IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 30, 2021, accepted July 11, 2021, date of publication July 26, 2021, date of current version August 30, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3099492

Incorporating Structural Plasticity Approaches in Spiking Neural Networks for EEG Modelling

MAHIMA MILINDA ALWIS WEERASINGHE[1], JOSAFATH I. ESPINOSA-RAMOS[1], GRACE Y. WANG[2], AND DAVE PARRY[1]

[1]Department of Computer Science, Auckland University of Technology, Auckland 1010, New Zealand
[2]Department of Psychology and Neuroscience, Auckland University of Technology, Auckland 0627, New Zealand

Corresponding author: Dave Parry (dave.parry@aut.ac.nz)

ABSTRACT Structural Plasticity (SP) in the brain is a process that allows structural neuronal changes, in response to learning. Spiking Neural Networks (SNN) are an emerging form of artificial neural networks that use brain-inspired techniques to learn. However, the application of SP in SNNs, its impact on overall learning, and network behaviour is rarely explored. In the present study, we use an SNN with a single hidden layer, to apply SP in classifying Electroencephalography (EEG) signals of two publicly available datasets. We considered classification accuracy as the learning capability and applied metaheuristics to derive the optimised number of neurons for the hidden layer along with other hyperparameters of the network. The optimised structure was then compared with overgrown and undergrown structures to compare the accuracy, stability, and behaviour of the network properties. Networks with SP yielded ∼94% and ∼92% accuracies in classifying wrist positions and mental states(stressed vs relaxed) respectively. The same SNN developed for mental stress classification produced ∼77% and ∼73% accuracies in classifying arousal and valence states. Moreover, the networks with SP demonstrated superior performance stability during iterative random initiations. Interestingly, these networks had a smaller number of inactive neurons and a preference for lowered neuron firing thresholds. This research highlights the importance of systematically selecting the hidden layer neurons over arbitrary settings, particularly for SNNs using Spike Time Dependent Plasticity learning and provides potential findings that may lead to the development of SP learning algorithms for SNNs.

INDEX TERMS Artificial neural networks, structural plasticity, electroencephalography, evolutionary computation.

I. INTRODUCTION

Spiking Neural Networks (SNNs) referred to as the third generation of neural networks [1], are capable of accommodating pattern recognition and function approximations with greater computational efficiency [2]. This can be credited to the approach of adopting biologically inspired information processing techniques in SNNs. Although this approach does not guarantee better accuracy in pattern recognition tasks, it is more likely to give detailed insight into brain-like computing [3], such as event-based spatiotemporal data processing.

However, the advantages of SNNs cannot be fully utilized in Machine Learning(ML) tasks due to the lack of robust and well-justified learning techniques [2]–[4], which inhibits accuracy. Therefore, it is a common practice to make SNNs larger and deeper [5], [6]. Additionally, SNNs operate with many hyperparameters, which can make network implementation and optimisation more computationally expensive. To resolve these issues, it is important to better understand the unified function of neurons, synapses and hyperparameters in accurately recognizing patterns.

The remarkable pattern recognition capability of the mammalian neocortex is achieved with a power consumption of 10 to 20 Watts [7]. One key attribute that accommodates this phenomenon is structural plasticity (SP). This is the self-regulation capability of neuronal circuits which was initially investigated by [8]. Moreover, neuroscience literature [9]–[12] further highlights the positive impact of SP

The associate editor coordinating the review of this manuscript and approving it for publication was Thomas Canhao Xu.

on learning. Though brain-inspired learning concepts such as spike time dependent plasticity(STDP) [13], [14] and its variants are being applied for training SNNs, the working of SP in an STDP learning environment is rarely investigated. Studies exploring the interplay between SP and computational properties of neurons and synapses such as firing thresholds are even rarer.

In this paper, we explore the impact of systematically varying the number of neurons in the hidden layer (denoted as η) based on prior learning, on ML performance. Furthermore, we investigate the role of hyperparameters (i.e., intrinsic properties of neurons and synapses) under this process to provide better insights to further expand the knowledge in the area. For this purpose, we use metaheuristics with SNNs. To the best of our knowledge, this is the first time where the impact of η on hyperparameters and network performance is analysed quantitatively and qualitatively. Our research used a 3 layered SNN architecture and two publicly available Electroencephalographic (EEG) datasets [15], [16].

This paper is organized as follows. Section II presents a summary of research in the application of SP in SNNs. Section III introduces the two EEG datasets, annotation and encoding procedures. Section IV describes the SNN setup and learning procedures involved. In Section V, we introduce the structural optimisation techniques and experimental framework. Section VI presents results and observations which is discussed and concluded in Sections VII and VIII, respectively.

II. STRUCTURAL PLASTICITY (SP) IN SNNs

In this section, we summarise SP methods found in SNN literature and, highlight the research gap. Here, we commonly refer to all structural adaptation methods in SNN literature as SP. This includes both growth and pruning of neurons and/or synapses. When we consider SP in SNNs, all research in the area can be divided into two main categories. Use of SP as a learning mechanism introduced in [17]–[20] or as a method to increase efficiency [21], [22].

In previous research [17], a four-layered SNN was introduced with 2D maps of integrate and fire neurons, specifically designed for visual recognition tasks. The first two layers filter incoming samples according to contrast and orientation. The third layer of this SNN learns using rank order(RO) rule [23]. Each time a sample is presented to the network, a new neuronal map is created in the third layer. Similarity of the newly created map with already exciting maps is calculated by applying inverse Euclidean Distance measure between weight matrices. If the similarity exceeds a predefined similarity threshold value, maps are merged and, if not, maintained separately. For each of the maps in the third layer, a single neuron is maintained at the fourth layer. The synapse in this layer responds only to excitatory signals where weight is increased by $+1$ for each spike arrival. During inferencing the class specific neuron at the fourth layer spikes.

Another SP algorithm is introduced in [18], a three layered SNN with population coding introduced in the first layer with multiple delays. The hidden layer consists of RBF neurons where the sample representative capability of a given input is determined by the time taken to produce a spike. Therefore, the neuron that spikes first in the hidden layer is considered as the winner neuron. The time taken by the winner to fire, is compared with a predefined threshold to decide the new neuron addition (i.e., existing neurons is not representative enough). Similarly, if two winner neurons fire with a certain time gap that is below a predefined pruning threshold, one neuron is removed (i.e., two neurons demonstrate similar representative capability). In the case of neuron addition, scaled weights are assigned to the newly added neuron, making it fire first for the current sample. STDP and anti-STDP rules are used to update the weights between hidden and output layers. Rate coding is used at the output layer to determine the class of a given input.

In [19], two-layered SNN is introduced with SP applied at the output layer. The input layer of the network is used for population coding and the output neurons are leaky integrate and fire (LIF) neurons. The weights between the layers are updated using the modified RO rule. The class of a given input signal is determined by the neuron that produces a spike earliest at the output layer. When a new sample is propagated, if the timestamps of the first spikes produced by output neurons exceed a predefined threshold, a neuron is added and, if it is below the threshold, the sample is skipped. The time gap between the predefined threshold and the actual spike time for the current sample determines whether a new neuron should be added, sample to be skipped or weights should be updated.

