INSTITUTE OF INFORMATION AND COMMUNICATION TECHNOLOGIES
BULGARIAN ACADEMY OF SCIENCES
CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume **26**, No 1
Sofia • 2026 Print ISSN: 1311-9702; Online ISSN: 1314-4081
DOI: 10.2478/cait-2026-0010

Spike Timing Dependent Plasticity Versus Intrinsic Plasticity as Feature Extraction Technique

Petia Koprinkova-Hristova

Institute of Information and Communication Technologies, Bulgarian Academy of Sciences, 1113 Sofia, Bulgaria
E-mail: petia.koprinkova@iict.bas.bg

Abstract**:** *This work investigates the effect of Spike Timing-Dependent Plasticity* (*STDP*) *of the synapses in the randomly connected Spiking Neural Networks* (*SNN*) *on the distribution of the firing rates of the individual neurons. It was observed that STDP, as a homeostatic plasticity rule, forces SNN activity to reflect the input structure. This effect is similar but not identical to the Intrinsic Plasticity* (*IP*) *tuning of Reservoir Computing* (*RC*) *recurrent neural networks. Both IP and STDP rules allow for capturing of the input data structure into the network state. This explains why STDP-trained SNNs are good for feature extraction from multidimensional data for classification purposes.*

Keywords**:** *Spiking neural network, Spike timing dependent plasticity, Clustering, Reservoir computing, Intrinsic plasticity.*

1. Introduction

Feature extraction methods aim transformation of the raw data into a more informative form for further analysis. There are a variety of approaches depending on the data type they work on. Nowadays huge amount of scientific literature deals with data-specific features extraction techniques. Two out of thousands of the informative reviews are [13, 22]. The methods can aim dimensionality reduction like PCA and LDA, statistical characteristics extraction like mean value and standard deviation, conversion of words to numerical vectors in language processing, extraction of frequency features from time series data, automatic learning of characteristics by deep neural networks, etc. Nevertheless, there is no universal recipe for the full spectrum of data we need to process. Many machine learning methods actually try to mimic the way the human brain extracts useful information from incoming sensory information and processes it for decision-making in everyday life. Hence, investigation of how the brain performs data feature extraction attracts interest from both neurobiological and computational intelligence viewpoints.

Inspired by the biologically plausible models of neural cells [10] and knowledge of how our brain processes information, Spiking Neural Networks

(SNNs) have become widely used in artificial intelligence nowadays [19]. In contrast to Artificial Neural Networks (ANNs), SNNs use binary representation of information in the form of sequences of discrete events called spikes, which allows for faster calculations with lower power consumption [9], called neuromorphic computing. Another intriguing feature of the brain is the ability to learn from continuously incoming data streams in real time, in contrast to the conventional gradient-based offline training methods applied for ANNs. Among the family of bio-inspired learning techniques, the Spike-Timing-Dependent Plasticity (STDP) attracted significant attention [25] for practical applications in neuromorphic computing.

STDP is a homeostatic plasticity rule that is a form of long-term modification of the synapses (biological counterpart of the connection weights) dependent on the order of pre- and post-synaptic spiking activity [8]. There are multiple studies on how STDP works [2-6, 18-20, 23, 27]. In [2], it was shown that STDP learning in a random initial network stimulated by a random spike sequence leads to the emergence of a small-world connectivity, i.e., formation of clusters of closely connected neurons. A similar STDP effect is also reported in [3]. In [4], it was shown that STDP rewires a randomly connected SNN of neurons having different activity characteristics in an activity-dependent manner. Similarly, Li and Small [18] demonstrated the synchronization effect of synaptic plasticity on a heterogeneous SNN. Chaudhari et al. [5] demonstrated how STDP can be used for unsupervised clustering of time series. Debanne and Inglebert [6] discussed neurobiological findings about the contribution of STDP-like synaptic plasticity in the formation of memory in vivo. In [19, 20], it was shown that STDP learning can localize a repeating spatiotemporal spike pattern, i.e., perform temporal coding. In [23], the frequency-clustering ability of the STDP rule is demonstrated by splitting a neural population into a few groups synchronized at different frequencies. Tal, Peled and Siegelmann [27] investigated STDP-provoked clustering of neurons in a heterogeneous SNN. These characteristics of the STDP rule were exploited in various practical tasks for spatio-temporal data series classification [12, 16].

