Applied Intelligence (2021) 51:2173–2187
https://doi.org/10.1007/s10489-020-01934-7

WOLIF: An efficiently tuned classifier that learns to classify non-linear temporal patterns without hidden layers

Irshed Hussain[1] · Dalton Meitei Thounaojam[1]

Accepted: 7 September 2020 / Published online: 29 October 2020

Abstract

We present in this paper a computationally efficient and biologically plausible classifier WOLIF, using Grey Wolf Optimizer (G**WO**) tuned error function obtained from Leaky-Integrate-and-Fire (**LIF**) spiking neuron. Unlike traditional artificial neuron, spiking neuron is capable of intelligently classifying non-linear temporal patterns without hidden layer(s), which makes a Spiking Neural Network (SNN) computationally efficient. There is no additional cost of adding hidden layer(s) in SNN, it is also biologically plausible, and energy efficient. Since supervised learning rule for SNN is still in infancy stage, we introduced WOLIF classifier and its supervised learning rule based on GWO algorithm. WOLIF uses a single LIF neuron thereby use less network parameters, and homo-synaptic static long-term synaptic weights (both excitatory and inhibitory). Note that, WOLIF also reduces the total simulation time which improves computational efficiency. It is benchmarked on seven different datasets drawn from the UCI machine learning repository and found better results both in terms of accuracy and computational cost than state-of-the-art methods.

Keywords LIF neuron · Hidden layer · GWO · Temporal pattern · Non-linear · Static long-term plasticity

1 Introduction

Sigmoidal neural network, referred to as the second generation of Artificial Neural Network (ANN) [1], is enjoying a great time in the field of computational intelligence, especially in the domain of classification, due to the capability of handling non-linear data and well known learning algorithm back-propagation. Moreover, with the flow of time, other algorithms based on metaheuristic approach is developed in the field of ANN to improve the optimisation capability of synaptic weights such as training ANN using the optimisation method based on Asexual Reproduction Optimization (ACO) [2] where ACO is applied initially to accomplish global searching of optimum synaptic weights and then back-propagation is applied to reduce overall training error, training ANN using the Butterfly Optimization Algorithm (BOA) [3] where BOA is efficiently applied to speed up the convergence rate and to minimise the risk of stagnating into local minimum. In addition, there are several other evolutionary algorithms used to efficiently train ANN for classification problems including some of neuroevolution based method for specific robotic task, in [4], a review of various such algorithms are discussed.

However, the traditional rate coding concept of sharing information among neurons in ANN through synapses is proven unlikely in neuroscience [5, 6] and because of this reason ANN fails to correctly mimic its role model i.e., the human brain. In addition, in order to solve non-linear classification problem, sigmoidal neural network uses hidden layer(s) to separate the hyperplane which is a threat to the computational cost in case of large datasets, because there is no generalized protocol of selecting the optimal number of hidden layers as well as the number of hidden neurons residing to the hidden layers. Unlike ANN, Spiking Neural Network (SNN), the third generation neural network [1] fundamentally works in a different manner which focuses more on the biological plausibility, energy efficiency and computational cost. SNN works more similar manner as the human brain works than that of its predecessor ANN. Spiking neurons in SNN are computationally powerful (use precise temporal information) [7], energy efficient and biologically plausible [8]. Spiking neurons are powerful since only using a single

✉ Irshed Hussain
ihussain.cse@gmail.com

[1] National Institute of Technology Silchar, (Computer Science and Engineering), Silchar, 788010, Assam, India

neuron it is possible to separate the hyperplane for solving non-linear classification problems.

Although neuroscience claims that the brain uses temporal coding, it is not very clear about the precise mechanism of coding [9, 10], but most popularly population coding is used by the researchers to carry out experiments with spiking neurons [11]. Biological plausibility of the spiking neuron lies on the firing behaviour, synaptic connections, and learning behaviour. Spiking neuron focuses on the potential difference between internal and external cell membrane akin to the biological neuron and when this internal state or membrane potential crosses a certain threshold the neuron fires a spike. The synapses are fully connected between each and every pre-synaptic (information sender) neuron and post-synaptic (information receiver) neuron. The value of Post-Synaptic Potential (PSP) of a post-synaptic neuron depends on the pre-synaptic spike times, synaptic weights, and synaptic delays. Although, there are several spiking neurons exist such as Spike Response Model (SRM) [12, 13], Hodgkin-Huxley [14], Leaky-Integrate-and-Fire (LIF) [15, 16] which is an improvement over the Integrate-and-Fire neuron [17], is computationally more simpler than others. Note that, LIF neuron can also be converted to SRM neuron as stated in [8].

The spike firing behaviour of the spiking neuron model used in this paper is described by the simple LIF neuron. In addition, synapse model uses double decaying kernel function where there is one-to-one mapping between pre-synaptic and post-synaptic neurons. The spike time yields from the time-to-first-spike [11] encoding scheme, is used as the spike times for input neurons. From the last couple of decays while SNN becomes popular, researchers put more attention towards the development of an efficient supervised learning algorithm.

In [11], Bohte et al. proposed an algorithm based on gradient approach called SpikeProp quite similar to the back-propagation algorithm. It uses hidden layers to separate non-linear classification problems followed by multilayer feed forward network topology. The limitations of SpikeProp include incapability of utilizing a mixture of inhibitory and excitatory neurons providing better convergence rate, which is a barrier mimicking the biological neuron. In addition, it stagnates at the local minimum and posses slow convergence rate. SpikeProp uses time-to-first-spike [11] encoding method and SRM [12, 13] neuron model. Tempotron proposed in [18], makes spiking neuron learn based on spike-time decisions. However, rather than precise training of the output neuron, it acts as a decider (neuron fires or not). In this algorithm, there is a lack of good balance between biological plausibility and computational cost. SWAT proposed in [19], trained spiking neurons to classify non-linear patterns precisely to their respective classes. The interesting fact about SWAT is that it uses dynamic synapse model [20] which work in terms of long-term plasticity. Although SWAT posses a good dimension towards the biological plausibility, it lacks behind in terms of computational cost. There are huge number of network parameters to be adjusted in SWAT which makes the algorithm incapable to work with a moderate computational power. In [21], a remote supervision based ReSuMe algorithm is proposed and well investigated for supervised learning which uses the Hebbian learning. Many other algorithms such as SPAN [22], Chronotron [23], a temporal coding based supervised learning [24], SRESN [25], and OSNN [26] is proposed to train SNN in an efficient manner. However, from the literature review we can infer that the main advantage of SNN which makes it more special than that of sigmoidal neural network, is the ability to work with a single spiking neuron and in case of non-linear data efficiently is not highlighted. In addition, there is an improper trade-off between biological plausibility and computational cost in all the aforementioned algorithms. Note that, in SEFRON [27] algorithm, the use of a single spiking neuron is explored. But, the number of encoding neurons and synapses can be reduced further to half without hampering the accuracy of the system. In this paper, a single LIF neuron and less network parameters than that of SEFRON is used and experimentally proven better with computational cost, classification accuracy, and stability.