An online learning mechanism based on adaptable connectivity is introduced in [20]. In this study, a winner-takes-all network is implemented with a single layer of LIF neurons with multiple synapses. A separate connection matrix is maintained which keeps a track of the connections to synapses and input streams. The efficacy of a certain connection to make a neuron spike is being calculated using STDP inspired learning algorithm. This value of spike efficacy is then compared with a dynamic threshold value to allow the elimination of connections after each training epoch. Therefore, apart from weight learning, synaptic rewiring takes place.

Pruning is another form of SP applied mainly to increase efficiency, often at the cost of learning capability. In [21], researchers introduce a synaptic pruning method based on STDP weights. There, a pruning threshold is defined and STDP weights below the threshold values are pruned. In the same research, the applicability of weight quantization is introduced which restricts all the values in the trained weight matrix to a predefined scale. Another form of synapse elimination is Soft-Pruning [22], where synapses selected to prune are set to their lowest value instead of completely removing during training epochs. The total elimination of synapses is made once training is completed. Both methods have proven

to have the capability of reducing the network parameters drastically while maintaining accuracy performance up to a certain extent.

As discussed [21], [22] are methods where SP can be implemented to increase resource efficiency which is particularly important in hardware implementations of SNNs. In contrast, [17]–[20] introduces SP as a form of learning method where neurons are added and/or removed with predefined thresholds to obtain desired spiking patterns. However, these methods are not intended to explore the impact of η under STDP learning or the impact of the same on overall ML performance and intrinsic properties of the network.

III. DATA USED IN THE STUDY

SNNs process and communicate information with sparse and asynchronous binary signals called 'spikes' [4]. This method of operation combined with unsupervised learning methods such as STDP makes SNNs an ideal solution for exploring the dynamics of spatiotemporal data [2], [3], [24]. EEG is one such data type with properties of autocorrelation and heterogeneity built into the temporal signal. In contrast to studies [25], [26] using manual feature extractions, we have used an SNN architecture with STDP learning for automated spatiotemporal feature recognition for pattern classification.

A. EEG DATASETS

1) DATASET 1- WRIST FLEXION DATASET

This dataset is from a study conducted to test the feasibility of an SNN architecture in detecting motor execution and motor intention [15]. EEG was collected from 3 healthy participants when performing 3 different wrist positions, namely flexion, extension, or rest. EEG was recorded using 14 channel Emotive Neuroheadset with international 10-20 locations and each recording lasted for 20 seconds. Recordings were obtained under an eyes-closed state minimizing artefacts due to eye blinking and, no additional data cleaning techniques were utilised except for the data encoding. The recordings were sampled at 128 Hz without additional data preparation steps. (Dataset is available at https://github.com/KEDRI-AUT/neucube-cloud-sample-file/blob/master/wrist_movement_eeg.zip)

2) DATASET 2- DEAP DATASET

We used an extraction of DEAP dataset [16] to detect emotional stress. This dataset consisted EEG recordings of 32 healthy participants collected while watching 1-minute video clips, consequently annotated by each individual for valence and arousal using a self-assessment manikin [27]. Each recording was conducted using 32 channels at 512Hz. Our experiments used 32 channel data downsampled to 128Hz, filtered with a 4Hz to 45Hz bandpass filter, and EOG removed. (The main pre-processed dataset is publicly available at https://www.eecs.qmul.ac.uk/mmv/datasets/deap/download.html). We then normalised each sample to bring all values to a common scale as per Fig.1. This process was conducted on all samples prior to data encoding.

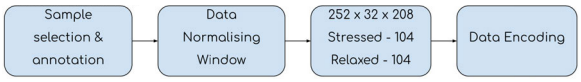


FIGURE 1. Data preparation process used for dataset 2.

B. DATA ANNOTATION

Firstly, we converted all samples of both datasets into comma-separated values (CSV) file format. Therefore, each sample file represented the format of an $m \times n$ matrix with rows representing time points and, the column representing EEG channels (i.e., Features/Attributes).

1) DATASET 1- WRIST FLEXION DATASET

A sample of this dataset is a 128 $\times$ 14 (i.e., time steps $\times$ channels) matrix. The dataset comprised 60 samples in total, with 20 samples corresponding to each of the wrist position. We annotated wrist flexion as class one, extension as class two, and resting state as class three.

2) DATASET 2- DEAP DATASET

The extraction from the main DEAP dataset used the following equations,

$$Stressed = (Arousal > 5) \cap (Valence < 3) \quad (1)$$

$$Relaxed = (Arousal < 4) \cap (4 < Valence < 6) \quad (2)$$

obtained from [28]. We assigned stressed samples as class one and relaxed as class two. A sample of the selected dataset is in the form of 252 $\times$ 32 (i.e., time steps $\times$ channels) matrix. A total of 208 samples were obtained using the above equations with 104 samples for each class. This same data extraction method with DEAP dataset was carried out by [29], [30] and [31] which enables performance comparison.

C. DATA ENCODING

Before feeding the data into the SNN, a spike conversion takes place. For this, we used the Address Event Representation (AER) [32] for both datasets considering the representation and noise filtering capability. Equation (3) provides the threshold calculation, where $M(dif)$ and $Std\ (dif)$ denotes the median and standard deviation of temporal difference signal dif. A dif of a particular EEG channel is calculated by subtracting amplitude at time t by the amplitude at $t-1$. The threshold factor f is set by the user (In our experiments this was set to 0.5). Once Tr is calculated according to (3), dif amplitudes at each time point are compared. If the dif amplitude exceeds Tr at time t, an excitatory spike is emitted and if the dif amplitude drops below - Tr, an inhibitory spike is emitted.

$$Tr = M(dif) + f * Std\ (dif) \quad (3)$$

As shown in Fig.2 where peaks of the original signal are represented with a volley of spikes, this encoding algorithm

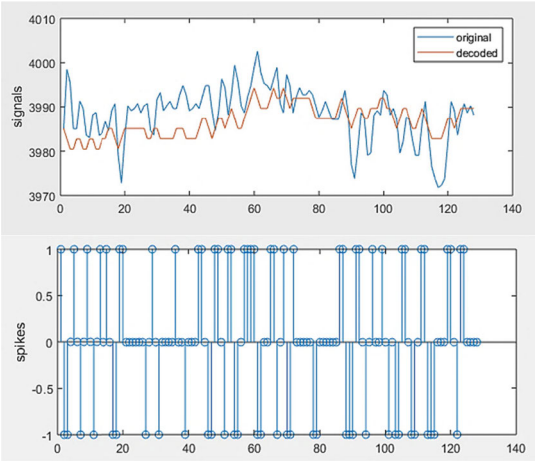


FIGURE 2. Spike representation of FP1 channel extracted from a stressed class sample. The top plot illustrates the original signal (Blue) and reconstruction of the same using spikes generated (Red).

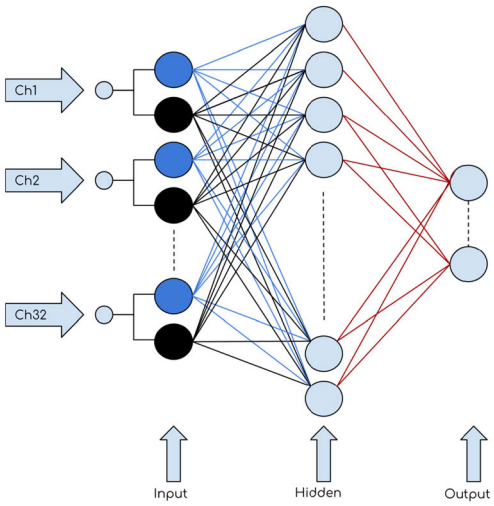


FIGURE 3. SNN architecture used for classification task using dataset 2 accommodating 32 inputs.

gives prominence to amplitude fluctuations, enabling preservation of salient events in the EEG signal. This method can filter out minute fluctuations caused by signal noise.