Inspired by the random connectivity in neocortical networks in the brain, a rather simplified recurrent ANN (RNNs) called Echo State Networks (ESN) were proposed by Jaeger [11]. Even though the original idea was to design fast online trainable RNNs working with continuous data, it appears that the states of their randomly connected pool of neurons can also be exploited for data feature extraction for classification purposes. The Intrinsic Plasticity (IP) tuning rule proposed in [26], aiming at the target distribution of reservoir activity, can be considered as a homeostatic plasticity rule for artificial RNNs analogous to the STDP for SNNs. It was demonstrated that an IP-tuned ESN reservoir is good for feature extraction of multidimensional data [14], gray-scale images [15], as well as from time series data [17].

Since both STDP and IP rules were applied for clustering and classification purposes, the present paper investigates the features extracted by a randomly connected SNN trained via STDP on a small test example, as was done for the

IP-tuned ESN in previous works [14, 15]. The experiment shows that even though the two training rules originate from different fields and have different aims, in fact, they perform the data feature extraction in a very similar way.

The rest of paper is organized as follows: Section 2 presents briefly the SNN basics and the STDP training rule; Section 3 – an experiment with artificially designed 3-Dimensional (3D) data set of three overlapping clusters is described and the analyses of features extracted from the static and STDP trained SNNs are presented; Section 4 reminds briefly the Echo State Network (ESN) basics and the IP tuning approach and results with respect data features extracted by IP tuned ESN reservoir from previous works; in Section 5 comparative discussion on both training approaches is presented and the paper finishes by the concluding remarks.

2. Basics of SNN and STDP training

Spike timing Neural Networks (SNN) consist of neurons using biologically inspired models of neural cells [10]. A simplest one is Leaky Integrate and Fire (LIF) [24]:

(1) $$\frac{dV_{\mathrm{m}}}{dt} = -\frac{V_{\mathrm{m}} - E_{\mathrm{L}}}{\tau_{\mathrm{m}}} + \frac{I_{\mathrm{syn}} + I_{\mathrm{e}}}{C_{\mathrm{m}}}.$$

The dynamics of each neural cell membrane potential V_{m} depends on the membrane resting potential E_{L}, time constant τ_{m}, membrane capacity C_{m}, and total input current that is the sum of the external constant input current I_{e} and the synaptic input current I_{syn} coming from the given neural cell's connections with the other neurons in the SNN. When the membrane potential reaches a given threshold V_{th}, the cell emits a binary signal called a spike and returns to its resting state V_{rest}.

Connections between neural cells, called synapses, can be static (with constant weights), like in ANNs, or plastic (with dynamically changing weights). Among a variety of plasticity rules, the Spike Timing Dependent one (STDP) is widely used. According to it, the change of a synaptic weight Δw of a synapse connecting two neurons is determined by the STDP rule [8] according to Equation (2). The parameter λ is learning rate. The temporal filter $K(\Delta t)$ (Equation (3)) depends on the difference Δt between pre-synaptic (t_{pre}) and post-synaptic (t_{post}) spike times. The updating functions f_+ and f_- from Equation (4) depend on the weight w as well as on two parameters, α and μ.

(2) $$\Delta w = \begin{cases} -\lambda f_-(w) \times K(\Delta t) & \text{if} \quad \Delta t \le 0, \\ \lambda f_+(w) \times K(\Delta t) & \text{if} \quad \Delta t > 0, \end{cases}$$

(3) $$K(\Delta t) = e^{-|\Delta t|/\tau}, \;\; \Delta t = t_{\mathrm{post}} - t_{\mathrm{pre}},$$

(4) $$f_+(w) = (1 - w)^{\mu}, \;\; f_-(w) = \alpha w^{\mu}.$$

This rule is considered a form of homeostatic plasticity since the synaptic weight changes in dependence on the reactions of both neural cells connected by a given synapse. Thus, the overall SNN dynamics are governed by the mutual influence of its neurons as well as the external inputs influencing their behavior.