The popularity of a classification algorithm lies basically on the learning principle, but the development of an efficient learning algorithm is a challenging task. Due to the advantages of metaheuristic approach such as simplicity, flexibility, avoiding local optimum, and derivative free mechanism, a nature inspired leadership based metaheuristic called Grey Wolf Optimizer (GWO) [28] is used to optimise the randomly initialised synaptic weights in this research. It is observed from the literature that various supervised learning algorithms do not have a well balanced trade off between computational cost and biological plausibility. Moreover, the use of too much network parameters did not improve the accuracy drastically. Note that, spiking neurons can be properly used without any hidden layers as well as hidden neurons for solving non-linear classification problems, but it is less explored due to the lack of proper mechanism to optimise synaptic weights resulting less error. Therefore, we focus on the use of single LIF neuron and its learning rule by introducing **GWO** tuned error function obtained from **LIF** neuron i.e., WOLIF classifier, which outperforms state-of-the-art methods.

Major contributions of this research paper are:

1. Proposed WOLIF classifier which uses the GWO algorithm to finely tune the error function derived from the output of LIF neuron.

2. The use of less network parameters achieved by removing hidden layer(s) and by working with less number of encoding neurons.
3. Utilization of static long-term synaptic weights (combination of both inhibitory and excitatory) to add a dimension towards the biological plausibility.
4. The optimised use of total simulation time to improve computational cost.

This paper is divided into sections followed by various subsection. The mapping of real valued features into temporal spikes, structure of the classifier, and spike firing behaviour of the neuron are discussed in Section 2. The optimisation mechanism of the synaptic weights through which the neuron learns from data followed by the description of the benchmarked datasets are discussed in Sections 3 and 4 respectively. Experimental results and interpretation of results are presented in Section 5. Finally, the conclusion drawn from this research is briefly summarized in Section 6.

2 Organisation and architecture of WOLIF

There are few things to focus on while utilizing spiking neurons in a classifier such as mapping real valued features into temporal spikes, selection of the synapse model, selection of the neuron model, and techniques to adjust the connection strength between neurons to an optimum level so that the neuron can learn from data properly. In this section, we discuss the mapping procedure of real valued features into temporal spikes followed by the architecture of the proposed model and spike firing capability of the neuron.

2.1 Mapping real features into temporal spikes

The real valued features x_f where $f \in [1, F]$ (F is the total number of real valued features) are converted into a set of pre-synaptic spike times using population encoding scheme [11] with η number of encoding neurons. Each neuron is allowed to fire a spike only once. Thus the values of x_f are converted into spike times $t_m \in [0, T_{ref}]$ (where, $m = F \times \eta$, and T_{ref} is the maximum value of encoding time i.e., set to 1 ms). Note that, $T_{ref} - 0$ is the encoding time interval denoted by ΔT and the value of ΔT is also 1 ms. Each receptive field neuron q ($q \in [1, \eta]$) has a firing strength $\mathcal{G}_f^q$, for the input data x_f which is computed using (1).

$$\mathcal{G}_f^q = exp\left(- \frac{(x_f - \mu_q)^2}{2\sigma^2} \right) \tag{1}$$

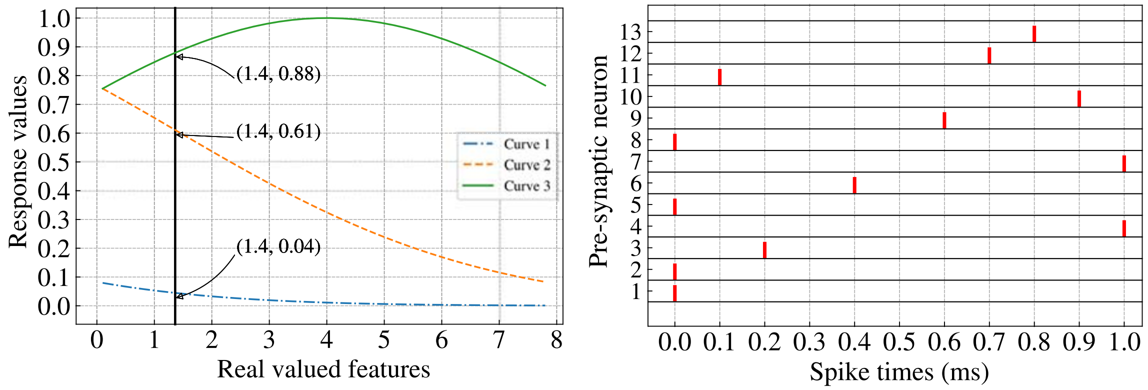
where, μ_q is the mean of the individual q Gaussian functions calculated from (2), and σ is the standard deviation represented by the (3).

$$\mu_q = \left(\frac{2q - 3}{2} \right) \times \left(\frac{1}{\eta - 2} \right) \tag{2}$$

$$\sigma = \frac{1}{\beta} \times \left(\frac{1}{\eta - 2} \right) \tag{3}$$

where, β is the adjustment factor which controls the overlapping of Gaussian curves. Finally, x_f is converted to a set of pre-synaptic spike times t_m using (4) and the response values computed from the (1).

$$t_m = T_{ref} \times \left(1 - \mathcal{G}_f^q \right) \tag{4}$$



(a) (b)

