creation

In this study, memristor devices could be created from the materials of either chalcogenides or metal-oxides. These devices have a common architecture, with the following parameters: the sinusoidal voltage is used as the input voltage source with a frequency of 200 Hz and the number of MSSs is set to 2000000. In the first step, all these channels are in state *A* (low conductance).

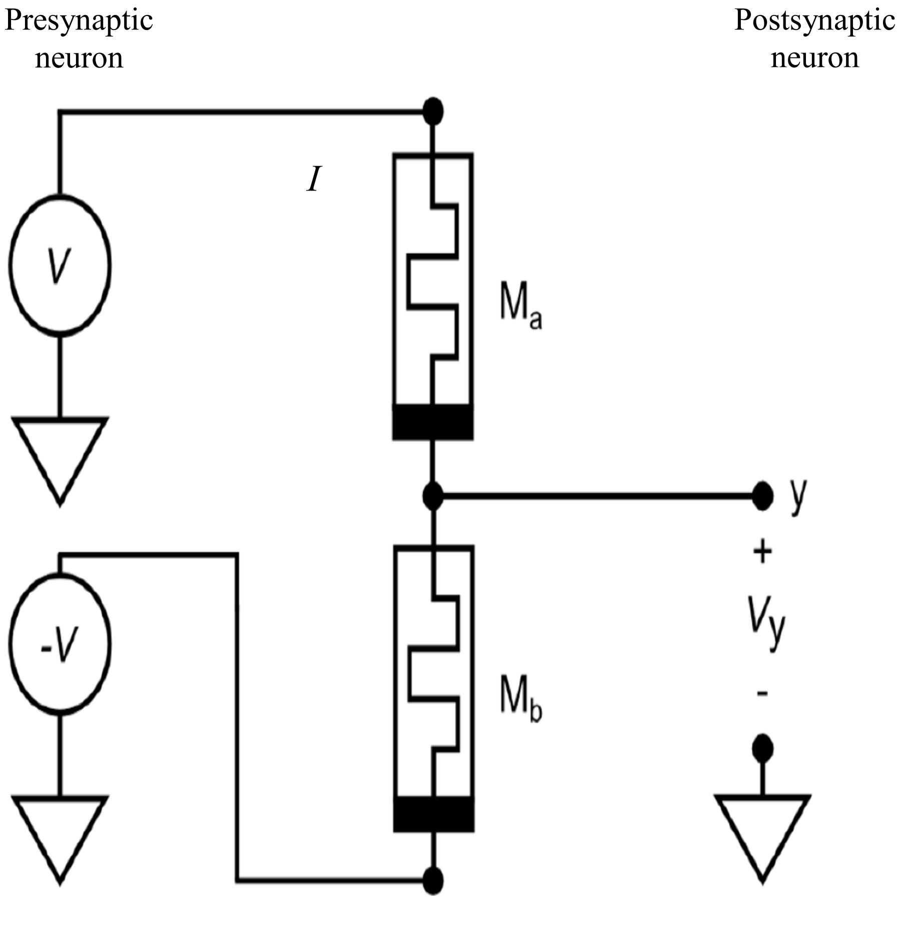

Figure 2- Two memristors to represent synapse. The voltage divider is used to compute output voltage at y between *v* and -*v* [13].

Clearly, to simulate biological synapses, two memristors are needed [3-4] which are connected serially with the same polarities. Thus, both memristors compete to dissipate pathways to achieve AHaH rule, as in Figure 2. A voltage divider produces output *y* between *v* and -*v.*

This model can be applied to different memristive devices of various materials, such as chalcogenides, metal-oxides, AIST and GST. Both memristors used sinusoidal voltage as the input and the same parameters used in the single memristors, except that the time scale is different in order to observe the dynamical states of each memristor separately. When the input voltages are applied, the memristance of both devices is increased or decreased to set the initial weights of the synapses. Then, the output of the voltage divider is computed from the following equation:

$$vb = I \div C_b \tag{4}$$

where vb is the voltage for M_b, I is the current of the memristor, and G_b is the conductance for the second memristor (M_b). The voltage for the M_a depends on both total voltage v and the voltage for the M_b.

$$v_a = v - v_b \tag{5}$$

where v and v_b are the total voltage and voltage for M_b, respectively.