IV. SNN ARCHITECTURE

In our experiments, we used a 3-layered SNN with leaky integrate and fire (LIF) neurons, developed based on the JNeuCube framework, publicly available at https://github.com/Auckland-University-of-Technology/NeuCube-java.

The input layer consisted of pairs of nodes capable of accommodating both excitatory and inhibitory spikes. The number of pairs depended on the number of EEG channels. (i.e., for Dataset 1, 14 pairs and dataset 2, 32 pairs). We fully connected the input to the hidden layer with pseudo-random weight initiations following the gaussian distribution and an unsupervised learning strategy based on spike time dependency(STDP) algorithm [13], [14] to update weights. The hidden to output layer also followed fully connected criteria. The weight update strategy was inspired by deSNN [33] algorithm.

A. LEAKY INTEGRATE AND FIRE NEURONS

For the neuron model selection, we focused on using a simple biological representation with tractability. Therefore, we used LIF model described in [34]. The behaviour of a LIF is modelled using a resistor-capacitor circuit adhering to the following equations,

$$\frac{d}{dt}(\upsilon) = \left[RI_{(t)} - (\upsilon_t - \upsilon_{rest})\right]/\tau_m \tag{4}$$

where υ denotes membrane potential, R resistance, $I_{(t)}$ for instantaneous current, υ_{rest} for resting voltage and τ_m the membrane time constant, which is calculated as,

$$\tau_m = RC \tag{5}$$

and when membrane potential υ exceeds a defined threshold υ_{thresh}, a spike denoted as s^{ti} is generated i.e.,

$$s^{ti} : \upsilon_{(ti)} \geq \upsilon_{thresh} \tag{6}$$

B. SPIKE TIME DEPENDENT PLASTICITY

STDP is a temporally asymmetric form of Hebbian learning which depends on the spiking time between pre and post-synaptic neurons to adjust the weight of the synapse [14]. Therefore, STDP is a local learning rule i.e., it does not consider the information of other synapses for weight update. The significance here is that STDP enables neurons to discriminate temporally distinct inputs and then integrate them to form a meaningful output [35]. Hence by using STDP in a fully connected network, we ought to capture the patterns between spatially distributed yet temporally synchronized spiking activity. One of the drawbacks in using STDP for learning is the runaway synaptic potentiation, i.e. the synapse getting caught up in a potentiation loop that increases weight even if the information coming into the synapse is insignificant [36]. For the experiments discussed in this paper, we have restricted the weight increase by defining boundaries ($+0.1$ and -0.1) and we have introduced inhibitory nodes in the input layer to balance the spiking activity.

The formalized mathematical model for STDP can be interpreted using (7) adapted from [37]. Depending on the coincidence between spiking activity within the learning window (i.e. the time window considered for weight adjustments between the pre and the post-synaptic spike), the long-term potentiation (LTP) i.e. synaptic strengthening by weight increasing or long-term depreciation(LTD) i.e. synaptic weakening by weight decreasing takes place according to (8) and (9) [13].

$$\Delta w_{ij} = \sum_k^y \sum_m^z F(t_i^m - t_j^k) \tag{7}$$

$$F_{(\Delta t)} = A_+ e^{(-\Delta t/\tau pos)} \quad \Delta t > 0 \tag{8}$$

$$F_{(\Delta t)} = -A_- e^{(\Delta t/\tau neg)} \quad \Delta t < 0 \tag{9}$$

As per (7), the weight adjustment Δw_{ij} is an accumulation of weight fluctuations calculated using function F over pre-synaptic spikes occurring from k to y and post-synaptic spikes occurring from m to z, where t_i^m represents post-synaptic spiking times and t_j^k represents pre-synaptic spiking times. In (8), Δt is the time gap between the two spikes and A_+ represents positive modification factor and τpos denotes the learning window for positive weight modifications. Vice versa, equation (9) denotes the negative weight modification with relevant modification factor A_-, and learning window, τneg. During our experiments, Δt was held equal and constant at 10 ms for excitation and inhibition, according to previous experimentations [14]. It is also understood that inequality of the same would create biases causing misrepresentations. Moreover, A_+ and A_- were selected as parameters to be tuned with a starting value of 0.001. The selection of these values depended on the simulations conducted with the encoded data.

C. CLASSIFIER LEARNING

We used a version of the Dynamic Evolving Spiking Neural Network (deSNN) [33] algorithm to create and adapt weights from hidden to output layer. This segment of the network considers the global activity of the hidden layer for learning and inferencing. Therefore, with STDP, it is a combination of local and global learning, a contrast to methods using only RO learning (i.e. global learning only) [17], [19].

The output layer (i.e., the sample classifying layer) is a setup that increases its number of neurons with each passing sample. Therefore, during the training cycle, each sample will be represented with a neuron at the output. This neuron is fully connected with the hidden layer. Initial weights of the hidden to output layer were established according to rank order rule [23] as per (10),

$$w_{ho} = \alpha.mod^{order(h,o)} \tag{10}$$

where w_{ho} represents synaptic weight between hidden and output neurons. α is the learning parameter; the constant factor which decides the magnitude of the weight. For these experiments, we have set this to 1 based on previous simulation experience. The modulation factor mod takes the values between 0 and 1. Each synapse coming from the hidden layer is set with an initial weight based on the spike arrival order $order(h, o)$. Here, mod is set to 0.8, and the corresponding neuron synapse will get a weight of 1 since the $order(h, o)$ would be set to 0. The spiking proceeded will receive weights from $order(h, o)$ increased 0 onwards.

Once the initial weight of each synapse is set, using two parameters namely *positive drift* and *negative drift*, weights are updated according to the spike arrival at each time step. The values of these parameters were tuned. If a spike arrived at a given time step, the new weight was set to $w_{ho}+$ *positive drift*, or $w_{ho}-$ *negative drift* otherwise.

At the beginning of testing, the output layer consists of N number of neurons, each representing a training sample with the label being known. During testing, a sample is propagated, a new neuron is created, and weights evolved at each time step. Thereafter, the weight vector of that sample is used to calculate the Euclidean distance. According to the Euclidean distance, the model selects the closest training sample neuron and predicts by assigning the same label to the testing sample.

Considering the uniqueness of each participant's EEG data, we used this technique for the classification layer without neuron clustering in contrast to methods in [17], [19], and [33].

TABLE 1. Hyperparameters of the experiment.

Step	Hyperparameter	Range Tuned or Held Constant
AER Encoding	Spike Threshold	0.5
LIF	Reset Voltage	0
	Resistance	1
	Capacitance	10
	Voltage Threshold*	[0.01 - 0.5]
	Refractory period*	[2 – 10]
Local Learning	STDP Positive Modification Time Window	10 ms
	STDP Negative Modification Time Window	10 ms
	Random weight initiation boundaries	[-0.1, 0.1]
	Maximum weight bounds	[-0.1, 0.1]
	Positive Synaptic Modification*	[0.001 – 0.05]
	Negative Synaptic Modification*	[0.001 – 0.05]
Global Learning	RO learning parameter	1
	RO modulation factor	0.8
	Positive Drift*	[0.001 – 0.05]
	Negative Drift*	[0.001 – 0.05]

Describes all hyperparameters used in the SNN algorithm. The hyperparameters used for tuning are noted with *, with search range.