Fig. 1 **a** Illustration of yielding Gaussian response values from three encoding neurons represented by Curve 1, Curve 2, and Curve 3 upon supplying real valued features. Here, for a real valued feature 1.4, three response values 0.04 (Curve 1), 0.61 (Curve 2), and 0.88 (Curve 3) are obtained. These response values are converted into spike times of 1.0 ms, 0.4 ms, and 0.1 ms within interval [0, 1] using (4) and rounded to the next time step i.e., 0.1 ms. **b** Raster plot of pre-synaptic neurons for iris dataset with their respective spike times

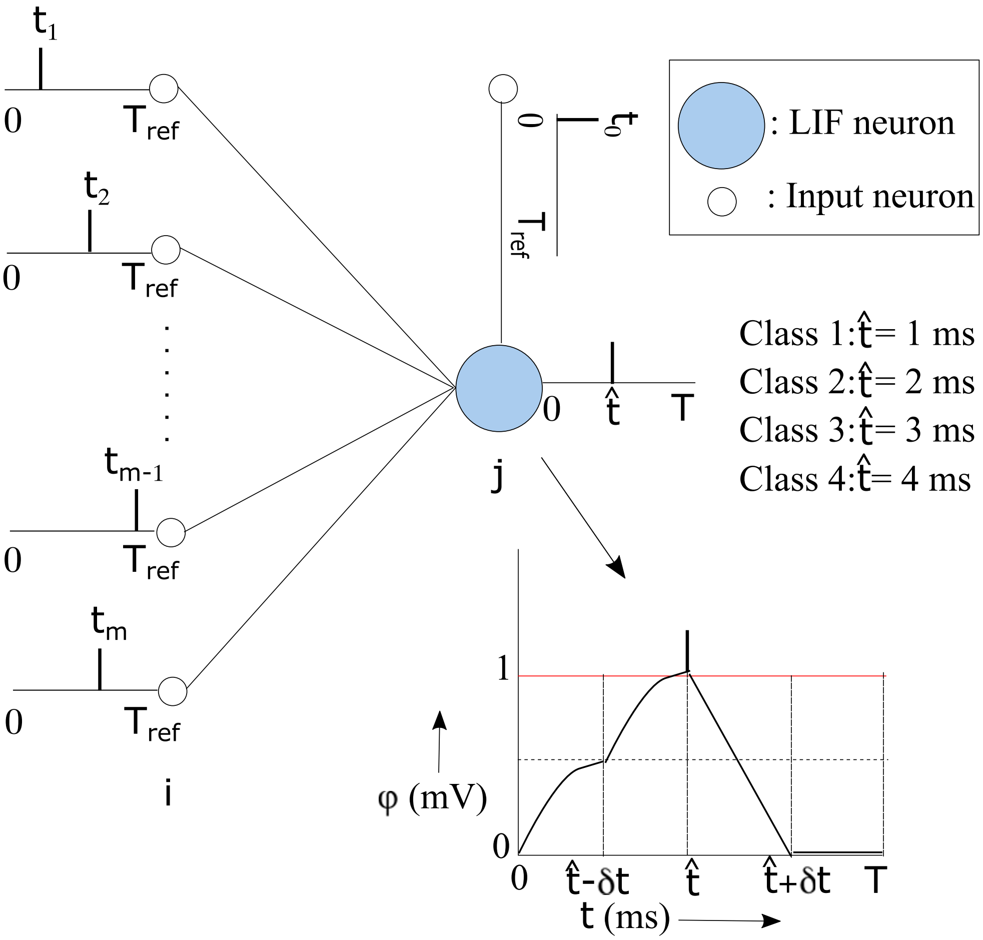


Fig. 2 Architecture of the WOLIF classifier. Pre-synaptic spikes $t_0, t_1, t_2, ..., t_m$ (t_0 is the spike time of bias neuron i.e., 0 ms) have an interval $t_i \in [0, T_{ref}]$ ms $\forall i$ and these spike times transmits through double decaying synapse model after being multiplied with synaptic weights $w_0, w_1, w_2, ..., w_m$ towards the only LIF neuron. At $\hat{t}$ ms, membrane potential crosses threshold i.e., 1 mV. Now, according to the spike time 1 ms, 2 ms, 3 ms, and 4 ms $\hat{t}$ tells the class for a particular temporal pattern

Figure 1a shows the generation of response values from Gaussian receptive fields for the iris dataset which is used for benchmarking. In Fig. 1b, each pre-synaptic neurons with its corresponding spike times are shown for one temporal pattern drawn from the encoded iris spike times.