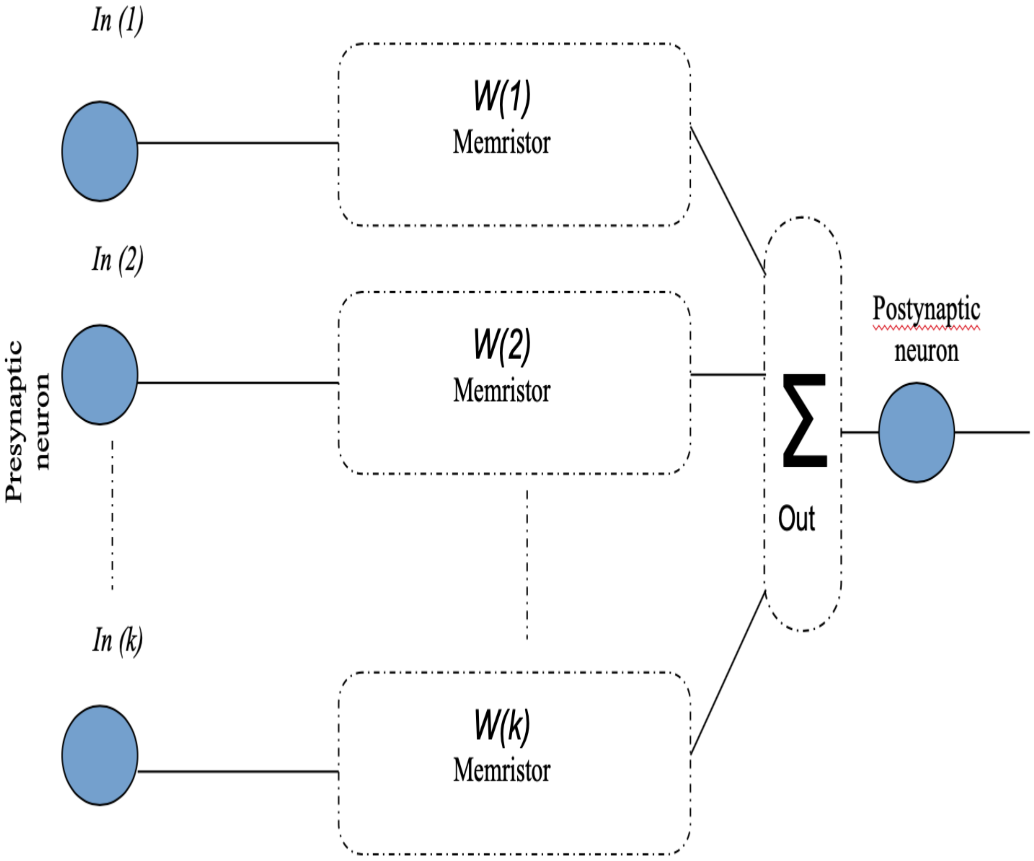

Figure 3- A pre-description of a perceptron network. The postsynaptic neuron is an output summation of the k synapses connected by k inputs.

$$Out = \begin{cases} 1 & \geqslant thr \\ 0 & < thr \end{cases} \quad (6)$$

Thus, the map weights of a particular perceptron network could be described by assigning each weight to a single memristor circuit, as shown in figure 3. A set of synapse weighting map (W1, W2, …. Wk) are arranged to transfer active signals from the inputs (1,2,….k) of the presynaptic neuron to the following (postsynaptic) neuron via the output. The value of the output is compared with a threshold, according to equation 6, to find whether its value is higher (or lower), and then determine the accumulated value for firing the postsynaptic neuron through sending an output spike.

5. Observational studies

In order to create AHaH node, an array of synapses is required, with the input being represented as spike and bias which refer to the driven voltage source to and analog output y, while driven voltage source F refers to feedback. This node implements AHaH plasticity with circuit simulation. The mechanism of the AHaH learning rule for circuit simulations is achieved by several steps. First, the material used for the memristor is determined, such as Ag-chalcogenide, AIST, GST, or WOx, because the update in conductance is different from one material to another. Then, the conductance is computed from equation (3) for both devices of the memristors (A and B) after the input data is converted into a spike. The weight for each synapse is computed from the following equation:

$$w = v_{dd} \times C_A + v_{ss} \times C_B \quad (7)$$

where C_A is the conductance for memristor A and C_B is the conductance for memristor B. The input voltages (v_{dd}= 0.5V and v_{ss}= - 0.5V) are set for the M_a and the M_b, respectively. Then, two processes are applied to update conductance, namely the read and write cycles. Through the read part, Kirchho's Current Law (KCL) is used to produce the output voltage across the y, where -v and +v are set to the input spike for M_a and M_b, respectively.