V. NETWORK OPTIMISATION

When it comes to SP, it is not justifiable to only change η and evaluate the performance of the network since the 6 hyperparameters mentioned in Table 1 directly influence spiking activity and classification accuracy. Therefore, each network with a certain η requires appropriate hyperparameter values. This section describes the process of tuning the hyperparameters.

A. DIFFERENTIAL EVOLUTION

We used Differential Evolution (DE) (i.e., DE/rand/2/bin version) as the optimisation algorithm to search for the most suitable parameters/hyperparameters that would produce the highest fitness. Amongst the advantages of DE, the capability to handle non-differentiable cost functions with minimum control variables, computational efficiency achieved through the independent generation of populations using stochastic perturbation and good convergence properties that have been experimentally proven were most important [38].

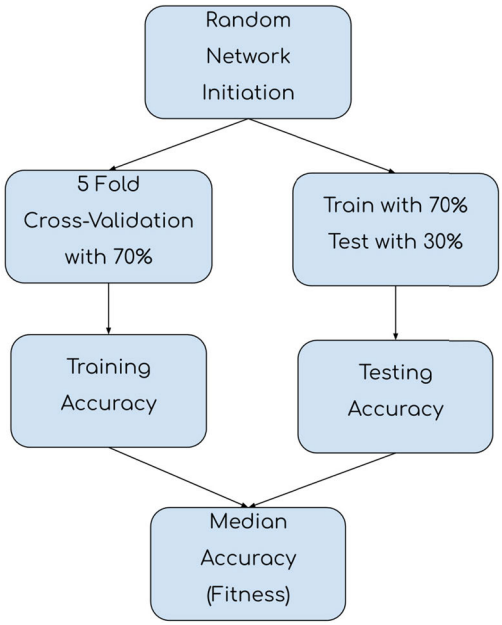


FIGURE 4. The fitness function of a single network initiation.

In DE, the greedy criterion is used to determine the acceptance of a given vector solution and an annealing process is introduced at trial vector generation to counter the search process getting trapped in a local optimum. A selected target vector solution performance is tested with a trial vector solution developed based on a mutation and crossover. The two variables of the algorithm are the weight factor used to amplify the differential variation in mutation and the crossover constant that determines the probability of the component representation from the mutant vector. In these experiments, we set the weight factor at 0.1 and constant for crossover probability at 0.8 according to the heuristic findings of [38].

B. EXPERIMENT FRAMEWORK

A single trial solution is tested on 10 pseudo-random network initiations. For each network initiation, data samples were selected randomly based on a 0.7 split, 70% of samples for training and 30% for testing. Each network initiation was validated for classification accuracy with 5-fold cross-validation and a testing round with unseen data. The fitness function of a given trial solution was obtained as the median of average accuracies of cross-validation and testing. (i.e. *(average cross-validation accuracy over 10 initiations + average testing accuracy over 10 initiations)/2*). Termination criteria were set to 100 iterations or maximization of the fitness function. Each iteration consisted of 50 agent solutions. Enabling validation of a maximum of 5000 candidate solutions.

1) EXPERIMENT 1 – OPTIMISING HIDDEN LAYER NEURON COUNT AND HYPERPARAMETERS

In this experiment, we optimised η and the 6 hyperparameters mentioned in Table 1. We selected 30 to 200 as the search space for η. This range was selected following random performance testing conducted. We conducted the same experimental procedures to both dataset 1 and dataset 2. Fig.5 shows the sample candidate solution used for experiment1. After completing experiment 1, we obtained structurally optimised networks with corresponding hyperparameter values that produced the highest fitness. This method enables the application of SP based on the network performance where each iteration generates information on η and other network properties that would enable better performance. This information is used as feedback in regenerating candidate solutions.

LIF parameters		STDP weight factors		deSNN weight factors		Structural Plasticity
Threshold	Refractory time	Positive modification	Negative modification	Positive modification	Negative modification	Neuron count

FIGURE 5. Representation of a sample trial solution vector with 6 hyperparameters (given under headings highlighted in blue) and hidden layer neuron count (highlighted in red) which was only used in Exp. 1.

2) EXPERIMENT 2 – OPTIMISING COMPARATIVE NETWORKS

If we consider the structurally optimised network to have p number of neurons in the hidden layer, we created 2 additional networks, with 1.5 times of p and 0.5 times of p (i.e., A network with 50% more and 50% fewer hidden neurons). These two networks (referred to as overgrown and undergrown) were separately optimised using DE to find the best values for the 6 hyperparameters mentioned in Table 1. The search spaces and the number of candidate solutions remained the same. Therefore, we concluded Experiment 2, with 3 optimised networks, for each dataset. (Optimised parameter values for each of the network and dataset is appended as Table 7).

During the evaluation stage, we reinitiated a selected network 30 times with optimised parameters and pseudo-random weights that followed gaussian distribution. During each initiation, data was split randomly with 70% for training and 30% for testing. This allowed a statistical evaluation of the performance.

VI. RESULTS

A. TESTING NETWORK PERFORMANCES

1) DATASET 1-WRIST FLEXION DATASET

As shown in Fig.6, the SNN with 111 hidden neurons produced ∼94% average classification accuracy for the 3-class classification task. The standard deviation of the network performance was recorded at 0.048.

We compared this network performance with other optimised networks. Fig.7 shows the distribution of testing accuracy across the 3 networks. The network with 111 neurons produced a 7% better average accuracy than 56 neurons. Moreover, 111 neurons produced lesser performance variance comparatively. The average accuracy of 111 neurons and 167 neurons was almost similar. Each of the networks

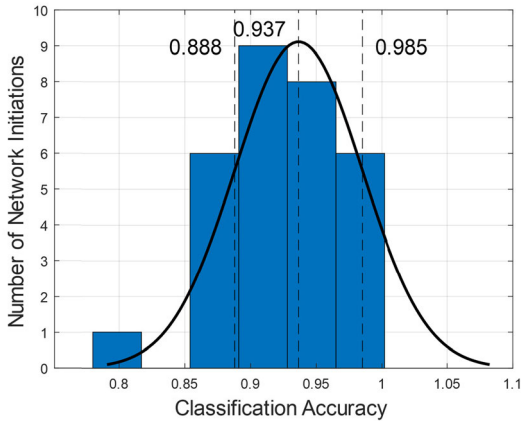


FIGURE 6. Statistical analysis of the performance for dataset 1 across 30 random initiations.

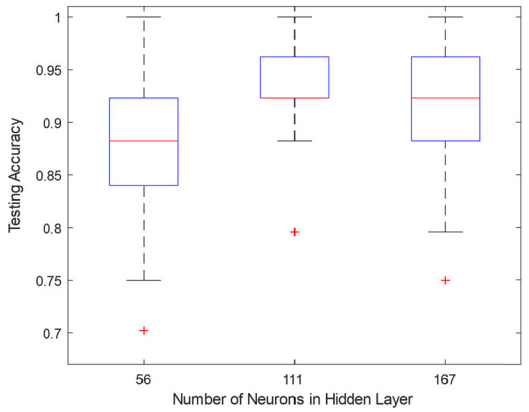


FIGURE 7. Comparative statistical analysis of the generalization capability of the 3 networks for dataset 1.