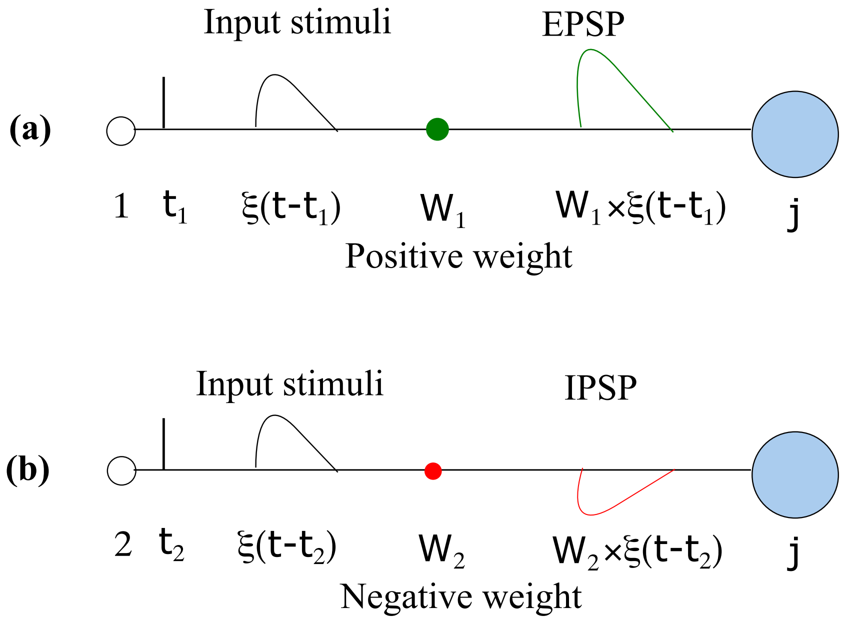


Fig. 3 **a** Excitatory synapse model. **b** Inhibitory synapse model

2.2 Encoding of temporal output spikes

Non-linear data is not easily separable and difficult to segregate into their respective classes in case of a pattern classification problem. Therefore, proper labelling of output spike times is very important and it is to be labelled in a manner so that each class is well separated from other classes in order to easily discriminate among different classes. Output spike times are generally selected trial and error basis in case of SNN within the range of total simulation time. We have selected output spike times as 1 ms for Class 1 and 2 ms for Class 2 in case of binary classification. In case of multi class (3-Class) classification such as iris dataset, we have selected output spike times as 1 ms for Class 1, 2 ms for Class 2, and 3 ms for Class 3. Also, for a 4-class classification problem such as Wireless indoor localization dataset, output spike times 1 ms represents Class 1, 2 ms represents Class 2, 3 ms represents Class 3, and 4 ms represents Class 4. Since time step is selected as 0.1 ms, each target class is well separated in all cases.

2.3 Architecture of the WOLIF

Figure 2 shows the architecture of the WOLIF classifier consisting of one input layer and one output layer. There are $m = F \times \eta$ input neurons in the input layer and only one output neuron in the output layer. The output neuron is the LIF neuron.

Although, WOLIF classifier is designed for classifying non-linear temporal pattern, there is no need of hidden layer(s) as well as hidden neurons as the only output LIF neuron is implemented and tuned very efficiently using GWO algorithm. Figure 3 shows the synapse model between pre-synaptic neuron 1 and only output neuron j, and pre-synaptic neuron 2 and only output neuron j. The synapse model selected is based on the principle of double decaying kernel function and it is described by (7). In addition, both excitatory and inhibitory synapses are used to contribute their respective excitation or inhibition into the sub threshold regime of membrane potential. Excitatory synapse produces Excitatory Post-Synaptic Potential (EPSP) for the membrane potential and inhibitory synapse produces Inhibitory Post-Synaptic Potential (IPSP) for the membrane potential. EPSP means an increase in the potential and IPSP means a decrease in the potential. In Fig. 3a it is observed that discrete spike time t_1 is converted to continuous curve (i.e., the input stimuli) using double decaying kernel function given in (7). EPSP is formed after multiplying positive weight $\mathbf{W_1}$ with input stimuli $\xi(t - t_1)$ i.e., the amplitude of the PSP raises. On the other hand, in Fig. 3b IPSP is formed after multiplying negative weight $\mathbf{W_2}$ with input stimuli $\xi(t - t_2)$ i.e., the amplitude of the PSP declines. At $\hat{t}$ ms, the membrane potential reaches threshold 1 mV and at $(\hat{t} + \delta t)$ ms comes to the rest. Since we consider the first spike only, the PSP value remains zero after firing the first spike. The output spike time $\hat{t}$ ms decides the class of a particular temporal pattern.

2.4 Spike firing behaviour of LIF neuron

The mechanism of firing a single spike is described in this section. The activity of the sub-threshold regime of the single output LIF spiking neuron, upon receiving weighted inputs through the double decay synapse model from pre-synaptic neurons, is characterised by the (5).

$$\phi(t) = \phi(t - \delta t) + \sum_{i=1}^{m} \mathbf{W_i} \times \psi(t) \tag{5}$$

where, m is the number of synapses connected to all pre-synaptic neurons, $\phi(t)$ determines the change in internal state of the only output neuron at time t, δt is the time step, $\phi(t - \delta t)$ determines the change in internal state of the output neuron at time $t - \delta t$ ($t = \delta t$ ms, indicates the neuron is at rest), $\mathbf{W} \in \mathbb{R}$ is the synaptic weight vector, and $\psi(t)$ is the input stimuli received by the output neuron from all pre-synaptic neurons m. The definition of the synapse model which produces the values for $\psi(t)$ is given in (6).

$$\psi(t) = \sum_{i=1}^{m} \sum_{t=\delta T}^{T+\delta T} \xi(t - t_i) \tag{6}$$

The definition of the double decaying kernel function is given in (7).

$$\xi(t - t_i) = \begin{cases} exp\left(\frac{t_i - t}{\tau_m}\right) - exp\left(\frac{t_i - t}{\tau_s}\right), & \text{if } t > t_i \\ 0, & \text{if } t \le t_i \end{cases} \tag{7}$$

where, τ_m is the time constant of the cell membrane, τ_s ($0 < \tau_s < \tau_m$) is another time constant called synaptic time constant. The value of τ_m and τ_s controls the rise and decay of the PSP respectively. The spike times for all the patterns presented to the output neuron is described by the set Γ_{i_p} given in (8).

$$\Gamma_{i_p} = \{t_i : 1 \le i \le m\} = \{t : \phi(t) \ge \Theta\} \tag{8}$$

where, p denotes the total number of input patterns presented to the network for training, and Θ is the threshold value.

3 Learning rule for WOLIF classifier

In this section, the tuning principle of the synaptic weights using GWO algorithm [28] is discussed. The fine tuning of network parameters in an optimised manner is referred to as the learning of the WOLIF. The primary objective of learning is to optimise the randomly initialised weights in an efficient manner, so that every neuron contribute into the membrane potential in the sub-threshold regime it can be optimised resulting less error. According to the synapse model shown in Fig. 3, it is observed that the synaptic weights form a vector $\mathbf{W}$ rather than a matrix due to the use of only one output neuron. The weight vector, $\mathbf{W} \in$ [-0.25, 1], guarantees that the synapse structure is a combination of both excitatory and inhibitory synapse. The main purpose of tuning vector $\mathbf{W}$ with the help of GWO algorithm [28] is to minimise the difference between desired spike times $\hat{t}_d^y$ and actual spike times $\hat{t}_a^y$ (where, y varies from 1 to X is the total number of samples presented to WOLIF). In GWO algorithm, the leadership is distributed among four wolves namely α, β, δ, and ω. We did not use ω wolf in this research for searching the optimum solution in the search space. The algorithm provides both exploitation (attacking prey), and exploration (searching for prey) in a very well manner which ensures that the algorithm is less likely to fall on pre-mature convergence as well as do not trap at the local optimum. The effective utilisation of GWO algorithm along with LIF neuron is described in Algorithm 1 and calculation of fitness function for the search agents are described in the Algorithm 2. In this research, a total of 30 search agents denoted by S are used to find the optimum solution in the search space by effective searching. It is observed that increasing the number of search agents did not really improve the classification accuracy, rather it increases the network load, therefore we did not vary the number of search agents.