3. Simulation experiment

For the experimental purpose, a randomly connected SNN composed of Leaky Integrate and Fire (LIF) neurons was simulated using NEST Simulator [7]. Neurons in the SNN are randomly connected via excitatory (positive) and inhibitory (negative) synapses, as shown in Fig. 1.

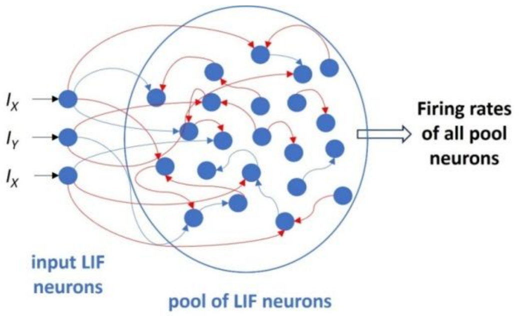

Fig. 1. SNN structure. Connections in red are excitatory synapses (with positive weight) and connections in blue are inhibitory (with negative weight) synapses

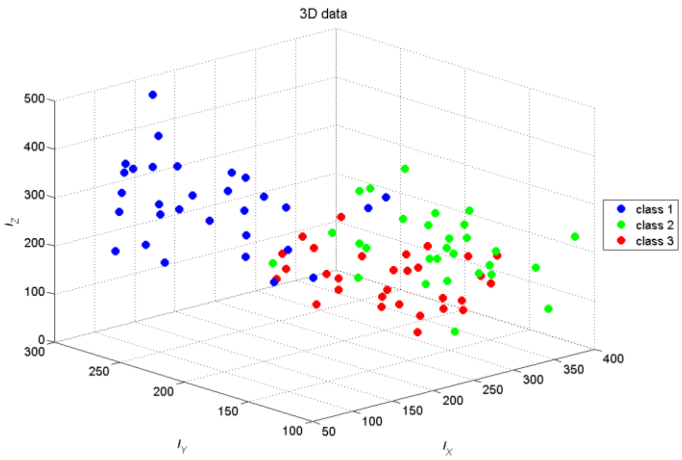

Fig. 2. 3D input data in three overlapping clusters shown by blue, green, and red dots

The constant external input currents I_X, I_Y, and I_Z to the SNN in Fig. 1 are converted to spike trains by the three input LIF neurons. For this aim, the input neurons are randomly connected by static synapses to the pool of neurons.

The artificially generated test data set consists of three groups, each with 30 randomly positioned dots in the three-dimensional space as shown in Fig. 2. The three groups (classes) overlap to some extent so they are not linearly separable. Each data point from the test set is presented as three constant input currents, I_X, I_Y, and I_Z, to the SNN input for 1 ms. The firing rates (number of spikes per given period of time) of all neurons compose the new feature vector extracted by the SNN. Before presenting each data item to the SNN, its connection weights were reinitialized to their initial values.

Two simulation experiments were performed: with a SNN pool of neurons with static connections (static SNN) and a SNN pool of neurons with dynamic connections trained via the STDP rule (STDP-trained SNN). Fig. 3 shows the firing rates of all neurons of the static and STDP-trained SNNs. Data samples from 1 up to 30 belong to class 1, those from 31 up to 60 belong to class 2, and the rest (from 61 up to 90) belong to class 3, as shown in Fig. 2.

While the observed static SNN neurons' firing pattern is random and depends only on the randomly generated connections within the pool of neurons, the activity of the STDP-trained SNN shows synchronization of all neurons in dependence on the input signal. Since all neurons in the pool have identical parameters, i.e., identical dynamic properties, no clustering in groups based on neuron characteristics, like reported by other authors, e.g., [4, 18], was observed. However, the overall activity in the dynamic SNN reflects the structure of the stimulating input, showing different mean levels of neuron firing rates for different data clusters.