$$v_y = (v_{dd} \sum_{i=0}^{M-1} C_A + v_{ss} \sum_{i=0}^{M-1} C_B) \div (\sum_{i=0}^{M-1} C_A + \sum_{i=0}^{M-1} C_B) \quad (8)$$

where M is the number of active spikes input only with bias. Both M_a and M_b are competed to dissipate the energy pathway. The adaption in synaptic weight is implemented through the read and write cycles, depending on the Anti-Hebbian and Hebbian learning. In the read cycle, it is important to point out that with the access to the synapse, the output (v_y) is read, but at the same time, adaption is happening (merging memory and processing), that is, the conductance is increased through Anti-Hebbian learning in the read phase. The difference between the driven voltage source for M_a (v_{dd}) and the output voltage y is the amount of the voltage applied across the switch v to change its state. This represents the conductance adaption in M_a. If the v_y accedes the height of barrier potential (the threshold value of the switch), then the switch changes its state and the active weights update their amount via generating a spike for firing the output neuron.

In the same way, the conductance of the M_b is updated by the difference between y and v_{ss} to govern the neuronal firing events. In both the above cases, the conductivity of the active input memristor is increased, while the conductivity of the bias memristor is decreased by Anti-Hebbian learning. This process will update the conductance of the synapses that have firing events, while the synapses that have no firing events will decrease.

In the writing process, depending on the driven voltage source and the output voltage, either the conductance of M_a or the conductance of M_b will update. M_a is rewarded in any of the following cases:

1. If feedback (F) is supervised with value 1.
2. If y is positive and F is unsupervised with value 0.

In the above two cases, M_b is the loser; therefore, it will decay by updating its conductance for active input and bias. The voltage applied across the switch is $v_{dd} - v_{ss}$ through the conductance adaption for the active input and reverse polarities, while it is $(-(v_{dd} - v_{ss}))$ for the bias input, and zero for the non-active input. Thus, the conductance of the MB will decrease while the M_a will be rewarded by not decaying (The voltage applied across the switch is zero).

On the other hand, MB is rewarded in any of the following cases:

1. If feedback (F) is supervised with value -1.
2. If y is negative and F is unsupervised with value 0.

Therefore, MA is the loser and it will decay. By the same way, when M_b is the loser, MA conductance will be decreased.

Table1-Sample of the conductance adaption that depends on the value of the output voltage through the read and the write cycles for MA. The conductance is increased through the reading process, and if the output voltage is lower than zero, then the conductance is decreased through the writing process.

Initial conductance (1/Ω)	Output voltage (mV)	Read update (1/Ω)	Write update (1/Ω)
904200	8.338	907163	90764
972692	11.4604	973571	973562
983843	6.48602	984350	984347
963994	0.9859	965115	965117
980925	5.27301	981539	981532
912916	-8.24796	915569	886904
95713	-0.0994328	958072	928238
982073	3.65474	982586	982584

Table 2-A sample of the conductance adaption that depends on the value of the output voltage of both the read and the write cycles for MB. The conductance is increased through the reading process, and if the output voltage is higher than zero, then the conductance is decreased through the writing process.

Initial conductance (1/Ω)	Output voltage (mV)	Read update (1/Ω)	Write update (1/Ω)
965352	-11.6386	966536	966532
957156	2.35617	958535	958537
931360	4.34479	933675	902213
945491	0.700627	947280	947274
988854	-3.67227	989204	989193
942652	4.73639	944518	912805
982269	0.786579	982860	950091
916938	6.64123	919614	888867

As a result, this process is a counteraction to the reading process, because in the writing process the conductance is decreasing and in the reading process the conductance is increasing, as illustrated in Tables (1, 2). This counteraction is important to avoid the conductance reaching a saturated state.

The most important advantage of the function simulation is that it has less computation than the circuit simulation, while both converge in the simulation results.