Initially, α_s (score of α wolf), β_s (score of β wolf), and δ_s (score of δ wolf) is set to ∞. Then according to the value of f_l^E (E is the total number of epoch), the value of α_s, β_s, δ_s as well as α_p (position of α wolf), β_p (position of β wolf), and δ_p (position of δ wolf) are updated as shown in Algorithm 1. The co-efficient of searching A depends on the value of a which controls the convergence or divergence property. The value of a linearly decreases from 2 to 0 and it emphasises the exploration and exploitation. The random searching co-efficient A is given by the (9).

$$A = 2 \times a \times R - a \tag{9}$$

Another random searching co-efficient C is characterised by the (10).

$$C = 2 \times R \tag{10}$$

where, R is random number $\in$ [0, 1]. When $|A| < 1$, the algorithm converges towards the optimum solution and when $|A| > 1$, the algorithm diverges from optimum solution. The positions i.e., synaptic weights in our case are updated according to the (11), (12), and (13).

$$\mathbf{W_1} = \alpha_p - A_1 \times D_\alpha \tag{11}$$

where, A_1 is the random searching co-efficient calculated using (9) and $D_\alpha = |C_1 \times \alpha_p - \mathbf{W}|$ (C_1 is another random searching co-efficient calculated using (10)).

$$\mathbf{W_2} = \beta_p - A_2 \times D_\beta \tag{12}$$

where, A_2 is a different random searching co-efficient than A_1, calculated using (9) and $D_\beta = |C_2 \times \beta_p - \mathbf{W}|$ (C_2 is another different random searching co-efficient than C_1, calculated using (10)).

$$\mathbf{W_3} = \delta_p - A_3 \times D_\delta \tag{13}$$

where, A_3 is another different random searching co-efficient than A_1, and A_2, also calculated using (9) and $D_\delta = |C_3 \times \delta_p - \mathbf{W}|$ (C_3 is also another different random searching co-efficient than C_1, and C_2, that is also calculated using (10)). Finally, weights for the next iteration $\mathbf{W_{l+1}}$ (l varies from 1 to E) is updated using the (14).

$$\mathbf{W_{l+1}} = \frac{\mathbf{W_1} + \mathbf{W_2} + \mathbf{W_3}}{3} \tag{14}$$

Algorithm 1 WOLIF CLASSIFIER.

```
Input: A set of temporal non-linear temporal patterns.
Output: The classification accuracies (training and
        testing).
for $l \leftarrow 1$ to $E$ do
    for $i \leftarrow 1$ to $S$ do
        $f_l^E \leftarrow$ EVALUATEFITNESS($\mathbf{W}[\mathbf{i}, :], t_i^y, \hat{t}_d^y$)
        if $f_l^E < \alpha_s$ then
            $\alpha_s \leftarrow f_l^E$
            $\alpha_p \leftarrow \mathbf{W}[\mathbf{i}, :]$
        if $(f_l^E > \alpha_s) \wedge (f_l^E < \beta_s)$ then
            $\beta_s \leftarrow f_l^E$
            $\beta_p \leftarrow \mathbf{W}[\mathbf{i}, :]$
        if $(f_l^E > \alpha_s) \wedge (f_l^E > \beta_s) \wedge (f_l^E < \delta_s)$ then
            $\delta_s \leftarrow f_l^E$
            $\delta_p \leftarrow \mathbf{W}[\mathbf{i}, :]$
    $a \leftarrow 2 - l \times \left(\frac{2}{E}\right)$
    Calculate $\mathbf{W}_1$, $\mathbf{W}_2$, and $\mathbf{W}_3$ using (11), (12), and
    (13).
    $\mathbf{W_{l+1}} \leftarrow \frac{(\mathbf{W}_1+\mathbf{W}_2+\mathbf{W}_3)}{3}$
    $f_{l+1}^E \leftarrow \frac{1}{(1+\alpha_s)}$
return
```

At the convergence epoch, WOLIF finds the optimum values for the synaptic weights $\mathbf{W}_{E'}$ (Where E' is the convergence epoch) by following the objective function as given by (15).

$$O(\alpha_s) = \frac{1}{1 + \alpha_s} \tag{15}$$

where, α_s depends on loss function $\mathcal{L}(\hat{t}_d^y, \hat{t}_a^y)$ which is treated as the fitness function in this research, defined by the

(16). The loss is calculated in terms of the Mean Squared Error (MSE).

$$\mathcal{L}(\hat{t}_d^y, \hat{t}_a^y) = \frac{1}{X}\sum_{y=1}^{X}(\hat{t}_a^y - \hat{t}_d^y)^2 \tag{16}$$

In order to maximise the objective function $O(\alpha_s)$, the value of α_s has to be minimised which indicates indirectly to minimise the loss function $\mathcal{L}(\hat{t}_d^y, \hat{t}_a^y)$. After training is over, the trained or optimised weights $\mathbf{W}_{E'}$ are used to test the performance of the WOLIF classifier on a new set of testing samples.