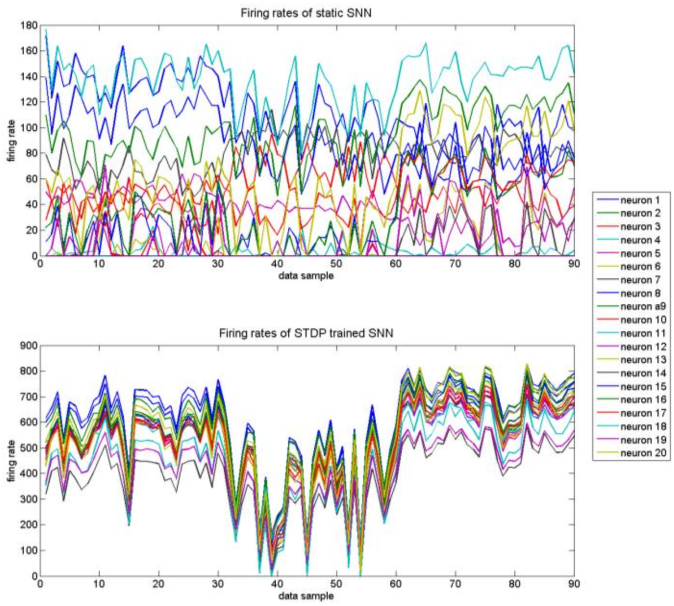

Fig. 3. Features extracted by the static SNN and the STDP trained SNNs

Since the features extracted from the 3D data are multidimensional (the SNN pool consists of 20 neurons in this experiment), the Andrews function that represents each vector of features as a continuous Fourier series curve [1] was chosen for visualization purposes. Usage of the principal components of features vector as input for the Andrews function enhances the visualization, so the Andrews plot from the PCA was chosen in the current work.

Fig. 4 shows a comparison of the Andrews plots from the PCA of both feature sets obtained from the static SNN and STDP trained SNN with the original 3D data features. Mean (solid line) and quartiles of ±25% (dashed lines) around it show the boundaries of three multidimensional data classes. While both original data and static SNN features are not clearly distinguishable, the STDP trained SNN features show almost no overlapping of the three classes.

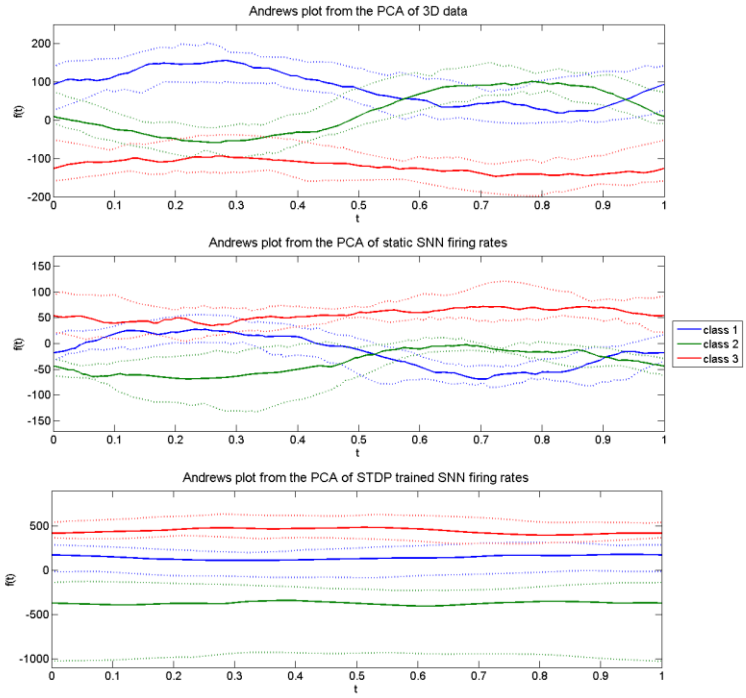

Fig. 4. Andrews plot from PCA of the 3D data and features extracted by the static and STDP trained SNNs. Solid lines represent mean values of the Andrews plot of all three classes while the dashed lines represent variance around the mean values

These results explain why classification or clustering of data improves by the features extracted by the STDP trained SNNs, as was reported in several works, e.g. [12, 16].

