Neuromorphic
Computing and
Engineering

PAPER • OPEN ACCESS

Time–frequency analysis using spiking neural network

To cite this article: Moshe Bensimon *et al* 2024 *Neuromorph. Comput. Eng.* **4** 044001

NEUROMORPHIC
Computing and Engineering

CrossMark

OPEN ACCESS

RECEIVED
5 June 2024

REVISED
15 September 2024

ACCEPTED FOR PUBLICATION
27 September 2024

PUBLISHED
10 October 2024

PAPER

Time–frequency analysis using spiking neural network

Moshe Bensimon[1,2,3], **Yakir Hadad**[1,3], **Yehuda Ben-Shimol**[1] **and Shlomo Greenberg**[1,2,*]

[1] School of Electrical and Computer Engineering, Ben-Gurion University of the Negev, Beer-Sheva, Israel
[2] Computer Science Department, Sami Shamoon College of Engineering, Beer-Sheva, Israel
[3] Contribute equally.
[*] Author to whom any correspondence should be addressed.

E-mail: shlomgr@sce.ac.il

Keywords: SNN, STDP, neuromorphic computing, time–frequency analysis, feature extraction, frequency detection, EEG.

Abstract

Time–frequency analysis plays a crucial role in various fields, including signal processing and feature extraction. In this article, we propose an alternative and innovative method for time–frequency analysis using a biologically inspired spiking neural network (SNN), encompassing both a specific spike-continuous-time-neuron-based neural architecture and an adaptive learning rule. We aim to efficiently detect frequencies embedded in a given signal for the purpose of feature extraction. To achieve this, we suggest using an SN-based network functioning as a resonator for the detection of specific frequencies. We developed a modified supervised spike timing-dependent plasticity learning rule to effectively adjust the network parameters. Unlike traditional methods for time–frequency analysis, our approach obviates the need to segment the signal into several frames, resulting in a streamlined and more effective frequency analysis process. Simulation results demonstrate the efficiency of the proposed method, showcasing its ability to detect frequencies and generate a Spikegram akin to the fast Fourier transform (FFT) based spectrogram. The proposed approach is applied to analyzing EEG signals, demonstrating an accurate correlation to the equivalent FFT transform. Results show a success rate of 94.3% in classifying EEG signals.

1. Introduction

Neuromorphic computing aims to substitute traditional approaches with neuromorphic algorithms inspired by the human brain. Recent advancements in the field of neuromorphic computing have shown that spiking neural networks (SNNs) manage to achieve performance equivalent to conventional deep learning methods with significantly reduced power consumption in various fields such as image processing, sound classification, and biophysical signals analysis [1, 2].

SNNs have garnered significant interest due to their ability to mimic the behavior of biological neurons. Spiking neurons (SNs) are characterized by the way they transmit information. Whenever the SN internal membrane (charged by asynchronous input spikes) potential reaches a specific predefined threshold, the neuron fires and generates a signal that travels to other SNs. Various mathematical models have been proposed to accurately model these SNs, each offering a different balance between simplicity and biological realism [3]. SNN implementations include analog, digital, and mixed hardware approaches [4, 5].

The leaky integrate-and-fire model (LIF) is the most common SN model [6]. This relatively simple model represents the neuron as a leaky capacitor that accumulates incoming asynchronous spikes, which charge the capacitor until a threshold is reached, causing it to 'fire' an output spike. Despite its simplicity, the LIF model effectively represents basic bio-inspired neuronal dynamics and is computationally efficient.

Izhikevich [7] introduced a more sophisticated neuron model capable of faithfully replicating a broad spectrum of biologically observed behaviors using various parameters and two differential equations. This model balances computational efficiency and biological plausibility while capturing a wider range of neuronal firing patterns, and still remaining relatively easy to simulate [8].

Hodgkin and Huxley [9] introduce the most complex and accurate bio-inspired SN model. The Hodgkin–Huxley model uses a set of four non-linear ordinary differential equations to describe the intricate dynamics of ion channels in the neuron's membrane. By modeling the flow of sodium and potassium ions with high precision, the Hodgkin–Huxley model offers a more detailed and biologically accurate representation of neuronal behavior, though it is more computationally demanding.

Each of these models serves different purposes and is applied based on the application's specific need, whether prioritizing simplicity for computational efficiency or biological accuracy.

Conventional artificial neural networks (ANNs) use the well-known backpropagation approach as the dominant training algorithm, which adjusts the network weights based on the gradient of the loss function. However, training SNNs presents unique challenges due to their being discrete and event-driven, and due to their computational power limit. Unlike ANNs, SNNs rely on the timing of spikes rather than continuous activations, which requires different training approaches.

Among the various algorithms developed for SNN training, the spike-timing dependent plasticity (STDP) is the most prevalent [10, 11]. STDP is a biologically inspired learning rule where the synaptic strength is adjusted based on the relative timing of pre-and post-synaptic spikes [10, 12]. When a presynaptic spike precedes a postsynaptic spike, the synapse is strengthened; otherwise, the synapse is weakened. This approach is aligned with biological learning processes and effectively learns complex spatiotemporal representations [12].

Although the STDP is primarily employed as an unsupervised learning algorithm, various adaptations to STDP have recently been proposed to tailor it to the configuration of supervised learning [13–16].

Following are some notable supervised STDP approaches: (a) reward-modulated STDP (R-STDP), which combines STDP with reinforcement learning by modulating synaptic weight changes based on an external reward signal [17]. (b) Supervised STDP with Error Signals [18], in which the synaptic changes are adjusted based on both the spike timing and the gradient of an error function. (c) Backpropagation through time, in which the gradients of an error function are computed with respect to the timing of spikes and then used to modify synaptic weights [19, 20]. (d) STDP with Supervisory Signals [21], where STDP is combined with external supervisory signals that modify the STDP learning rule by providing feedback on how synaptic weights should change to minimize the error between network output and target output.

In this work we propose a unique supervised STDP approach which its principle is quite similar to the STDP with Supervisory Signals method [21]. Our proposed approach does not make use of any kind of backpropagation, nor does it apply reinforcement learning.

Numerous studies have demonstrated that SNs can be described as oscillators [22, 23]. Klaus Stiefel and Ermentrout [22] show how a regularly SN can be viewed as an oscillator and how the phase-response curve describes the response of the neuron's spike times to small perturbations. They also review the insights gained from the concept that SNs can be defined as oscillators and their relevance for neuroscience. Underlying oscillators generate most physical and biological systems characterized by regular rhythmicity. Therefore, neurons that produce motor rhythms and cortical neurons can be viewed as oscillators.

Kashchenko *et al* [24] propose a model of a neural environment capable of storing information presented in a waveform. The spiking model exhibits fundamental characteristics that enable a signal delay over time, which allows the ability to generate a defined phase shift by adjusting the neuron parameters [25, 26].

The description of a neuron as an oscillator is essentially an abstraction of the complexity of the SN model and provides insight into neural oscillators. Therefore, it is relevant for characterizing neural oscillations in the brain (such as in EEG applications). In this work, we apply this important inherent SN feature to utilize the SNs to build a frequency detector and for time–frequency analysis. Various methods for applying time–frequency analysis have been proposed for feature extraction [27, 28]. Adeli *et al* [29] use a cochleagram feature extraction method incorporating a bio-inspired model for time–frequency processing.