Algorithm 2 EvaluateFitness($\mathbf{W}, t_m^y, \hat{t}_d^y$).

```
Input: A set of randomly initialised synaptic weights
       $\langle w_1, w_2, \ldots, w_n \rangle$ i.e., $\mathbf{W}$; input spike time for
       pattern $y$ i.e., $t_m^y$ (where $m$ is the dimension),
       and the desired spike time for pattern $y$ i.e., $\hat{t}_d^y$.
Output: The loss in training $\mathcal{L}$ i.e. treated as fitness.
for $y \leftarrow 1$ to $X$ do
    for $n \leftarrow 1$ to $m$ do
        for $t \leftarrow \delta t$ to $T$ do
            if $t > t_n^y$ then
                $\xi(t - t_n^y) \leftarrow exp\left(\frac{t_n - t}{\tau_m}\right) - exp\left(\frac{t_n - t}{\tau_s}\right)$
            $\phi_n(t) \leftarrow \phi_n(t - \delta t) + \mathbf{W}(n) \times \xi(t - t_n^y)$
            if $\phi_n(t) \geq \Theta$ then
                $\hat{t}_a^y \leftarrow t$
$\mathcal{L} \leftarrow \frac{1}{X}\sum_{y=1}^{X}(\hat{t}_a^y - \hat{t}_d^y)^2$
return $\mathcal{L}$
```

4 Benchmarking datasets

We have used four binary datasets (Breast cancer, Ionosphere, Liver disorders, and Pima diabetes) and one multi class dataset (Iris flower) to benchmark WOLIF classifier.

4.1 Breast cancer

Breast cancer (WBC) dataset obtained from the University of Wisconsin Hospital [29] which utilises breast cytology gained with the help of fine needle aspirations. The WBC dataset consists of 699 samples out of which 16 missing values and these missing values are removed to get a total of 683 samples in this research. There are 444 samples belong to *Benign* class and 239 samples belong to *Malignant* class [30]. The 9 real valued features of WBC are converted into 9×3+1=28 pre-synaptic input spike times (3 encoding neurons and 1 bias neuron). Hence, 28:1 is the network topology for WBC dataset where there are 28 input neurons and 1 output neuron. Output spike times are 1 ms for *Benign* class and 2 ms for *Malignant* class.

4.2 Ionosphere

Ionosphere dataset is a collection of radar data collected through antenna to classify the condition of the ionosphere whether it is in *Good* or *Bad* condition. There are a total of 351 samples, each having 33 attributes representing features [31, 32]. The network topology for ionosphere is selected as 33×3+1=100 input neurons and 1 output neuron. A spike time of 1 ms represents the condition of the ionosphere as *Good* and 2 ms represents the condition of the ionosphere as *Bad*.

4.3 Liver disorders

The liver disorder dataset consists a total of 345 samples, each sample having 6 attributes describing the features of the samples [32, 33]. This dataset classifies the condition of liver into two classes namely *Healthy* and *Unhealthy*. There are 6×3+1=19 (3 encoding neurons, and 1 bias neuron) input neurons and 1 output neuron in the network topology. The output spike time 1 ms specifies a class to the *Healthy* category and 2 ms specifies a class to the *Unhealthy* category.

Table 1 A brief summary of all datasets used for benchmarking

Dataset	Features (real)	Features (encoded)	Classes	Samples	
				Training	Testing
Breast cancer	9	28	2	350	333
Liver disorders	6	19	2	170	175
Ionosphere	33	100	2	175	176
Pima diabetes	8	25	2	384	384
Banknote authentication	4	13	2	1097	275
Iris flower	4	13	3	75	75
Wireless indoor localization	7	22	4	1600	400

Table 2 Overall performance comparison for binary datasets

Dataset	Algorithm	Topology	SL	Classification accuracy	
				Training (%)	Testing (%)
Breast cancer	SpikeProp	55:15:2	855	97.3±0.6	**97.2±0.6**
	SWAT	54:702:2	39,312	96.5±0.5	95.8±1.0
	OSNN	54:(10-16):2	560-896	91.1±2.0	90.4±1.8
	SRESN	54:(5-8)	270-432	93.9±1.8	94.0±2.6
	SEFRON	55:1	55	**98.3±0.8**	96.4±0.7
	WOLIF	**28:1**	**28**	97.8±0.4	97.0±0.2
Ionosphere	SpikeProp	199:25:2	5025	89.0±7.9	86.5±7.2
	SWAT	198:2574:2	5,14,800	86.5±6.7	90.0±2.3
	OSNN	198:(4-11):2	800-2,200	76.7±2.4	76.6±4.8
	SRESN	198:(6-13)	1,188-2,574	85.1±1.9	79.3±3.0
	SEFRON	199:1	199	**97.0±2.5**	88.9±1.7
	WOLIF	**100:1**	**100**	94.4±1.3	**90.6±1.4**
Liver disorders	SpikeProp	37:15:2	585	71.5±5.2	65.1±4.7
	SWAT	36:468:2	17,784	74.8±2.1	60.9±3.2
	OSNN	36:(4-7):2	152-266	58.7±2.2	56.7±1.8
	SRESN	36:(5-8)	180-288	59.8±1.2	57.4±1.1
	SEFRON	37:1	37	**91.5±5.4**	67.7±1.3
	WOLIF	**19:1**	**19**	81.9±0.3	**80.3±0.2**
Pima diabetes	SpikeProp	49:20:2	1020	78.6±2.5	76.2±1.8
	SWAT	48:624:2	31,200	77.0±2.1	72.1±1.8
	OSNN	48:(8-18):2	400-900	68.2±2.0	63.5±3.0
	SRESN	48:(6-12)	288-576	67.0±0.8	66.1±1.4
	SEFRON	49:1	49	**84.1±1.5**	74.0±1.2
	WOLIF	**25:1**	**25**	84.0±0.6	**83.3±0.7**

Bold entries indicate best results

4.4 Pima diabetes

Pima Indian diabetes dataset is a collection of attributes to predict whether a person is suffering from diabetes or not. All patients are female of Pima Indian heritage. There are 768 samples each having 8 real valued features [32]. A total of 8×3+1=25 pre-synaptic spike times where there are 3 encoding neurons and 1 bias neuron. The output spike time 1 ms classifies a patient to the *Diabetic* category, and 2 ms classifies to the *Non-diabetic* category. Network topology follows 25:1 structure, where there are 25 input neurons and 1 output neuron. A brief summary of the datasets containing features, classes, the number of training, and testing samples are presented in the Table 1.