4. Echo state networks and IP tuning

Echo State Networks (ESN) are artificial RNNs proposed by Jaeger in 2002 [11]. They belong to a novel and rapidly developing family of reservoir computing approaches whose aim was the development of fast trainable RNN architectures able to approximate nonlinear time series. They consist of a pool of neurons with sigmoid activation function (usually the hyperbolic tangent) that has randomly generated recurrent connection weights like the pool of LIF neurons in Fig. 1. The reservoir state $R(t)$ for the current time instant t depends both on its previous state $R(t-1)$ and the current input $\mathrm{in}(t)$ as follows:

$$R(t) = (1-a)R(t-1) + a\tanh\left(AW^{\mathrm{in}}\mathrm{in}(t) + W^{\mathrm{res}}R(t-1) + B\right).$$

Here: W^{in} is a matrix of input to reservoir connection weights that are randomly generated; W^{res} is the internal reservoir connection weight matrix that is sparse and also randomly generated according to recipes from literature with a spectral radius below 1; a is the leaking rate parameter.

The vectors A (gain) and B (bias) are optional and can be tuned by the IP rule proposed in [25]. The main aim of this rule is to minimize the D_{KL} divergence between the actual and target distribution of reservoir neurons' states $R(t)$. In the case of hyperbolic tangent nonlinearity, the target distribution is Gaussian since it maximizes the capacity of the reservoir according to information theory.

The ESN output out(t) is calculated as the identity function of the linear combination of the concatenation of the input in(t) and reservoir states [$R(t)$ in(t)] weighted by the output weight matrix W^{out}:

(5) $$\mathrm{out}(t) = W^{\mathrm{out}}[R(t) \quad \mathrm{in}(t)].$$

The only trainable parameters of ESN are the output weights W^{out}. Since the output is a linear function, the least squares method is applied to train the ESN in a single iteration. For the aims of on-line training, the Recursive version of Least Squares (RLS) can be applied too.

In [14, 15, 17], the equilibrium states of the ESN reservoir were exploited as features extracted from various data sets, and it was demonstrated that this improves classification or clustering of various static and dynamic data sets.

5. Discussion

In [14], it was demonstrated that after IP tuning ESN reservoir is able to extract a set of features from a similarly designed three-dimensional data that allows for its better separation into several clusters. Particularly, it was shown that various combinations of two out of multiple extracted features can reproduce the original data structure. In [15], the same approach was applied for gray scale images clustering. Further tests on different multi-dimensional data sets (EEG for emotion recognition, sound camera images, industrial plant state, etc.) demonstrated improved classification or clustering results using features extracted by the IP-tuned ESN. In [17], the same effect was demonstrated on clustering of time series data from an experiment on human eye movement and its relation to the decision making.

Fig. 5 shows an example distribution of reservoir neurons' equilibrium states before and after IP tuning of an ESN reservoir from [15]. This particular example shows that IP tuning redistributes the neurons' states according to the data structure (which has 3 classes), thus allowing for better separation of the data.

For the comparative aim, Fig. 6 presents the distributions of features extracted by the SNN pool of neurons from the experiment above. As can be seen, again the static SNN does not reflect clearly the data structure, while the STDP trained SNN concentrates firing activity of neurons approximately around three maximal values.