The extraction of features in the frequency domain is a common approach for various applications [30–32]. Usually, this involves dividing the signal into overlapping segments (i.e. frames) and extracting features from each segment individually. However, this approach introduces several drawbacks, including computation delays, redundant calculations, and limited resolution [33, 34]. We adopt a continuous-time approach for feature extraction, which overcomes these limitations, allowing a more efficient and accurate time signal analysis [35].

SNNs are particularly well-suited for bio-medical signal analysis due to their ability to analyze time series signals as well as support event-driven signals. Burelo *et al* [36] present a neuromorphic SNN that detects epileptic high-frequency oscillations using EEG signals. Costa *et al* [37] present a real-time SNN hardware framework for robust compression and detecting epileptiform patterns in intraoperative electrocorticography ECoG.

This work is based on a unique low-power SN, the spike continuous time neuron (SCTN), previously introduced by Bensimon *et al* [2]. The SCTN model is based on the classical LIF SN, incorporating several modifications to enhance its functionality. These enhancements include seven leak modes, three activation functions, and a dynamic threshold setting, allowing for versatile neuron configurations. Furthermore, the

SCTN supports embedded learning modules based on STDP, enabling adaptive learning capabilities. In a recent study, we demonstrated the efficiency of using an SCTN-based phase-shifting circuit for frequency detection and extracting MFCC-like features in acoustic applications [38]. This paper proposes an innovative method for time–frequency analysis using SCTN-based biologically inspired SNN.

Our main contributions are listed below:

(i) We suggest using an SCTN-based network for feature extraction in the frequency domain, which functions as a resonator circuit to detect specific frequencies.
(ii) A modified supervised STDP learning approach is presented to train the SNN network and optimize the weights and biases for targeted frequency detection.
(iii) We utilize the proposed neuromorphic architecture for feature extraction in the frequency domain to demonstrate a high correlation with the common fast Fourier transform (FFT) features. The proposed approach is applied to analyze and classify EEG signals efficiently. Simulation results demonstrate the ability of the proposed method to extract frequency features efficiently.

The rest of the paper is organized as follows: section 2 describes the proposed neuron model and the SNN network used for frequency detection. Section 3 presents the proposed STDP-based supervised learning. Section 4 outlines the experimental results, and section 5 concludes the paper.

2. SCTN-based architecture for frequency analysis

This section describes the mathematical model of the SCTN SN and the proposed SCTN-based resonator circuit used as a frequency detector.

2.1. SCTN-Spike continuous time neuron model

This section briefly describes the SCTN model presented in [2]. The SCTN model is given by:

$$Vm(t) = \alpha \cdot \left(Vm(t-1) + \sum_{j=1}^{N} W_j \cdot I_j(t) \right) + \Theta \tag{1}$$

where the membrane potential, $Vm(t)$, of a neuron with N input synapses can be further expressed as follows:

$$Vm(t) = \begin{cases} Vm_0, t=0 & \\ \alpha \cdot \left(Vm(t-1) + \sum_{j=1}^{N} w_j \cdot I_j(t) \right) + \Theta, & t \cdot \mathrm{mod}(\mathrm{LP}) = 0 \\ Vm(t-1) + \sum_{j=1}^{N} w_j \cdot I_j(t) + \Theta, & t \cdot \mathrm{mod}(\mathrm{LP}) \neq 0 \end{cases} \tag{2}$$

where $I_j(t)$ is the synapse input, w_j represents the corresponding weight, and Θ represents the neuron bias. Each SCTN cell encompasses a leaky integrator whose time constant is defined by a specific parameter $\alpha = 1 - 2^{-\mathrm{LF}}$ [2]. This enables the SCTN to delay incoming signals based on a projected leakage time constant. The parameters, leakage factor (LF) and leakage period (LP) represent the neuron's integrator leakage rate and the integrator operation rate, respectively. When the membrane potential (V_m) exceeds the threshold value, the neuron fires a spike, and then the membrane potential discharges to a resting potential (like in a biological neuron) [7].

Bensimon *et al* [38] demonstrate the ability of the SCTN neuron to generate a phase-shifting emulating a common analog first-order RC circuit. A spiking LIF neuron model is often compared to an RC circuit, where the capacitor (C) symbolizes the neuron membrane potential and the resistor (R) represents the neuron leakage [39]. The transfer function of an RC circuit is given by:

$$H(j\omega) = \frac{V_o}{V_{\mathrm{in}}} = \frac{1 - j\omega \mathrm{RC}}{1 + (\omega \mathrm{RC})^2} \tag{3}$$

where the radial frequency is $\omega = 2\pi f$. For the cutoff radial frequency, $\omega_0 = 1/\mathrm{RC} = 1/\tau$, the magnitude and phase are given by:

$$|H(j\omega_0)| = \frac{1}{\sqrt{2}} \tag{4}$$

$$arg(H(j\omega_0)) = -\frac{\pi}{4}. \tag{5}$$

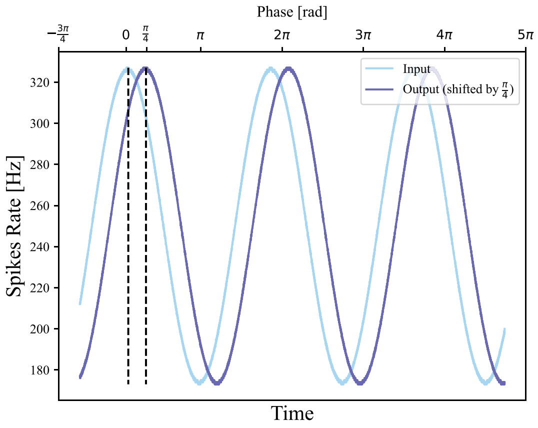

Figure 1. A phase shift of $\frac{\pi}{4}$ using a single SCTN neuron. The y-axis depicts the spike rate observed within a sampling window T.

Therefore, for an input frequency of ω_0, a phase shift of $-\frac{\pi}{4}$ is observed.

The SCTN has the ability to postpone the incoming signals according to a predefined leakage time constant [2], i.e. set a delay τ which is given by:

$$\tau = T_{\text{clk}} \cdot \frac{1}{1-\alpha} \cdot (1 + \text{LP})\,. \tag{6}$$

Hence, ω_0 is given by:

$$\omega_0 = \frac{1}{\tau} = \frac{1-\alpha}{T_{\text{clk}} \cdot (1 + \text{LP})} \tag{7}$$

and the cutoff frequency f_0 is given by:

$$f_0 = \frac{f_{\text{clk}} \cdot (1-\alpha)}{2\pi \cdot (1 + \text{LP})} = \frac{f_{\text{clk}}}{2^{LF} \cdot 2\pi \cdot (1 + \text{LP})} \tag{8}$$

where $f_{\text{clk}} = \frac{1}{T_{\text{clk}}}$ represents the system clock, Therefore, for an appropriate choice of the LP and LF parameters, the SCTN has the ability to generate a phase shift of $\frac{\pi}{4}$ (for a given frequency f_0).