4.5 Banknote authentication

Banknote authentication dataset is a non-linear binary classification problem consists of a total number of 1372 samples [32]. For this dataset, 80% of total samples were

Table 3 Comparison of computational cost of binary datasets with SEFRON

Dataset	Algorithm	SL	Epoch (E)	δ**t (ms)**	T (ms)	Computational cost ($\mathbf{C_{cost}}$)
Breast cancer	SEFRON [27]	55	**100**	0.01	4	2.2×10^{6}
	WOLIF	**28**	500	**0.1**	**2**	$\mathbf{2.8 \times 10^{5}}$
Ionosphere	SEFRON [27]	199	**100**	0.01	4	7.9×10^{6}
	WOLIF	**100**	500	**0.1**	**2**	$\mathbf{1 \times 10^{6}}$
Liver disorders	SEFRON [27]	37	**100**	0.01	4	1.48×10^{6}
	WOLIF	**19**	500	**0.1**	**2**	$\mathbf{1.9 \times 10^{5}}$
Pima diabetes	SEFRON [27]	49	**100**	0.01	4	1.96×10^{6}
	WOLIF	**25**	500	**0.1**	**2**	$\mathbf{2.5 \times 10^{5}}$

Bold entries indicate best results

Table 4 Parameter values used by WOLIF for different types of datasets

Datasets	T (ms)	δt (ms)	τ_m (ms)	τ_s (ms)	Θ (mV)	lb	ub	η
Binary	2	0.1	1.1	0.55	1	−0.25	1	3
Multiclass (3-Class)	3	0.1	1.1	0.55	1	−0.25	1	3
Multiclass (4-Class)	4	0.1	1.1	0.55	1	−0.25	1	3

used for training and 20% of the total samples were used for testing. The desired task is to classify the bank notes whether the notes are authentic or not. There are 4 real valued features based on which prediction is to be done. These real valued features are mapped into 4×3+1=13 pre-synaptic input spike times (3 encoded neurons, and 1 bias neuron). Output spike time 1 ms represents the *Authentic* class and 2 ms represents the *Non-authentic* class. Network topology for the dataset follows 13:1 structure, where there are 13 input neurons and 1 output neuron.

4.6 Iris flower

Iris dataset is a three class non-linear problem having 150 samples out of which every class consists of 50 samples each [32, 34]. Three classes represent the species of Iris plant namely *Setosa*, *Versicolor*, and *Virginica*. There are four real values features and those are mapped into 4×3+1=13 pre-synaptic input spike times (3 encoded neurons, and 1 bias neuron). Output spike time 1 ms represents *Setosa* species, 2 ms represents *Versicolor* species, and 3 ms represents *Virginicia* species. The network topology for Iris dataset is 13:1 (13 input neurons, and 1 output neuron).

4.7 Wireless indoor localization

Wireless indoor localization is a multiclass non-linear classification problem where there are 4 classes having 2000 total number of samples [35, 36]. In this case, 80% of total samples were used for training and 20% of the total samples were used for testing. There are 7 real valued attributes representing features and these are mapped into 7×3+1=22 pre-synaptic input spike times (3 encoded neurons, and 1 bias neuron). The output spike time 1 ms represents the location of *First-room*, 2 ms *Second-room*, 3 ms *Third-room*, and 4 ms *Fourth-room*. The network topology for the dataset is 22:1 (22 input neurons, and 1 output neuron).

5 Results and Discussion

To check the performance of the WOLIF classifier, it is experimented with four binary classification problems as well as one multi class classification problem.

The experimental results are compared with the state-of-the-art algorithms and found better in terms of Classification accuracy, and overall network parameters i.e., Synaptic Load (SL). Table 2 shows the network Topology, SL, Classification accuracy (Training), Classification accuracy (Testing) as well as comparison with the state-of-the-art algorithms.

In addition, Table 3 shows the comparison between SEFRON and WOLIF in case of binary classification and SpikeProp and WOLIF in case of Iris classification problem in terms of computational cost (C_{cost}) defined as in (17).

$$C_{cost} = \frac{SL \times E \times T}{\delta t} \tag{17}$$

In (17), a lower value of C_{cost} indicates a computationally efficient model. All experiments were carried out using Python 3 in a 64-bit Windows 10 operating system installed in a desktop-PC having Intel Xeon processor configured with 8 GB RAM, and 3.0 GHz clock speed. For each dataset, 10 random training trial set were generated. Table 2 shows the average training accuracy along with the standard deviation. The selection of parameter values is one of the most crucial and important step in order to work efficiently with SNN. In this section, the role and effectiveness of

Table 5 Overall performance comparison for multi class dataset

Dataset	Algorithm	Topology	SL	Classification accuracy	
				Training (%)	Testing (%)
Iris flower	SpikeProp [11]	50:10:3	530	**97.4±0.1**	96.1±0.1
	SWAT [19]	16:208:3	3952	95.5±0.6	95.3±3.6
	SRESN [25]	20:5	100	-	**97.0±0.7**
	WOLIF	**13:1**	**13**	94.1±0.2	95.1±0.4

Bold entries indicate best results

Table 6 Comparison of computational cost of multi class dataset with SpikeProp

Dataset	Algorithm	SL	Epoch (E)	δ**t (ms)**	T (ms)	Computational cost ($\mathbf{C_{cost}}$)
Iris flower	SpikeProp [11]	530	1000	0.01	50	2.65×10^9
	WOLIF	**13**	**500**	**0.1**	**3**	$\mathbf{1.95 \times 10^5}$

Bold entries indicate best results

major network parameters are also discussed along with experimental results.

Table 4 shows a brief summary of parameter values used by WOLIF where η is the number of encoding neurons, **lb**, and **ub** represents the lower and upper bound of random weight initialisation respectively.