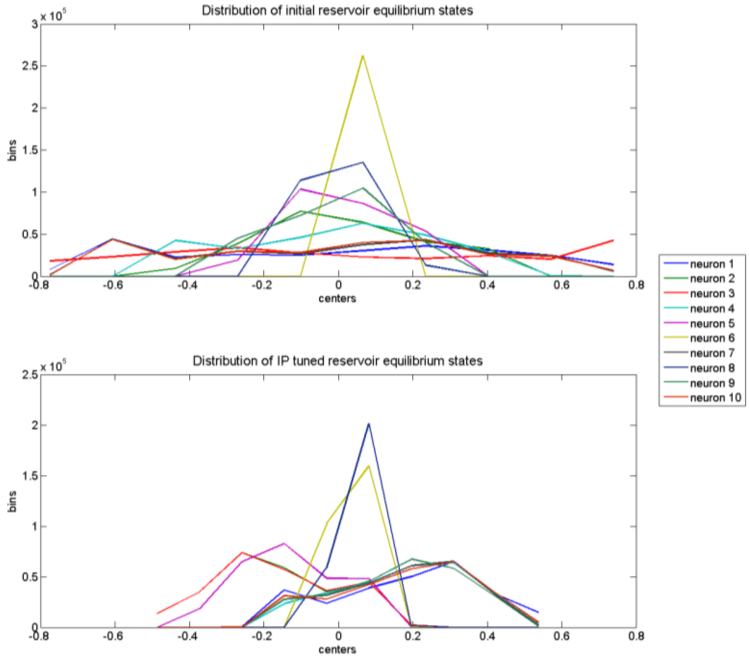

Fig. 5. Distributions of initial and IP-tuned ESN reservoir equilibrium states of neurons [15]

Both Figs 5 and 6 demonstrate that IP tuning and STDP training provokes formation of several groups of neurons whose activity (steady state for ESN and firing rate of SNN) has a Gaussian shape distribution with different mean values. Hence, both homeostatic plasticity rules have a similar effect of grouping neurons in several clusters with similar behavior.

While the STDP training aims synchronization of neurons with similar parameters determining their dynamics and groups them into clusters of identical behavior, the IP tuning of ESN reservoir connectivity aims at the target distribution of its output. However, it is obvious that both RNN and SNN structures map the original input data to a new feature set capturing the data structure.

As it was reported in some works, e.g. [2, 3], the STDP training of heterogeneous SNNs has a clustering effect with respect to the neuron's activity, i.e., grouping the similar neurons into clusters by creating a small-world connectivity among them.

The experiment reported above, however, uses a homogeneous SNN. It demonstrates that different driving input leads to synchronization of all neuron activity (firing rates) that can also be exploited for input data clustering.

The IP tuning achieves grouping of neuron activity according to a target (in reported cases Gaussian distribution). While it was reported that the best mean value of the target distribution should be zero, the tuning of the bias (*B*) term changes the mean value of the target distribution of individual neurons, so they are again grouped in several clusters in dependence on their bias. Changes in the input shift a given neuron activity closer or away from its cluster center – an effect similar to changed firing activity in SNNs.

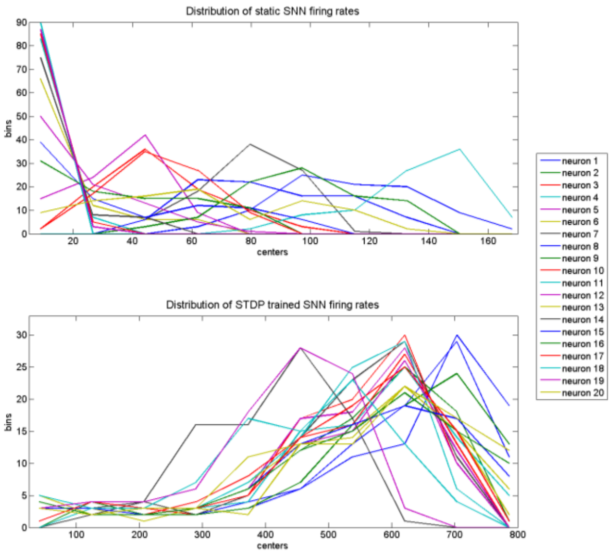

Fig. 6. Distributions of static and STDP-trained SNNs’ firing rates of neurons

While in the case of ESN with IP tuning choice of two out of multiple extracted features was shown to be a useful tool for their two-dimensional visualization, in the case of STDP trained SNNs Andrews function from the PCA of the extracted features allowed for low dimensional representation of the data structure in a similar way.