Figure 1 demonstrates a $\frac{\pi}{4}$ phase shift for a sinusoidal input signal achieved by tuning the SCTN neuron parameters. The input signal is encoded using the pulse density modulation (PDM) method to represent the sampled analog signal as a stream of pulses. Each sample of the sine signal is encoded into a predefined number of pulses according to the amplitude, and is represented by a 500-bits binary vector. The total number of "1" in the binary vector represents the number of spikes corresponding to the sampled analog value and, therefore, to the phase shift. The order of the "1" in the vector has no importance and is randomly spread within the binary vector (to increase noise immunity). For each sample, a random shuffling of the "1" in the vector is performed, and therefore, the phase shift is not expressed only in a shifted binary vector, but it relates to the total number of spikes ("1") in the vector.

2.2. SNN-based architecture for frequency detection

This section presents the SCTN-based resonator circuit, which serves as a frequency detector utilizing the SCTN basic phase shifting feature. Leveraging the fundamental phase-shifting feature of SCTN, we suggest using SNN architecture to detect the embedded frequency components inherent in the given signal.

Figure 2 shows the SCTN-based network that operates as a resonator circuit. The proposed circuit comprises four sequentially linked SCTNs. Each neuron is designed to perform a phase shift of $\frac{\pi}{4}$, while the phase shift at the output of the first neuron (χ) ranges from 0 to $\frac{\pi}{4}$. The first neuron, SN_1, has two input synapses: (a) the input signal and (b) feedback from the fourth neuron (SN_4) representing a shift of $\chi + \frac{3\pi}{4}$. The feedback synapse undergoes an additional shift of $\frac{\pi}{4}$. Therefore, achieving a total shift of π relative to χ.

Figure 3 illustrates the equivalent RC-based oscillator circuit, known as a phase-shift oscillator. This basic RC oscillator generates a sine-wave output signal by employing feedback derived from the resistor–capacitor (RC) network. The capacitor stores the electric charge akin to the membrane potential in SNs. The four RC units are concatenated to achieve a total phase shift of 180 degrees.

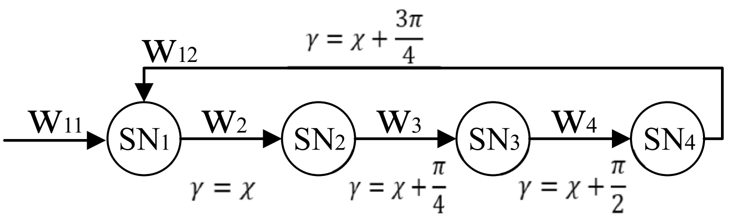
W12
$\gamma = \chi + \frac{3\pi}{4}$
W11
SN1
W2
SN2
W3
SN3
W4
SN4
$\gamma = \chi$
$\gamma = \chi + \frac{\pi}{4}$
$\gamma = \chi + \frac{\pi}{2}$

Figure 2. SNN-based resonator architecture.

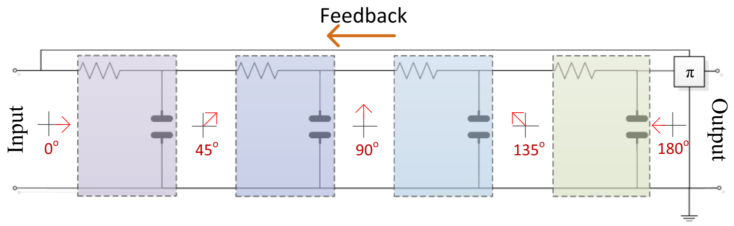

Figure 3. Equivalent RC based oscillator circuit.

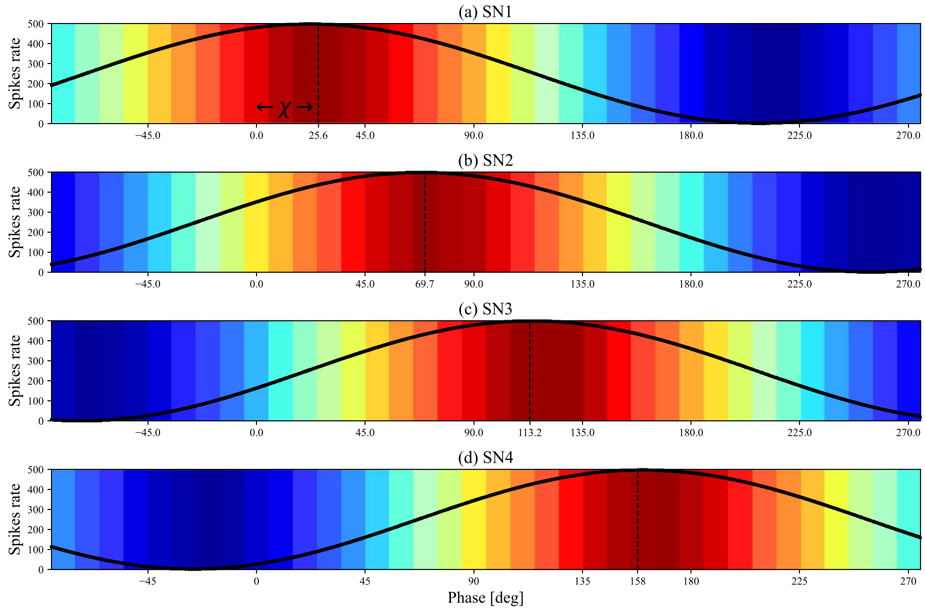

Figure 4. The output of the four SCTN neurons: (a) shift of $\chi\,(25^\circ)$ by SN$_1$, (b)–(d) shift of $\frac{\pi}{4}$ by SN$_2$, SN$_3$, SN$_4$. Each color bin reflects the number of spikes in the sampling interval.

Figure 4 depicts the four SCTN neurons' output, demonstrating a shift of $\frac{\pi}{4}$ by each neuron. Setting the phase shift of the first neuron as χ allows an additional degree of freedom in designing and determining a band-pass filter within the range of $f_0 \pm \Delta f_0$.

The desired χ can be achieved by training and synaptic weights update. Figure 5 depicts the synapse weights w_{ij} and bias Θ_1 as a function of the shift phase χ of the first neuron (SN$_1$). The superposition of the two input synapses to SN$_1$ yields a shift phase χ in the range of 0 to $\frac{\pi}{4}$.

The resonator circuit has the capability to detect a broad spectrum of frequencies embedded in a time–series signal utilizing the SCTN basic phase shifting feature. The LP and LF parameters can be configured to detect any desired frequency, represented by the resonance frequency, f_0. Upon detecting frequencies in the range of $f_0 \pm \Delta f_0$, the output neuron (SN$_4$) fires a train of pulses, where the maximum rate is achieved at f_0.