TABLE 2. Network performance comparison dataset 1.

Network (η)	Test accuracy	Cohen's Kappa	F1-score
56	0.865 ± 0.091	0.678 ± 0.129	0.817 ± 0.114
111	0.937 ± 0.048	0.819 ± 0.099	0.909 ± 0.068
167	0.922 ± 0.058	0.819 ± 0.122	0.889 ± 0.078

produced a single outlier performance with 111 neurons having the highest accuracy amongst them.

Moreover, we compared the performance of the 3 networks on Cohen's Kappa and F1s-score using the confusion matrices produced during iterative network testing. A statistical summary of the performance is given in Table 2. SNN with 111 neurons performing better than other SNNs in all measures.

Table 3 compares the performance of this study with previous studies that used the same dataset for classification. Please note that [15] used a 50:50 split for training and validation for the experiments with NeuCube architecture without hyperparameter optimisation.

TABLE 3. Dataset 1 classification accuracies.

Study	Method	Average Classification Accuracy
Taylor[15]	MLP	55%
	SVM	62%
	NeuCube architecture	76%
This Study	3-Layer SNN	93.7 ± 4.8 %

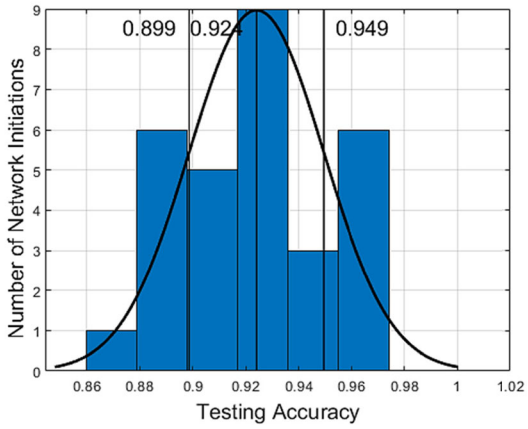


FIGURE 8. Statistical analysis of the performance for dataset 2 across 30 random initiations.

2) DATASET 2-DEAP DATASET

We obtained ∼92% average classification accuracy for 2-class classification tasks used to determine between Stressed and Relaxed brain states using the SNN with 130 hidden layer neurons. As shown in Fig.8, the Standard deviation of the network performance was recorded at 0.025.

Fig.8 shows the distribution of testing accuracy across the 3 networks. The network with 130 neurons produced 6% better average accuracy than 65 neurons and 1% better than 195 neurons. Moreover, 130 neurons produced comparatively smaller performance variance.

TABLE 4. Network performance comparison dataset 2.

Network (η)	Test accuracy	Cohen's Kappa	F1-score
65	0.860 ± 0.033	0.615 ± 0.076	0.8 ± 0.048
130	0.924 ± 0.025	0.823 ± 0.073	0.909 ± 0.039
195	0.910 ± 0.036	0.631 ± 0.086	0.796 ± 0.053

Table 4 presents the Cohen's Kappa and F1-score calculated over testing iterations where structurally optimised network indicated to have better performance in all measures.

Table 5 presents a comparison of the studies that used the same data previously for classification tasks with different machine learning techniques and feature extraction methods.

TABLE 5. Dataset 2 classification accuracies.

Study	Method	Average Classification Accuracy
Bastos[29]	K-NN with feature extraction	70.1%
Garcia[30]	SVM with feature extraction	81.31%
Shon[31]	K-NN with feature extraction	71.76%
This study	3-Layer SNN without feature extraction	92.4 ± 2.5 %

TABLE 6. Dataset 2 classification valence and arousal dimensions.

Method	Valence		Arousal	
	Accuracy	F1 Score	Accuracy	F1 Score
GNB[16]	57.6%	56.3%	62%	58.3%
DBN[40]	60.9%	-	51.20%	-
CNN/RNN [41]	72.06%	-	74.12%	-
DBN-SVM[41]	58.40%	-	64.20%	-
DNN[42]	75.78%	-	73.13%	-
CNN[42]	81.41%	-	73.36%	-
DBN-GC-based ensemble DNN[39]	76.83%	70.15%	75.92%	69.31%
This study	73.10 ± 1.5%	69.5 ± 1.8%	76.86 ± 1.3 %	70.92 ± 1.7%

We tested the same SNN with hyperparameters used for stress state classification, for emotion recognition based on valence and arousal dimensions with 10-fold cross validation to compare with the results summarised in [39]. Table 6 presents this performance comparison.

B. INACTIVE NEURONS AND LIF THRESHOLD

With the differences in performance observed, we analysed the average inactive neurons to the total number of hidden layer neurons ratios for the three networks for each of the dataset. We obtained the average inactive neuron count from the iterations used for statistical performance evaluation in part A. The results of this analysis are summarized in Fig.10 for both datasets. We observed a reduction in the ratio, for the optimised network structures for both datasets. In other words, in comparison to undergrown and overgrow networks, the network with the optimised number of neurons had a lesser number of inactive neurons compared to the total number of neurons.

In terms of the LIF threshold values, for 56, 111 and 167 η values, we recorded 0.081, 0.064 and 0.067 respectively for dataset 1. Similarly, for dataset 2, for 65, 130 and 195 η values, LIF threshold was recorded at 0.049, 0.013 and 0.059. This indicates a drop in LIF threshold for the structurally optimised network for both datasets. Fig.11 graphically represent this drop for dataset 2.

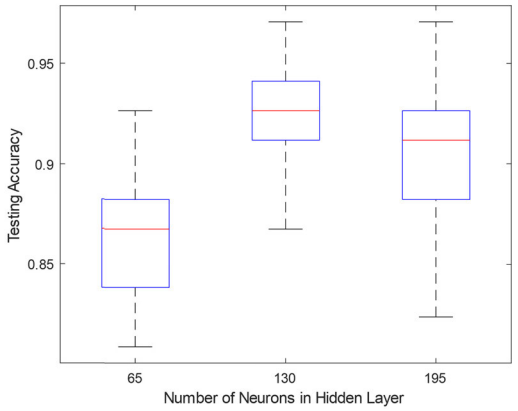


FIGURE 9. Comparative statistical analysis of the generalization capability of the 3 networks for dataset 2.

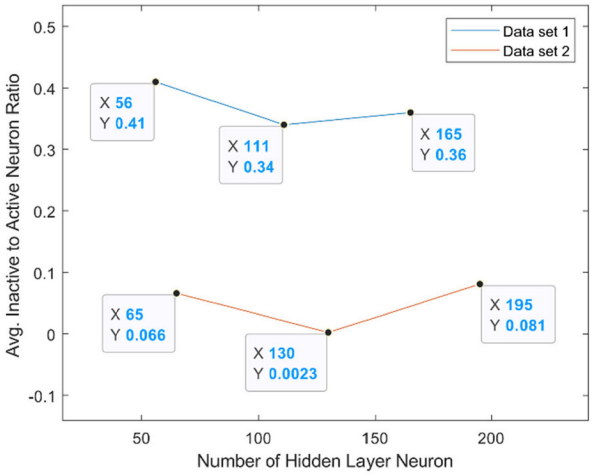


FIGURE 10. Graphical representation comparing inactive neurons to total neurons of different structures tested with dataset 1 (Blue) and dataset 2 (Red).