5. Conclusion

The performed simulation investigations on a test data set demonstrated the STDP training ability to tune the SNN neurons’ firing rates according to the data structure in a way similar to IP tuning of ESNs.

Even though SNNs and ESNs differ with respect to the signals they work with – discrete and continuous, respectively – both can adapt in real time to the incoming data streams via their homeostatic plasticity rules. This allows us to use both of them not only as non-linear models of complex dynamical systems, but also for capturing data structure.

In fact, the human brain relies on extracting useful information from the constantly incoming streams of sensory data in a similar way. A widely exploited example is the well-studied visual system, where multiple hierarchically positioned brain structures process the visual data streams, extracting position, orientation, movement direction, shapes, etc., from the visual flow. In [19, 20] STDP was proven to be the way our visual system adapts its internal “filters”, like in modern convolutional neural networks.

In addition, SNNs connectivity based on the spatial positions of neurons, like in the brain, helps to reflect not only temporal dependencies in time series but also their special interactions [12] that can be useful for spatially distributed sensor data like brain recordings.

Recent developments in neuromorphic hardware target both reservoir computing and SNNs, so the future of both types of NNs for AI applications will go in parallel.

References

1. Andrews, D. F. Plots of High-Dimensional Data. ─ Biometrics, Vol. **28**, 1972, No 1, pp. 125-136.
2. Basalyga, G., P. M. Gleiser, T. Wennekers. Emergence of Small-World Structure in Networks of Spiking Neurons through STDP Plasticity. ─ Adv. Exp. Med. Biol., Vol. **718**, 2011, pp. 33-9.
3. Belatreche, B., R. Paul. Dynamic Cluster Formation Using Populations of Spiking Neurons. – In: Proc. of IEEE World Congress on Computational Intelligence "WCCI'12", Brisbane, Australia, 10-15 June 2012.
4. Cateau, H., K. Kitano, T. Fukai. Interplay between a Phase Response Curve and Spike-Timing-Dependent Plasticity Leading to Wireless Clustering. ─ Phys. Rev. E, Vol. **77**, 2008, No 5, 051909.
5. Chaudhari, S., H. Nair, J. M. F. Moura, J. Paul Shen. Unsupervised Clustering of Time Series Signals Using Neuromorphic Energy-Efficient Temporal Neural Networks. – In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'21), Toronto, ON, Canada, 2021, pp. 7873-7877.
6. Debanne, D., Y. Inglebert. Spike Timing-Dependent Plasticity and Memory. – Current Opinion in Neurobiology, Vol. **80**, 2023, 102707.
7. Gewaltig, M.-O., M. Diesmann. NEST (Neural Simulation Tool). – Scholarpedia, Vol. **2**, 2007, No 4, 1430.
8. Guetig, R., R. Aharonov, S. Rotter, H. Sompolinsky. Learning Input Correlations through Nonlinear Temporally Asymmetric Hebbian Plasticity. ─ Journal of Neuroscience, Vol. **23**, 2003, pp. 3697-3714.
9. Huo, B., F. Li, S. Peng, H. Chen, S. Xin, H. Wang. Research on SNN Learning Algorithms and Networks Based on Biological Plausibility. ─ IEEE Access, Vol. **13**, 2025, pp. 95243-95256.
10. Izhikevich, E. M. Dynamical Systems in Neuroscience: The Geometry of Excitability and Bursting. The MIT Press, 2007.
11. Jaeger, H. Tutorial on Training Recurrent Neural Networks, Covering BPPT, RTRL, EKF, and the "Echo State Network" Approach. GMD Report 159, German National Research Center for Information Technology, 2002.
12. Kasabov, N. Time-Space, Spiking Neural Networks and Brain-Inspired Artificial Intelligence. Springer Series on Bio- and Neurosystems, 2019.
13. Khalid, T. K., S. Nasreen. A Survey of Feature Selection and Feature Extraction Techniques in Machine Learning. – In: Proc. of Science and Information Conference, London, UK, 2014, pp. 372-378.
14. Koprinkova-Hristova, P. Multi-Dimensional Data Clustering and Visualization via Echo State Networks. ─ In: R. Kountchev, K. Nakamatsu, Eds. New Approaches in Intelligent Image Analysis. Intelligent Systems Reference Library. Vol. **108.** Cham, Springer, 2016.
15. Koprinkova-Hristova, P., I. Georgiev, M. Raykovska. Echo State Network for Features Extraction and Segmentation of Tomography Images. ─ Computer Science and Information Systems, Vol. **21**, 2024, No 1, pp. 379-393.
16. Koprinkova-Hristova, P., D. Penkov, S. Nedelcheva, S. Yordanov, N. Kasabov. On-Line Learning, Classification and Interpretation of Brain Signals Using 3D SNN and ESN. – In: Proc. of International Joint Conference on Neural Networks (IJCNN'23), Gold Coast, Australia, 2023, pp. 1-6.