The filter quality factor of a filter, Q, defines its band-pass in relation to its center frequency and indicates the filter's resonant properties. The higher the Q, the more resonant the filter, and the narrower the range of

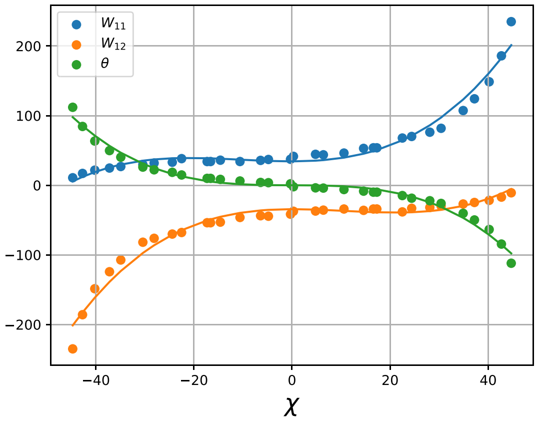

Figure 5. Synapses weights (w_{ij}) and bias (Θ_i) as a function of the phase shift χ.

frequencies that are allowed to pass. The Q factor is given by $Q = f_c/\Delta f$. The choice of the χ parameter, i.e. the initial phase shift of the first neuron, directly affects the shape of the resonator output signal and its Q-factor, as indicated in figure 6. A linear dependency of Q relative to χ is observed.

3. A modified STDP-based supervised learning

This section presents the proposed supervised modified-STDP learning algorithm and provides a detailed description of the training process of the spiking network. The training process of the network is aimed at adjusting the synaptic weights to detect the desired frequency (f_0). First, we need to determine the LF and LP parameters of each SCTN using equation (8). Then, the SNN training phase is carried out using a modified STDP learning rule for adjusting and determining the W_{ij} and Θ_i of the four neurons to achieve the desired frequency response.

3.1. Loss function

The loss function is defined as the mean squared error (MSE) between each SCTN neuron's output $\mathrm{SN}^{\mathrm{out}}$ and the desired output V^{ref}. This MSE error should be minimized to ensure a phase shift of $\xi_i^r \pm \Delta\xi$, and a magnitude of $g_i^r \pm \Delta g$. Where ξ_i, g_i, and ξ_i^r, g_i^r are the actual and the desired phase and gain, respectively. For simulation purposes, we generate four predefined desired signals (one per each SCTN neuron's output) for supervised training. The loss function for each SN is formalized as follows:

$$\mathrm{MSE} = \frac{1}{L}\sum_{j=1}^{L}\left(\mathrm{SN}_j^{\mathrm{out}}\left(\xi_j, g_j\right) - V_j^{\mathrm{ref}}\left(\xi_j^r, g_j^r\right)\right)^2 \tag{9}$$

where L is the number of sampling intervals (depicted in figure 4) in one sine cycle, and $\mathrm{SN}_i^{\mathrm{out}}$ represents the sum of spikes acquired over a sampling interval T_i where i ranges from 1 to L. The total loss is calculated by averaging the MSE of the four SCTN neurons:

$$\mathrm{Loss} = \frac{1}{4}\sum_{i=1}^{4}\mathrm{MSE}_i. \tag{10}$$

3.2. SNN training algorithm

This section presents the learning algorithm used to train the SNN model. To determine the values of the weight and biases, we employ a modified STDP learning algorithm. Unlike the common unsupervised STDP method [40], we suggest applying a supervised learning approach. This means that updating the network weights is not triggered only when the SCTN fires a pulse but also according to the reference desired signal.

Below is a detailed description of the learning process. First, we randomly determine W_{ij} and Θ_i and the values of the neuron's synapses. In the first step, the goal is to adjust the Θ_i bias values in order to produce at each SCTN output an average pulse rate of $T/2$ pulses per cycle, assuming a predefined encoding ratio of T pulses for the maximum amplitude. This actually represents the DC of the signal and allows the stretching of

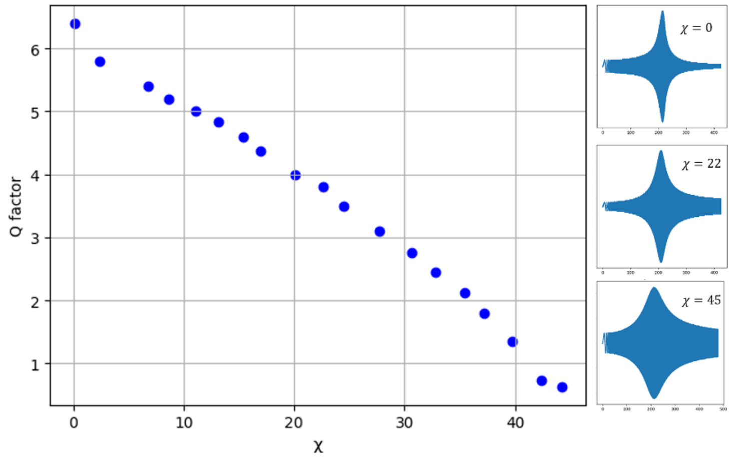

Figure 6. Left: Q factor as a function of χ. Right: the resulting filter for different phase shifting χ $(0^\circ, 10^\circ$, and 22° degrees).

the amplitude in the entire pulse range (0 to T). To obtain the desired pulse rate, we change the bias values Θ_i during the learning process, using an inverse Sigmoid-like function as follows:

$$\Delta\Theta = \left(\frac{2x - T}{T}\right)^2 \cdot \operatorname{sign}\left(\frac{T}{2} - x\right) \tag{11}$$

where x is the current pulse rate, After the Θ_i has been tuned and the desired average pulse rate has been achieved, the learning process continues in order to determine the network weights through supervised STDP learning.

We propose a supervised training approach in which the STDP learning rule is modified by providing feedback on how the synaptic weights should change to minimize the error between network output and target output. Instead of relying purely on spike timing, synaptic changes are guided by the error (mismatch) between the desired value, represented by the encoded reference signal (V_i^{ref}), and the existing value as observed in the SCTN neuron output (SN_i^{out}). The weight updates are applied based on both the pre-synaptic spike timing and the mismatch error signal according to the decision rules defined in table 1.

Figure 7 illustrates the learning rule applied in each scenario. We update the weight only when there is a contradiction between the actual SN output (firing a spike or not) and the desired encoded reference signal. If the SN fires a spike when the reference signal indicates that no spike should be fired, the synapse is weakened by applying the Anti-STDP* rule (shown in red in figure 7) to adjust the synapse weight. On the other hand, when a spike should have been generated according to the reference signal and the SN does not fire a spike, the synapse should be strengthened, and therefore, the weight should be updated according to the STDP* rule. The modified STDP* and anti-STDP* refer to the fact that no weight update is made in case a pre-synaptic spike occurs after the post-synaptic spike.

During the learning process, the weights of each SCTN neuron are updated according to the following rule and only in a case where $t_{\text{pre}} > t_{\text{post}}$:

$$\Delta W_{i,j} = \sum_{k \in S_{\text{pre}}} \sum_{l \in S_{\text{post}}} \begin{cases} +A \cdot \mathrm{e}^{-|t_k - t_l|/\tau_s}, & \text{if } V_i^{\text{ref}} = 1 \,\text{and}\, \text{SN}_i^{\text{out}} = 0 \\ -A \cdot \mathrm{e}^{-|t_k - t_l|/\tau_s}, & \text{if } V_i^{\text{ref}} = 0 \,\text{and}\, \text{SN}_i^{\text{out}} = 1 \end{cases} \tag{12}$$

where A represents the learning rate, and τ_s is the time constant used to control the synaptic potentiation and depression.