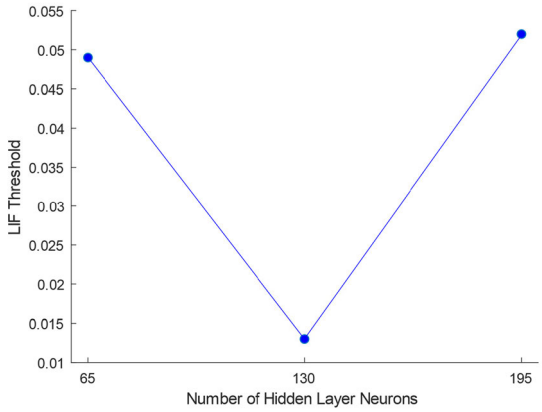


FIGURE 11. LIF Threshold selections for the SNNs with different η values for dataset 2.

VII. DISCUSSION

One of the key findings of this study is the level of performance and stability obtained using SP, reported in Table 2 and IV. For all performance measurements namely, accuracy, F1-Score and Kappa, network with SP produced comparatively higher mean and lower standard deviation. Therefore, in addition to having higher accuracy, the SP network performed well in terms of adaptability (i.e., performance stability under changing inputs and/or environments) [43]. This speculation was made with the randomization we introduced in, sample selection, weight initiations and input mapping, during network testing. This indicates the importance of considering SP in ML tasks, particularly when using STDP for learning, which aligns with neuroscientific findings [9]–[12]. Moreover, it is also clear that for SP to provide this performance, the corresponding hyperparameters needs to be optimised.

Investigating the better performance of SP revealed lowered LIF thresholds for both datasets in the optimised structure which results in higher probability for the neurons to fire. Fig.11 presents this finding for dataset 2. However, this spiking was sparser amongst neurons since the relative number of active neurons of the optimised networks were increased (i.e., inactive to total number of neurons decreased) as per Fig.10. This relates to most neurons having balanced firing rates, preventing selected set of neurons from over-activation which is desired in SNNs [5]. However, this desired state is not achieved in bigger networks presented in this paper. For an instance Fig.13 (appended), indicates firing activity of the trained network for dataset 1, where overgrown network with 167 neurons produced 5 over-activated neurons. These over-activated neurons can be attributed to, excessive exposure to similar patterns in training data and/or runaway synaptic potentiation [36] caused by STDP. Either case, this will have a negative impact on generalisation capability as observed. Since the classifier presented here is based on RO and spike counts, over-activation restrains pattern separation. This spiking domination can be potentially minimised by introducing self-regulation methods such as inhibitory neurons [12], lateral inhibition and homeostatic intrinsic plasticity [5] or reducing training simulation time [44]. As per the smaller networks presented, the level of activation is insufficient to capture the temporal dynamics in the data.

Furthermore, we have obtained the state-of-the-art classification accuracies of ∼94% and ∼92% for datasets 1 and 2, respectively. Moreover, the same SNN developed with SP for mental state recognition yielded 76.86% and 73.1% in classifying arousal and valence, respectively under 10-fold cross validation. As per Table 6, this result is comparable with contemporary deep learning methods tested using DEAP dataset. The performance of this method can be attributed to the capability of SNNs to learn from temporal spiking sequences, especially favouring data that are spatiotemporal by nature [2], [3], [24]. Responsiveness of STDP learning in recognising temporal patterns in the data automatically [45]–[47] may further enhance the said capability.

TABLE 7. Optimised parameter values.

Step	Hyperparameter	Exp. 1	Exp. 2	
Dataset 1	Voltage Threshold	0.064	0.081	0.067
	Refractory period	5.5	5.6	6.1
	STDP Positive Synaptic Modification	0.0027	0.0015	0.0038
	STDP Negative Synaptic Modification	0.0029	0.0018	0.0041
	Positive Drift	0.0372	0.0005	0.0114
	Negative Drift	0.0257	0.0071	0.0129
	Number of neurons	111	56*	167*
Dataset 2	Voltage Threshold	0.013	0.049	0.052
	Refractory period	5	5	5
	STDP Positive Synaptic Modification	0.079	0.0009	0.059
	STDP Negative Synaptic Modification	0.001	0.001	0.001
	Positive Drift	0.040	1.05	0.025
	Negative Drift	0.001	0.001	0.001
	Number of neurons	130	65*	195 *

Presents optimised values of the hyperparameters used in the SNN algorithm. The hyperparameters with * indicate values set manually.

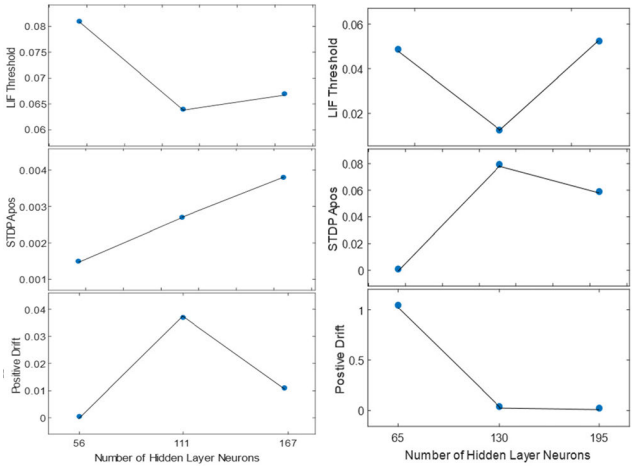


FIGURE 12. Hyperparameter (i.e., Neurons & synaptic properties) preferences over 3 SNN structures compared for dataset 1(Left) and dataset 2 (Right).

Interestingly, [39] also demonstrates the importance of fusing temporal and frequency characteristics of EEG for affect recognition. However, in SNNs, the formulation of an algorithm for salient pattern recognition continues to be challenging. In this study, we have empirically studied the importance of SP in the said formulation process and the findings correlate with foundational works in SNNs [1] showing the ability to form function approximation with lesser computational units (i.e., neurons).

In terms of the STDP learning algorithm, we could not observe particular patterns of hyperparameters common to both datasets (Fig. 12 appended), however, larger values for A_+ and A_- seemed to push synaptic values towards the boundary values quicker (i.e., with fewer training samples) thereby missing valuable information in the data

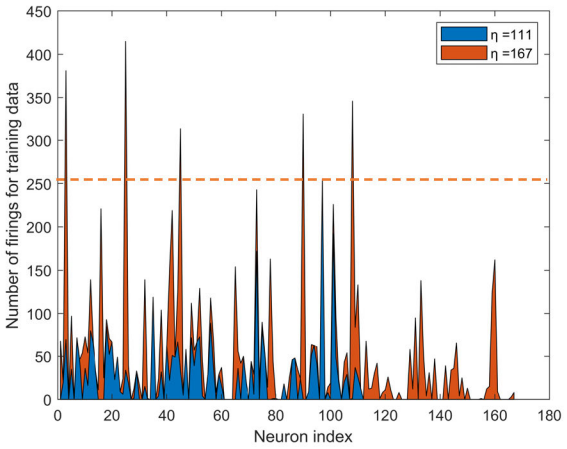


FIGURE 13. Comparison of firing count of each neuron after training with dataset 1 for network with 111 neurons (Blue) and 167 neurons (Orange).

during training. It is important to experiment further on STDP weight fluctuations with time and spiking activity of the network during training to understand behaviour patterns of A_+ and A_- that correlates with SP.