17. Koprinkova-Hristova, P., M. Stefanova, B. Genova, N. Bocheva. Echo State Network for Classification of Human Eye Movements during Decision Making. – In: L. Iliadis, I. Maglogiannis, V. Plagianakos, Eds. Artificial Intelligence Applications and Innovations (AIAI'18). IFIP Advances in Information and Communication Technology. Vol. **519**. Cham, Springer, 2018.
18. Li, X., M. Small. Enhancement of Signal Sensitivity in a Heterogeneous Neural Network Refined from Synaptic Plasticity. – New J. Phys., Vol. **12**, 2010, 083045.
19. Majumdar, P. Spiking Neural Networks: A Comprehensive Review of Diverse Applications, Research Progress, Challenges, and Future Research Directions. – Evolving Systems, Vol. **16**, 2025, 125.
20. Masquelier, T., R. Guyonneau, S. J. Thorpe. Spike Timing Dependent Plasticity Finds the Start of Repeating Patterns in Continuous Spike Trains. – PLoS ONE, Vol. **3**, 2008, No 1, e1377.
21. Masquelier, T., S. J. Thorpe. Unsupervised Learning of Visual Features through Spike Timing Dependent Plasticity. – PLoS Comput Biol., Vol. **3**, 2007, No 2, e31.
22. Mutlag, W. K., S. K. Ali, Z. M. Aydam, B. H. Taher. Feature Extraction Methods: A Review. – J. Phys.: Conf. Ser., Vol. **1591**, 2020, 012028.
23. Rohr, V., R. Berner, E. L. Lameu, O. V. Popovych, S. Yanchuk. Frequency Cluster Formation and Slow Oscillations in Neural Populations with Plasticity. – PLoS ONE, Vol. **14**, 2019, No 11, e0225094.
24. Rotter, S., M. Diesmann. Exact Digital Simulation of Time-Invariant Linear Systems with Applications to Neuronal Modeling. – Biol. Cybern., Vol. **81**, 1999, pp. 381-402.
25. Safa, A. Continual Learning with Hebbian Plasticity in Sparse and Predictive Coding Networks: A Survey and Perspective. – Neuromorph. Comput. Eng., Vol. **4**, 2024, 042001.
26. Schrauwen, B., M. Wandermann, D. Verstraeten, J. J. Steil, D. Stroobandt. Improving Reservoirs Using Intrinsic Plasticity. – Neurocomputing, Vol. **71**, 2008, pp. 1159-1171.
27. Tal, A., N. Peled, H. T. Siegelmann. Biologically Inspired Load Balancing Mechanism in Neocortical Competitive Learning. – Frontiers in Neural Circuits, Vol. **8**, 2014, 18.

Fast-track. Received: 03.02.2026, Revised version: 01.03.2025, Accepted: 09.03.2025