The time constant τ_s may be adjusted according to the absolute value of the phase error. A large error enables a larger time interval for controlling the required weight changes for convergence and vice versa. Therefore, τ_s is defined as a function of the phase error and is given by:

$$\tau_s = K \cdot \frac{|\xi_i^r - \xi_i|}{2\pi} \tag{13}$$

where K is determined as a function of the sampling frequency.

The magnitude of the learning rate decreases exponentially with the absolute value of the timing difference. When multiple spikes are simultaneously fired, the required weight change is calculated by

Table 1. Decision function applied to the supervised STDP.

$\mathrm{SN}_i^{\mathrm{out}}$	V_i^{ref}	Decision function
0	0	No action
0	1	STDP
1	0	Anti-STDP
1	1	No action

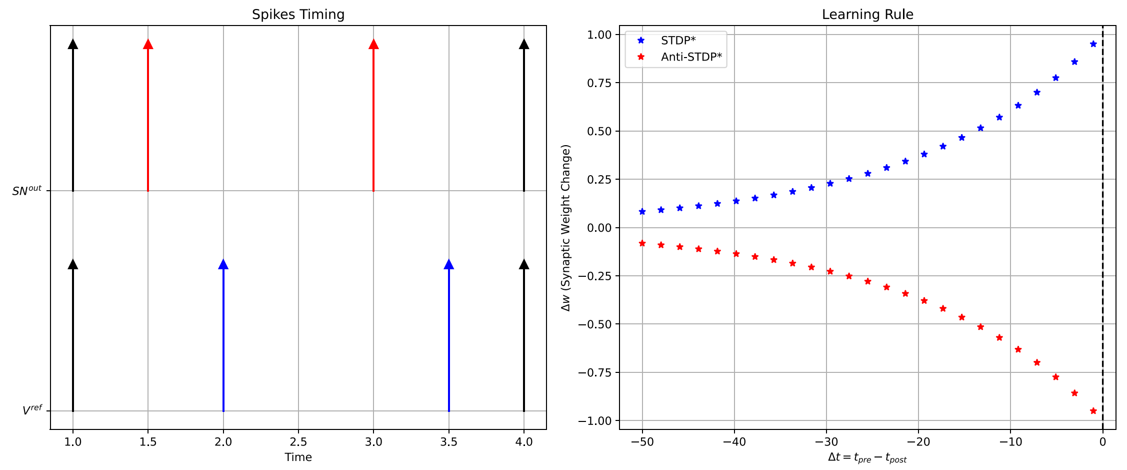

Figure 7. Left: pre-synaptic spike timing as observed in the SCTN neuron output ($\mathrm{SN}^{\mathrm{out}}$), and the desired target timing signal (V^{ref}). Right: the applied modified STDP learning rule.

summing the individual changes derived from all possible spike pairs. S_{pre} and S_{post} are sets of spikes representing the pre and post-synaptic neurons, respectively, where t_k stands for the timing of spike k (from S_{pre} set), and t_l represents the timing of spike l (from S_{post} set).

Choosing a relatively small learning rate A ensures robust learning [41]. The time constant τ_s (usually on the order of several milliseconds) defines the time interval used to control the weight changes. The learning rate A depends on the difference between the actual and the desired amplitude and is defined by:

$$A = (1+\rho)\cdot 10^{-4} \qquad (14)$$

where:

$$\rho = \frac{|g_i^r - g_i|}{g_i^r}. \qquad (15)$$

As stated before, the updating of the network weights is not triggered only when the SCTN fires a pulse but also according to the value of the reference desired signal. The reference-shifted sine signal is encoded into a pulse-like signal (using PDM) to compare the desired signal to the SCTN output. As part of the encoding process, the spikes representing the sampled signal were randomly distributed within the binary vector.

The desired number of spikes for a given sample are used as a reference for the STDP learning rule. The decision to update the SCTN weights depends on both the SCTN output ($\mathrm{SN}^{\mathrm{out}}$) and the encoded reference signal (V^{ref}), as depicted in table 1. Strengthening or weakening of the synapse weight is carried out according to the STDP rules given by equation (12).

3.3. Avoiding local minima

The initial values of the biases Θ_i (as described in section 3.2) may lead to convergence to a local minima. In practice, during the learning process, there may be a scenario where the phase converges while the amplitude error remains constantly large. To avoid this, we suggest to continue simultaneously updating both the bias Θ_i and the weights W_{ij}, according to the following equations:

$$\Theta_i = \Theta_i \pm \Delta\psi,\; g_i^r - g_i \gtrless 0 \qquad (16)$$

$$W_{ij} = W_{ij} \mp 2\Delta\psi,\; g_i^r - g_i \gtrless 0 \qquad (17)$$

where $\Delta\psi$ is a constant value empirically set to 0.002.

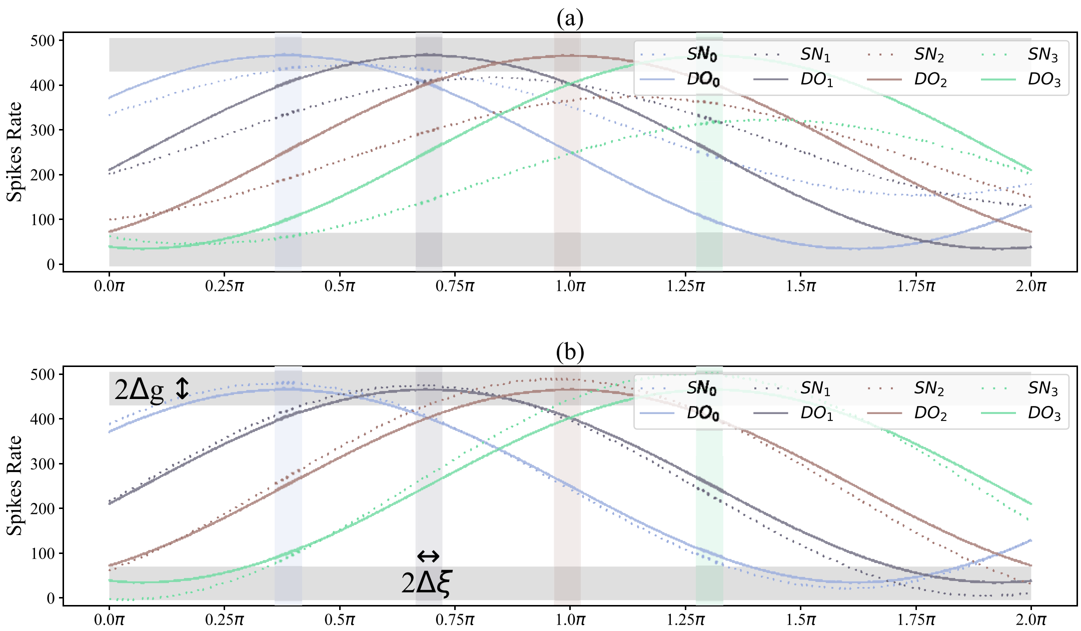

Figure 8. The learning process: (a) the neuron's output for random weight values, and (b) convergence of the network to the desired output (DO), for $\Delta g = 15$ and $\Delta\xi\ = 3^\circ$.