V. Conclusions

In this work, we describe a memristive device for general purpose computing and machine learning through the implementation of neuromorphic learning rules based on memristive nano-devices. The memristor works with changing resistance depending on the current flowing through it. In addition, this work shows how synapse weights can be built from two memristors. An array of these synapses is used to build AHaH node, which could be used for machine learning applications with minimal computational overheads and cost. There are different applications of machine learning; for example, supervised and unsupervised classification and clustering based on a memristive nano-device, which has important characteristics in storage and computations.

References

[1] Tetzla R. **2014**. Memristors and memristive systems, Technische University Dresden, Dresden Germany.

[2] Chua LO. **1971**. Memristor: the missing circuit element, IEEE Transactions on Circuit Theory 18(5), 507-519.

[3] Merrikh-Bayat F, Shouraki S.B. **2011**. Memristor-based circuits for performingbasic arithmetic operations. Procedia Computer Science 3: 128-132.

[4] Merrikh-Bayat F, Shouraki S.B. **2010**. Bottleneck of using single memristor as a synapse and its solution. Preprintar URL arXiv:1008.3450v2.

[5] Tuma, T., Pantazi, A., Le Gallo, M., Sebastian, A., & Eleftheriou, E. **2016**. Stochastic phase-change neurons. Nature nanotechnology, 11(8), 693.

[6] Kang, M., Gonugondla, S., & Shanbhag, N. R. **2020**. Future Prospects. In Deep In-memory Architectures for Machine Learning (pp. 161-162), Springer, Cham.

[7] Chua LO, Kang SM. **1976**. Memristive devices and systems, Proceedings of the IEEE, 64, 209-223.

[8] Yang JJ, Pickett MD, Li X, Ohlberg DA, Stewart DR and Williams RS. **2008**. Memristive switching mechanism for metal/oxide/metal nanodevices, Nature Nanotechnology, 429-433.

[9] Snider GS. **2008**. Spike-timing-dependent learning in memristive nanodevices, In Proc. IEEE International Symposium on Nanoscale Architectures (NANOARCH), 85, 92.

[10] Snider GS. **2008**. Cortical computing with memristive nanodevices, Sci-DAC Rev., 10, 58-65.

[11] Jiang F, Shi B. E. **2009**. The memristive grid outperforms the resistive grid for edge preserving smoothing, European Conference on Circuit Theory and Design Conference Program (ECCTD), 181-184.

[12] Gelencser A, Prodromakis T, Toumazou C, Roska T. **2012**. Biomimetic Model of the Outer Plexiform Layer by Incorporating Memristive Devices, Phys. Rev. E- Statistical, 85(4), 041918,1-10.

[13] Nugent, M.A and Melter TW. **2014**. AHaH computing-from metastable switches to attractors to machine learning, PLOS ONE, 9(2), e85175.

[14] Nugent, M. A. and Molter. T. W. **2018**. Thermodynamic-RAM Technology Stack, Int. J. Parall. Emerg. Distrib. Syst.,33, 430–444.

[15] Kuzum, D., Yu, S., & Wong, H. P. **2013**. Synaptic electronics: materials, devices and applications. Nanotechnology, 24(38), 382001.

[16] Dastanova, N., Duisenbay, S., Krestinskaya, O., & James, A. P. **2018**. Bit-plane extracted moving-object detection using memristive crossbar-cam arrays for edge computing image devices, IEEE Access, 6, 18954-18966.

[17] Sahin, M. E., Taskiran, Z. G. C., Guler, H., & Hamamci, S. E. **2020**. Application and modeling of a novel 4D memristive chaotic system for communication systems, Circuits, Systems, and Signal Processing, 1-30.

[18] Pershin, Y. V., & Di Ventra, M. **2010**. Experimental demonstration of associative memory with memristive neural networks, Neural networks, 23(7), 881-886.

[19] Alatabi, H.A. and Abbas, A.R. **2020**. Sentiment analysis in social media using machine learning techniques. Iraqi Journal of Science, pp.193-201.

[20] AL-Rawi, K.R. and AL-Rawi, S.K. **2020**. Smart Doctor: Performance of Supervised ART-I Artificial Neural Network for Breast Cancer Diagnoses. Iraqi Journal of Science, pp.2385-2394.

[21] Gemo, E., Faneca, J., G.-C. Carrillo, S., Baldycheva, A., Pernice, W.H.P., Bhaskaran, H. and Wright, C.D. **2021**. A plasmonically enhanced route to faster and more energy-efficient phase-change integrated photonic memory and computing devices. Journal of Applied Physics, 129(11), p.110902.