VIII. CONCLUSION & FUTURE WORK

The "tuning" of SNN structures is often seen as a matter of expert opinion and may not always be reported in sufficient depth to reproduce existing results. There is a danger that such a "black box" approach may reduce the credibility and acceptability. While this work does not solve this problem completely, it offers some potential approaches to move forward in the area.

The findings of this study a) challenge the common practice of having larger η values to produce better accuracies in SNNs and discourage arbitrary setting of the same, but do not suggest that there is one perfect neuron structure for a given data modelling task, b) highlight the importance of SP in achieving higher accuracy and adaptability of the network to increase generalization capability, and c) demonstrate structurally optimised networks producing sparser spiking activity.

The methods introduced in this paper apply SP based on evolutionary knowledge of network performance passed on over iterations. Structural adaptation using STDP learning can only make SP much more efficient. Extending this research, we foresee the possibility of developing autonomous SP algorithms using the ratio of active to total number of neurons as a measure of network learning controlled by η and LIF threshold. This can then be applied as an extension of STDP learning.

APPENDIX

See Table 7 and Figs. 12 and 13.

ACKNOWLEDGMENT

The authors would like to thank Yixon Chen and Jin Hu—visiting students from CASIA Beijing to KEDRI under the supervision of Prof. Kasabov and Prof. Hou for collecting the wrist flexion dataset.

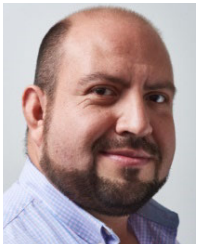
REFERENCES

[1] W. Maass, "Networks of spiking neurons: The third generation of neural network models," *Neural Netw.*, vol. 10, no. 9, pp. 1659–1671, Dec. 1997.
[2] A. Tavanaei, M. Ghodrati, S. R. Kheradpisheh, T. Masquelier, and A. Maida, "Deep learning in spiking neural networks," *Neural Netw.*, vol. 111, pp. 47–63, Mar. 2019.
[3] M. Pfeiffer and T. Pfeil, "Deep learning with spiking neurons: Opportunities and challenges," *Frontiers Neurosci.*, vol. 12, p. 774, Oct. 2018.
[4] A. Grüning and S. M. Bohte, "Spiking neural networks: Principles and challenges," in *Proc. 22nd Eur. Symp. Artif. Neural Netw., Comput. Intell. Mach. Learn. (ESANN)*, 2014, pp. 1–10.
[5] P. U. Diehl and M. Cook, "Unsupervised learning of digit recognition using spike-timing-dependent plasticity," *Frontiers Comput. Neurosci.*, vol. 9, p. 99, Aug. 2015.
[6] J. H. Lee, T. Delbruck, and M. Pfeiffer, "Training deep spiking neural networks using backpropagation," *Frontiers Neurosci.*, vol. 10, p. 508, Nov. 2016.
[7] F. Javed, Q. He, L. E. Davidson, J. C. Thornton, J. Albu, L. Boxt, N. Krasnow, M. Elia, P. Kang, S. Heshka, and D. Gallagher, "Brain and high metabolic rate organ mass: Contributions to resting energy expenditure beyond fat-free mass," *Amer. J. Clin. Nutrition*, vol. 91, no. 4, pp. 907–912, Apr. 2010.
[8] H. R. Peter, "Synaptic density in human frontal cortex—Developmental changes and effects of aging," *Brain Res.*, vol. 163, no. 2, pp. 195–205, Mar. 1979.
[9] R. Spiess, R. George, M. Cook, and P. U. Diehl, "Structural plasticity denoises responses and improves learning speed," *Frontiers Comput. Neurosci.*, vol. 10, pp. 1–13, Sep. 2016.
[10] J. Iglesias and A. E. P. Villa, "Effect of stimulus-driven pruning on the detection of spatiotemporal patterns of activity in large neural networks," *Biosystems*, vol. 89, nos. 1–3, pp. 287–293, May 2007.
[11] J. B. Aimone, "Computational modeling of adult neurogenesis," *Cold Spring Harbor Perspect. Biol.*, vol. 8, no. 4, Apr. 2016, Art. no. a018960.
[12] Y. Yuan, J. Liu, P. Zhao, F. Xing, H. Huo, and T. Fang, "Structural insights into the dynamic evolution of neuronal networks as synaptic density decreases," *Frontiers Neurosci.*, vol. 13, p. 892, Aug. 2019.
[13] G. Q. Bi and M. M. Poo, "Synaptic modifications in cultured hippocampal neurons: Dependence on spike timing, synaptic strength, and postsynaptic cell type," *J. Neurosci.*, vol. 18, no. 24, pp. 10464–10472, Dec. 1998.
[14] S. Song, K. D. Miller, and L. F. Abbott, "Competitive Hebbian learning through spike-timing-dependent synaptic plasticity," *Nature Neurosci.*, vol. 3, no. 9, pp. 919–926, Sep. 2000.
[15] D. Taylor, N. Scott, N. Kasabov, E. Capecci, E. Tu, N. Saywell, Y. Chen, J. Hu, and Z.-G. Hou, "Feasibility of NeuCube SNN architecture for detecting motor execution and motor intention for use in BCIapplications," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2014, pp. 3221–3225.
[16] S. Koelstra, C. Muhl, M. Soleymani, J.-S. Lee, A. Yazdani, T. Ebrahimi, T. Pun, A. Nijholt, and I. Patras, "Deap: A database for emotion analysis ;using physiological signals," *IEEE Trans. Affect. Comput.*, vol. 3, no. 1, pp. 18–31, Jan./Mar. 2012.
[17] S. G. Wysoski, L. Benuskova, and N. Kasabov, "Fast and adaptive network of spiking neurons for multi-view visual pattern recognition," *Neurocomputing*, vol. 71, nos. 13–15, pp. 2563–2575, Aug. 2008.
[18] J. Wang, A. Belatreche, L. Maguire, and T. M. McGinnity, "An online supervised learning method for spiking neural networks with adaptive structure," *Neurocomputing*, vol. 144, pp. 526–536, Nov. 2014.
[19] S. Dora, K. Subramanian, S. Suresh, and N. Sundararajan, "Development of a self-regulating evolving spiking neural network for classification problem," *Neurocomputing*, vol. 171, pp. 1216–1229, Jan. 2016.
[20] S. Roy and A. Basu, "An online unsupervised structural plasticity algorithm for spiking neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 4, pp. 900–910, Apr. 2017.
[21] N. Rathi, P. Panda, and K. Roy, "STDP-based pruning of connections and weight quantization in spiking neural networks for energy-efficient recognition," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 38, no. 4, pp. 668–677, Apr. 2019.
[22] Y. Shi, L. Nguyen, S. Oh, X. Liu, and D. Kuzum, "A soft-pruning method applied during training of spiking neural networks for in-memory computing applications," *Frontiers Neurosci.*, vol. 13, pp. 1–13, Apr. 2019.