Using these rules preserves the desired average pulse rate ($T/2$ pulses per cycle) and enables stretching of the amplitude. Upon detecting a local minimum, i.e. if the amplitude error remains fixed and not converged, the bias determined in the initial learning phase is changed. Thus, avoiding falling into a local minimum is enabled by repeating the STDP-based learning process to determine the network weights, resulting in decreasing amplitude error. Simulations show that the improved learning process optimizes the learning speed and avoids convergence to local minima.

3.4. Convergence of the learning process

The learning process aims to converge to the desired signal such that the amplitude is extended over a predefined range, and each SCTN neuron generates the appropriate shift phase (i.e. χ for the first neuron and $\frac{\pi}{4}$ for the others). Figure 8 depicts the convergence learning process. The x-axis represents the phase, and the y-axis is the magnitude encoded according to the pulse rate (like PDM). Figure 8(a) depicts the initial SCTN neuron's outputs for random weight values. Figure 8(b) demonstrates the convergence of the SNN network to the desired values for the 4 SCTN neurons. Each neuron's current and desired signals are marked with solid and dashed lines, respectively. The desired phase ranges ($\xi_i^r \pm \Delta\xi$) for each neuron are indicated by colored longitudinal bars. The stopping condition occurs when all neurons meet the following two criteria: (a) the phase converges to the desired value: $\xi_i^r \pm \Delta\xi$. (b) the magnitude converges to a selected range: $g_i^r \pm \Delta g$. The amplitude values are represented as a function of the desired pulse rate in a given sampling interval (T). Figure 9 depicts the MSE convergence and the amplitude and phase errors during the learning process for the 4 SCTN neurons. Figure 9(a) demonstrates a local minimum scenario in which the phase error is well converged for all four neurons (after about 130 epochs) while the amplitude error of SN_3 and SN_4 remains high. Therefore, the suggested improved learning process is triggered, and the bias Θ_i and weights W_{ij} are updated according to equations (16) and (17). The proposed change of the initial bias first causes an increase in the phase error, as expressed in figure 9(b). At the same time, the amplitude of all neurons converges, as depicted in figure 9(c). Finally, the proposed learning process is well converged. A minimal error is achieved after about 650 epochs for both the phase and the amplitude. Figures 9(d) and (e) depict the weights update and bias update during the learning process, respectively.

4. Experiential result

4.1. Database

To evaluate the efficiency of the proposed approach, we applied the SNN-based resonator to detect the frequencies embedded in EEG signals. The EEG signal is composed of 5 subbands: Delta (0.1–4 Hz), Theta

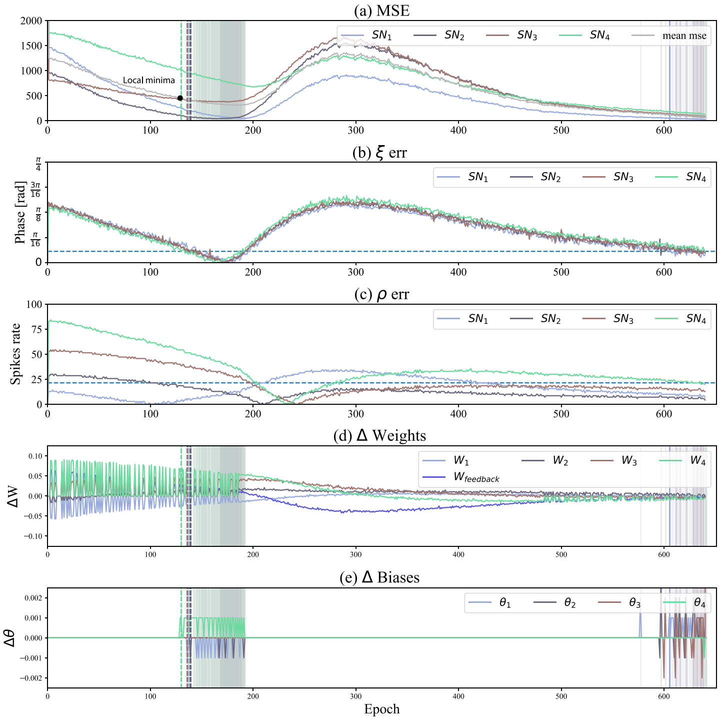

Figure 9. Error convergence during the learning process: (a) MSE, (b) amplitude, (c) phase error, and the (d) weights update, and (e) bias updates. The horizontal dashed lines indicate the maximum allowed error.

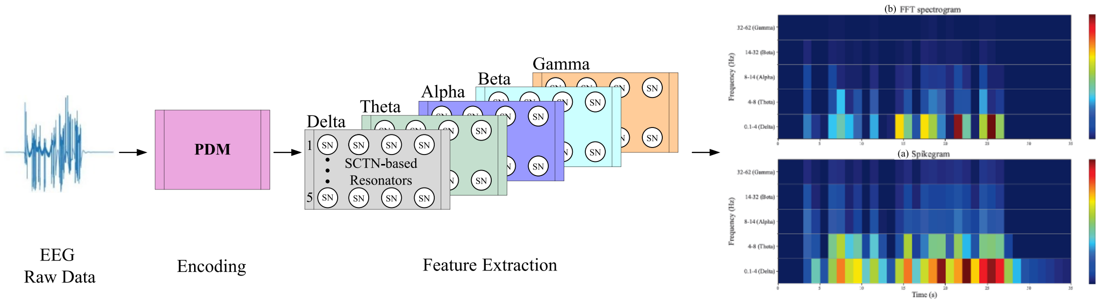

Figure 10. Extraction of EEG frequency features: The EEG signal is converted into spikes, followed by 5 SCTN-based resonators for each EEG subband. The extracted features are presented using (a) the Spikegram method compared to the (b) FFT spectrogram.

(4–8 Hz), Alpha (8–14 Hz), Beta (14–32 Hz), and Gamma (32–60 Hz). Each of the five EEG subbands is further divided into five specific frequencies to cover the whole frequency range.

To assess the quality of the SCTN-based features, we utilize them to classify EEG-based situation awareness into three categories. The performance evaluation of the proposed approach has been carried out using the EEG dataset described in [42]. This dataset includes 25 h EEG recordings collected from five participants engaged in a low-intensity control task and has been used for classifying three mental states: focused, unfocused, and drowsing state [42]. Eighty percent of the dataset was used for training, while 20% was used for testing.

4.2. Simulation configuration and results

Figure 10 depicts the proposed feature extraction process applied to the EEG signal. First, the EEG signal is encoded into spikes using PDM. The LF and LP parameters of the four SCTN neurons, which compose the resonator, are determined according to equation (8) to represent specific EEG frequency within the delta range (0.1–4 Hz). Then, the SNN is trained to detect the desired frequency, adapting the weights (W_{ij}) and the biases (Θ_i) for each SCTN neuron. The learning process converges after 1000 epochs on average, where each training round contains a complete cycle of the sine signal with 40 sampling windows and 500-time samples in each window (i.e. a total of 20 K samples for a complete cycle of the signal).

Table 2. EEG Classification results.

	Accuracy [%]	
Classification Model	FFT	SCTN
Weighted Ensemble [43]	90.00	**94.34**
XGBoost [44]	88.80	**92.64**
CatBoost [45]	88.20	**92.39**
LightGBM [46]	85.50	**92.00**
Random Forest [47]	85.60	**90.86**
Neural Network [48]	86.20	**88.40**
KNN [49]	67.40	**72.00**

The frequency outputs of the resonators are represented using the Spikegram method, as depicted in figure 10(a), demonstrating the occurrence of spikes over time for the five subbands. The strength of the signal is shown in color, where red represents the maximum spike rate. Figure 10(b) depicts the spectrogram of the FFT of the same EEG signal under-test for the five subbands. A cross-correlation between the frequency features in the FFT domain and those extracted by the proposed SNN-based resonators shows a high correlation between the two feature sets, ranging from 0.8 to 0.93. This very interesting result leads to the ability to use the proposed SNN-based approach as a low-cost and efficient alternative to the well-known FFT transformation.

Table 2 shows the classification accuracy results using an open-source classification tool [50] introducing seven classification models. The performance of the classifier was evaluated for both the SCTN-based and the FFT features. Results show that EEG classification based on our proposed approach (SCTN) outperforms the FFT-based classification for all seven models and demonstrates an accuracy improvement of up to 6.5% (for the LightGBM classifier [46]). Results show that the *Weighted Ensemble* classifier [43] outperforms all other classification models and demonstrates an accuracy of 94.34%.

To further evaluate the proposed SCTN-based features, we compare our approach with the work presented in [42] using the same dataset. Results show an accuracy improvement of 2.6% for classifying the three mental states, demonstrating an accuracy of 94.34% with our features compared to 91.7% while using FFT features and the classical SVM method [42].

The efficiency of the proposed solution is also demonstrated both in terms of power consumption and area. The EEG use case shows that an array of 25 SNN-based resonators (five for each sub-band) is required to detect all of the EEG frequency components. Each resonator comprises four neurons; therefore, the SNN-based feature extraction network comprises only 100 SCTN neurons.

In a previous study, we demonstrated the low energy consumption of our proposed spike neuron. The SCTN is characterized with an average low energy consumption of 3.5 mW cm^{-2} [2]. Our simulations demonstrate a power consumption of a single filter amounts to 40 nW, while the overall power consumption of the SNN-based feature extraction circuit reaches 1 μW.

5. Conclusion

This work revisits time frequency signal analysis through a biologically inspired approach. The approach is based on a SNN composed of a SCTN model. Specifically, we propose an SNN-based resonator for accurate frequency detection and extracting features in the frequency domain.

A new supervised STDP learning approach demonstrates a fast and efficient convergence. The quality of the extracted features has been examined for classifying EEG signals, demonstrating high classification accuracy. A comparison to frequency features in the FFT domain shows a high correlation with the proposed SNN-based extracted features.

Experimental results show the ability of the proposed architecture to perform a time–frequency analysis of a signal (specifically an EEG signal) similar to an FFT spectrogram but with a lower cost. This interesting result leads to the ability to use the proposed SNN-based approach as a low-cost and efficient alternative to the well-known FFT transformation. The proposed approach can be effectively applied to various applications to analyze time series and detect frequencies in a wide range while applying diverse frequency filters.

The preprocessing stage is typically executed by a dedicated electrical circuitry comprising analog-to-digital converters. This stage involves segmenting the signal into segments with overlap, converting the signal from the time domain to the frequency domain, and extracting features in the frequency domain [51, 52]. The proposed method allows for a direct connection between the SCTN-based network and time series-based biomedical sensors, eliminating the common need for signal preprocessing. Additionally, spike neurons are characterized by continuous processing over time, so segmenting the input signal into discrete frames is unnecessary.

This study describes the potential usage of SNNs as spectral analyzers and a competitive tool against the usual classic time–frequency analysis tools. Future work can expand the proposed method to include a neuromorphic SCTN-based classification network, suggesting a complete end-to-end bio-inspired neural network for encoding purposes, feature extraction, and classification.

Data availability statement

The source code and dataset can be found in the following https://github.com/NeuromorphicLabBGU/Time-Frequency-Analysis-Using-Spiking-Neural-Network. No new data were created or analysed in this study.

Conflict of interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

ORCID iDs

Moshe Bensimon https://orcid.org/0000-0002-9291-1703
Shlomo Greenberg https://orcid.org/0000-0002-1385-8394

References

[1] Schuman C D, Kulkarni S R, Parsa M, Mitchell J P, Date P and Kay B 2022 Opportunities for neuromorphic computing algorithms and applications *Nat. Comput. Sci.* **2** 10–19
[2] Bensimon M, Greenberg S, Ben-Shimol Y and Haiut M 2021 A new sctn digital low power spiking neuron *IEEE Trans. Circuits Syst. II* **68** 2937–41
[3] Fortuna L and Buscarino A 2023 Spiking neuron mathematical models: a compact overview *Bioengineering* **10** 174
[4] Abbott L F 1999 Lapicque's introduction of the integrate-and-fire model neuron (1907) *Brain Res. Bull.* **50** 303–4
[5] Steriade M 2004 Neocortical cell classes are flexible entities *Nat. Rev. Neurosci.* **5** 121–34
[6] Gerstner W and Kistler W M 2002 *Spiking Neuron Models: Single Neurons, Populations, Plasticity* (Cambridge university press)
[7] Izhikevich E M 2004 Which model to use for cortical spiking neurons? *IEEE Trans. Neural Netw.* **15** 1063–70
[8] Izhikevich E M and Edelman G M 2008 Large-scale model of mammalian thalamocortical systems *Proc. Nat Acad Sci* vol 105 pp 3593–8
[9] Hodgkin A L and Huxley A F 1952 A quantitative description of membrane current and its application to conduction and excitation in nerve *J. Physiol.* **117** 500
[10] Nunes J D, Carvalho M, Carneiro D and Cardoso J S 2022 Spiking neural networks: a survey *IEEE Access* **10** 60738–60764
[11] Pietrzak P, Szczesny S, Huderek D and Przyborowski Ł 2023 Overview of spiking neural network learning approaches and their computational complexities *Sensors* **23** 3037
[12] Hebb D O 2005 *The Organization of Behavior: A Neuropsychological Theory* (Psychology press)
[13] Vigneron A and Martinet J 2020 A critical survey of stdp in spiking neural networks for pattern recognition *2020 Int. Joint Conf. on Neural Networks (IJCNN)* (IEEE) pp 1–9
[14] Hussain I and Thounaojam D M 2021 An extensive review of the supervised learning algorithms for spiking neural networks *Int. Conf. on Big Data, Machine Learning and Applications* (Springer) pp 63–80
[15] Wang X, Lin X and Dang X 2020 Supervised learning in spiking neural networks: a review of algorithms and evaluations *Neural Netw.* **125** 258–80
[16] Agebure M A, Wumnaya P A and Baagyere E Y 2021 A survey of supervised learning models for spiking neural network *Networks* **5** 35–49
[17] Legenstein R, Pecevski D and Maass W 2008 A learning theory for reward-modulated spike-timing-dependent plasticity with application to biofeedback *PLoS Comput. Biol.* **4** e1000180
[18] Caporale N and Dan Y 2008 Spike timing–dependent plasticity: a hebbian learning rule *Annu. Rev. Neurosci.* **31** 25–46
[19] Guo W, Lin X and Yang X 2021 A supervised learning algorithm for recurrent spiking neural networks based on bp-stdp *Neural Information Processing: 28th Int. Conf. ICONIP 2021* (*Sanur, Bali, Indonesia, 8–12 December 2021*) (*Proc. Part V*) vol *28* (Springer) pp 583–90
[20] Guo W, Fouda M E, Eltawil A M and Salama K N 2023 Efficient training of spiking neural networks with temporally-truncated local backpropagation through time *Front. Neurosci.* **17** 1047008
[21] Hu Z, Wang T and Hu X 2017 An stdp-based supervised learning algorithm for spiking neural networks *Neural Information Processing: 24th Int. Conf., ICONIP 2017* (*Guangzhou, China, 14–18 November 2017*) (*Proc. Part II*) vol *24* (Springer) pp 92–100
[22] Stiefel K M and Ermentrout G B 2016 Neurons as oscillators *J. Neurophysiol.* **116** 2950–60
[23] Wu L, Wang Z, Bao L, Yu Z, Chen Q, Ling Y, Qin Y, Bao S, Chen Z and Bai G *et al* 2022 Implementation of neuronal intrinsic plasticity by oscillatory device in spiking neural network *IEEE Trans. Electron Devices* **69** 1830–4
[24] Kashchenko S, Mayorov V and Mayorova N 2023 Analysis of oscillating processes in spiking neural networks *Eur. Phys. J. Spec. Top.* **232** 509–27
[25] Izhikevich E M 1998 Phase models with explicit time delays *Phys. Rev.* E **58** 905
[26] Lakhmitskii N, Chizhevskii V and Kilin S Y 2023 Stochastic resonance in optoelectronic artificial spiking neuron *J. Appl. Spectrosc.* **90** 1069–73