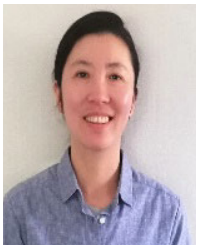
[23] S. Thorpe and J. Gautrais, "Rank order coding," in *Computational Neuroscience*. Boston, MA, USA: Springer, 1998, pp. 113–118.
[24] K. Roy, A. Jaiswal, and P. Panda, "Towards spike-based machine intelligence with neuromorphic computing," *Nature*, vol. 575, no. 7784, pp. 607–617, Nov. 2019.
[25] R. K. Chikara and L.-W. Ko, "Neural activities classification of human inhibitory control using hierarchical model," *Sensors*, vol. 19, no. 17, p. 3791, Sep. 2019.
[26] L. Ko, R. K. Chikara, Y. Lee, and W. Lin, "Exploration of user's mental state changes during performing brain–computer interface," *Sensors*, vol. 20, no. 11, p. 3169, Jun. 2020.
[27] M. M. Bradley and P. J. Lang, "Measuring emotion: The self-assessment manikin and the semantic differential," *J. Behav. Therapy Exp. Psychiatry*, vol. 25, no. 1, pp. 49–59, Mar. 1994.
[28] A. Savran, K. Ciftci, G. Chanel, and J. Mota, "Emotion detection in the loop from brain signals and facial images," in *Proc. eNTERFACE Workshop*, 2006, pp. 205–218.
[29] T. F. Bastos-Filho, A. Ferreira, A. C. Atencio, S. Arjunan, and D. Kumar, "Evaluation of feature extraction techniques in emotional state recognition," in *Proc. 4th Int. Conf. Intell. Human Comput. Interact. (IHCI)*, Dec. 2012, pp. 1–6.
[30] B. García-Martínez, A. Martínez-Rodrigo, R. Zangróniz, J. Pastor, and R. Alcaraz, "Symbolic analysis of brain dynamics detects negative stress," *Entropy*, vol. 19, no. 5, p. 196, Apr. 2017.
[31] D. Shon, K. Im, J.-H. Park, D.-S. Lim, B. Jang, and J.-M. Kim, "Emotional stress state detection using genetic algorithm-based feature selection on EEG signals," *Int. J. Environ. Res. Public Health*, vol. 15, no. 11, p. 2461, Nov. 2018.
[32] T. Delbruck and P. Lichtsteiner, "Fast sensory motor control based on event-based hybrid neuromorphic-procedural system," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2007, pp. 845–848.
[33] N. Kasabov, K. Dhoble, N. Nuntalid, and G. Indiveri, "Dynamic evolving spiking neural networks for on-line spatio- and spectro-temporal pattern recognition," *Neural Netw.*, vol. 41, pp. 188–201, May 2013.
[34] R. Naud and W. Gerstner, "The performance (and limits) of simple neuron models: Generalizations of the leaky integrate-and-fire model," in *Computational Systems Neurobiology*. New York, NY, USA: Springer, 2012, pp. 163–192.
[35] H. Markram, W. Gerstner, and P. J. Sjöström, "A history of spike-timing-dependent plasticity," *Frontiers Synaptic Neurosci.*, vol. 3, p. 4, Aug. 2011.
[36] A. J. Watt, "Homeostatic plasticity and STDP: Keeping a neuron's cool in a fluctuating world," *Frontiers Synaptic Neurosci.*, vol. 2, p. 5, Jun. 2010.
[37] W. Gerstner, R. Kempter, J. L. van Hemmen, and H. Wagner, "A neuronal learning rule for sub-millisecond temporal coding," *Nature*, vol. 383, no. 6595, pp. 76–78, Sep. 1996.
[38] R. Storn and K. Price, "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces," *J. Global Optim.*, vol. 11, no. 4, pp. 341–359, 1997.
[39] H. Chao, H. Zhi, L. Dong, and Y. Liu, "Recognition of emotions using multichannel EEG data and DBN-GC-Based ensemble deep learning framework," *Comput. Intell. Neurosci.*, vol. 2018, Dec. 2018, Art. no. 9750904.
[40] D. Wang and Y. Shang, "Modeling physiological data with deep belief networks," *Int. J. Inf. Educ. Technol.*, vol. 3, no. 5, pp. 505–511, 2013.
[41] X. Li, D. Song, P. Zhang, G. Yu, Y. Hou, and B. Hu, "Emotion recognition from multi-channel EEG data through convolutional recurrent neural network," in *Proc. IEEE Int. Conf. Bioinf. Biomed. (BIBM)*, Dec. 2016, pp. 352–359.
[42] S. Tripathi, S. Acharya, R. D. Sharma, S. Mittal, and S. Bhattacharya, "Using deep and convolutional neural networks for accurate emotion classification on DEAP dataset," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 4746–4752.
[43] S. Navlakha, Z. Bar-Joseph, and A. L. Barth, "Network design and the brain," *Trends Cogn. Sci.*, vol. 22, no. 1, pp. 64–78, Jan. 2018.
[44] P. U. Diehl, D. Neil, J. Binas, M. Cook, S.-C. Liu, and M. Pfeiffer, "Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2015, pp. 1–8.
[45] N. K. Kasabov, "NeuCube: A spiking neural network architecture for mapping, learning and understanding of spatio-temporal brain data," *Neural Netw.*, vol. 52, pp. 62–76, Apr. 2014.
[46] R. Guyonneau, R. VanRullen, and S. Thorpe, "Neurons tune to the earliest spikes through STDP," *Neural Comput.*, vol. 17, no. 4, pp. 859–879, Apr. 2005.
[47] B. Rekabdar, M. Nicolescu, R. Kelley, and M. Nicolescu, "Unsupervised learning of spatio-temporal patterns using spike timing dependent plasticity," in *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (Lecture Notes in Artificial Intelligence), vol. 8598. New York, NY, USA: Springer, 2014, pp. 254–257.

MAHIMA MILINDA ALWIS WEERASINGHE was born in Sri Lanka, in 1989. He received the B.Eng. degree (Hons.) in electronic engineering from Sheffield Hallam University, U.K., in 2011, and the M.Sc. degree in applied electronics from the University of Colombo, Sri Lanka, in 2016. He is currently pursuing the Ph.D. degree with the Auckland University of Technology, New Zealand.

He has also completed Research Assistantships in knowledge discovery research in predicting surgical outcomes and youth wellbeing, in New Zealand, where his research interests include using artificial intelligence techniques for human health and wellbeing. His current research interests include spiking neural networks, brain data processing, and computational neuroscience.



JOSAFATH I. ESPINOSA-RAMOS received the M.Sc. degree in cybernetics from La Salle University, Mexico, and the Ph.D. degree in computer science from the Centre for Computing Research, National Polytechnic Institute, Mexico. He is currently a Research Fellow with the Auckland University of Technology, New Zealand. The aim of his research is modeling multisensory and multivariate streaming data and analyzing the spatial and temporal relationships among the variables that describe the dynamics of a sensor networks. His current research interests include computational neuroscience, evolutionary algorithms, and machine learning.



GRACE Y. WANG received the B.A. degree (Hons.) in psychology, the M.Sc. degree in health sciences, and the Ph.D. degree in pharmacy from The University of Auckland, New Zealand. She is currently a Senior Lecturer of psychology and neuroscience with the Auckland University of Technology. Her recent research interests include publications on the application of computational modelling to EEG data, and the role of immune function in cognition and wellbeing in the general population.

DAVE PARRY received the degree in physics from Imperial College London, the master's degree in medical physics from St. Bartholomew's Medical College, the M.Sc. by Research degree in computer science from the University of Otago, New Zealand, and the Ph.D. degree in information retrieval from the Auckland University of Technology (AUT). Before coming to New Zealand in 1994, he worked in a number of health IT and physics jobs in U.K. He has published over 100 articles in these areas. His research interests include health informatics, knowledge-based systems, and AI in healthcare.