[27] Zhang L 2017 Real-time feature extraction for multi-channel EEG signals time-frequency analysis *2017 8th Int. IEEE/EMBS Conf. on Neural Engineering (NER)* (IEEE) pp 493–6

[28] Boashash B 2015 *Time-Frequency Signal Analysis and Processing: a Comprehensive Reference* (Academic)

[29] Adeli M, Rouat J, Wood S, Molotchnikoff S and Plourde E 2016 A flexible bio-inspired hierarchical model for analyzing musical timbre *IEEE/ACM Trans. Audio Speech Lang. Process.* **24** 875–89

[30] Salau A O and Jain S 2019 Feature extraction: a survey of the types, techniques, applications *2019 Int. Conf. on Signal Processing and Communication (ICSC)* (IEEE) pp 158–64

[31] Mistry Y D, Birajdar G K and Khodke A M 2023 Time-frequency visual representation and texture features for audio applications: a comprehensive review, recent trends and challenges *Multimedia Tools Appl.* **82** 36143–77

[32] Kataria P, Sharma T and Narayan Y 2022 A review of time, frequency and hybrid domain features in pattern recognition techniques *Emerging Technologies in Data Mining and Information Security: Proc. IEMIS 2022* vol2 pp 411–22

[33] Yang Y, Peng Z, Zhang W and Meng G 2019 Parameterised time-frequency analysis methods and their engineering applications: a review of recent advances *Mech. Syst. Signal Process.* **119** 182–221

[34] Qing-Hua W, Li-Na W and Song X 2020 Classification of EEG signals based on time-frequency analysis and spiking neural network *2020 IEEE Int. Conf. on Signal Processing, Communications and Computing (ICSPCC)* (IEEE) pp 1–5

[35] Sejdić E, Orović I and Stanković S 2018 Compressive sensing meets time–frequency: An overview of recent advances in time–frequency processing of sparse signals *Digit. Signal Process.* **77** 22–35

[36] Burelo K, Ramantani G, Indiveri G and Sarnthein J 2022 A neuromorphic spiking neural network detects epileptic high frequency oscillations in the scalp EEG *Sci. Rep.* **12** 1798

[37] Costa F, Schaft E V, Huiskamp G, Aarnoutse E J, van't Klooster M A, Krayenbühl N, Ramantani G, Zijlmans M, Indiveri G and Sarnthein J 2024 Robust compression and detection of epileptiform patterns in ecog using a real-time spiking neural network hardware framework *Nat. Commun.* **15** 3255

[38] Bensimon M, Greenberg S and Haiut M 2021 Using a low-power spiking continuous time neuron (sctn) for sound signal processing *Sensors* **21** 1065

[39] Gerstner W, Kistler W M, Naud R and Paninski L 2014 *Neuronal Dynamics: From Single Neurons to Networks and Models of Cognition* (Cambridge University Press)

[40] Song S, Miller K D and Abbott L F 2000 Competitive hebbian learning through spike-timing-dependent synaptic plasticity *Nat. Neurosci.* **3** 919–26

[41] Thiele J C, Bichler O and Dupret A 2018 Event-based, timescale invariant unsupervised online deep learning with stdp *Front. Comput. Neurosci.* **12** 46

[42] Acı, Kaya M and Mishchenko Y 2019 Distinguishing mental attention states of humans via an EEG-based passive bci using machine learning methods *Expert Syst. Appl.* **134** 153–66

[43] Mahendran N, Vincent D R, Srinivasan K, Chang C-Y, Garg A, Gao L and Reina D G 2019 Sensor-assisted weighted average ensemble model for detecting major depressive disorder *Sensors* **19** 4822

[44] Chen T and Guestrin C 2016 Xgboost: A scalable tree boosting system *Proc. 22nd ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining* pp 785–94

[45] Prokhorenkova L, Gusev G, Vorobev A, Dorogush A V and Gulin A 2018 Catboost: unbiased boosting with categorical features *Advances in Neural Information Processing Systems* vol 31

[46] Ke G, Meng Q, Finley T, Wang T, Chen W, Ma W, Ye Q and Liu T-Y 2017 Lightgbm: A highly efficient gradient boosting decision tree *Advances in Neural Information Processing Systems* vol 30

[47] Biau G and Scornet E 2016 A random forest guided tour *Test* **25** 197–227

[48] Fakoor R, Mueller J W, Erickson N, Chaudhari P and Smola A J 2020 Fast, accurate and simple models for tabular data via augmented distillation *Advances in Neural Information Processing Systems* vol 33, ed H Larochelle, M Ranzato, R Hadsell, M Balcan and H Lin (Curran Associates, Inc) pp 8671–81

[49] Peterson L E 2009 K-nearest neighbor *Scholarpedia* **4** 1883

[50] Erickson N, Mueller J, Shirkov A, Zhang H, Larroy P, Li M, and Smola A, 2020 Autogluon-tabular: Robust and accurate automl for structured data (arXiv:2003.06505)

[51] Wang J and Wang M 2021 Review of the emotional feature extraction and classification using EEG signals *Cog. Robot.* **1** 29–40

[52] Wei Y, Zhou J, Wang Y, Liu Y, Liu Q, Luo J, Wang C, Ren F and Huang L 2020 A review of algorithm & hardware design for ai-based biomedical applications *IEEE Trans. Biomed. Circuits Syst.* **14** 145–63