5.1 Classification accuracy and C_{cost}

From Table 2, it is observed that in case of Breast cancer dataset, WOLIF achieves training accuracy of 97.8% and testing accuracy of 97.0% while SEFRON gives 98.3% training accuracy, SpikeProp gives 97.2% testing accuracy those are little bit higher than that of WOLIF. However, WOLIF uses very less synaptic load i.e., 28 almost half compared to SEFRON and very very less compared to other algorithms mentioned in the Table 2. In addition, Table 3 shows the better C_{cost} of 2.8×10^5 for WOLIF than that of SEFRON which is 2.2×10^6 in case of Breast cancer dataset. In case of Ionosphere dataset, training accuracy 94.4% is achieved by WOLIF that can be considered better although SEFRON gets 97.0% since SEFRON performs poor in testing accuracy and also synaptic load for WOLIF is 100 that is almost half compared to SEFRON. Moreover, for Ionosphere dataset, WOLIF has a C_{cost} of 1×10^6 and SEFRON has 7.9×10^6. From Table 2, we observe that Liver disorders dataset and Pima diabetes dataset is not easily separable from their respective classes. In these two dataset, WOLIF outperforms all the other algorithms in terms of testing accuracy and synaptic load. WOLIF gives 80.3% and 83.3% testing accuracy for Liver disorders dataset and Pima diabetes dataset respectively. Although SEFRON attains little bit higher training accuracy for the two aforementioned dataset, it shows poor performance in testing. Note that, both of these dataset has a better C_{cost} value with WOLIF as shown in Table 3. Iris dataset is a multi class classification problem where WOLIF shows 94.1% training accuracy and 95.1% testing accuracy those are very comparable with other algorithms shown in the Table 5 when synaptic load is also considered as it is one of the most important factor in case of SNN. WOLIF has a synaptic load of almost 8 times lesser than SRESN and very very less when compared with state-of-the-art algorithms.

In addition, SpikeProp has C_{cost} value of 6.625×10^9 while WOLIF has 1.95×10^5 as shown in Table 6.

In Table 7, the overall performance of WOLIF in case of all the benchmarked datasets are presented to analyse the performance of WOLIF in different classification problems having a wide variety of number of samples. It is observed from the Table 7 that the in case of 4 class classification problem (Wireless indoor localization) also, WOLIF shows a good training accuracy of 84.6% and a good testing accuracy of 84.8% if we consider the number of epochs. In addition, for Banknote authentication dataset, WOLIF shows very satisfactory training accuracy and testing accuracy those are 95.5% and 93.2% respectively.

Figure 4a shows the training accuracy curve for Breast cancer dataset and Fig. 4b shows the training accuracy curve

Table 7 Overall performance of WOLIF in case of all the seven datasets

Algorithm	Datasets	Epoch	Classification accuracy	
			Training ($\pm\sigma$) %	Testing ($\pm\sigma$) %
WOLIF	Breast cancer	500	97.8 (0.4)	97.0 (0.2)
	Ionosphere	500	94.4 (1.3)	90.6 (1.4)
	Liver disorders	500	81.9 (0.3)	80.3 (0.2)
	Pima diabetes	500	84.0 (0.6)	83.3 (0.7)
	Iris flower	500	94.1 (0.2)	95.1 (0.4)
	Banknote authentication	500	95.5 (0.9)	93.2 (1.7)
	Wireless indoor localization	500	84.6 (1.5)	84.8 (1.8)
	Average ($\pm\sigma$)	500 (0.0)	90.3 (6.0)	89.2 (5.9)

Bold entries indicate best results

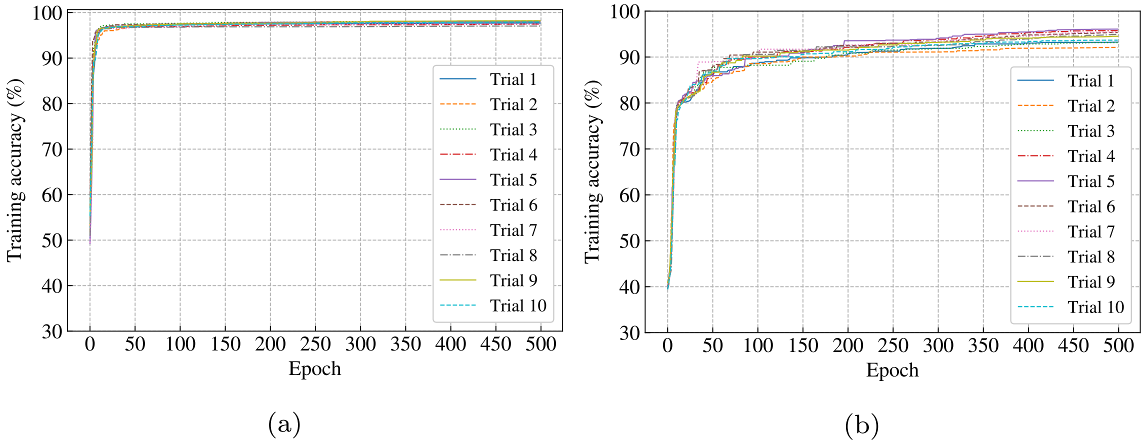


Fig. 4 **a** Training curve of Breast cancer dataset with all 10 set of random trials. **b** Training curve of Ionosphere dataset with all 10 set of random trials

for Ionosphere dataset along with all 10 set of random trials. Figure 5a and b show the training accuracy curve for Liver disorders dataset and Pima diabetes dataset respectively along with all 10 set of random trials. In Fig. 6a, training accuracy curve for Iris flower dataset is presented along with 10 set of random trials.

Figure 6b shows the behaviour of PSPs in case of three different class samples drawn from Iris dataset after training. The first PSP reaches the threshold at 1.2 ms that was supposed to be 1 ms in case of the best case scenario that means PSP reaches the threshold 0.2 ms late. Likewise other two PSPs reaches the threshold at 2.2 ms (desired is 2 ms), and 2.9 ms (desired is 3 ms) respectively. In this case, MSE for class *Setosa* is 4%, MSE for class *Versicolor* is 4%, and MSE for *Virginica* is 1%. Overall loss in terms of MSE is 3% i.e., training accuracy of 97% where each class is having only one sample. Figure 7a and b shows the behaviour of WOLIF while training in case of Banknote authentication and Wireless indoor localization datasets respectively. All 10 random trials are clearly shown in Fig. 7a and b respectively.

5.2 Effect of τ_m and τ_s

The major focus in SNN is the efficient updating process of the membrane potential so that PSP can reach the threshold value not so early as well as not so late. Time constants such as τ_m and τ_s plays a very crucial role along with synaptic weights $\mathbf{W}$ in the synapse model those contributes information to the sub-threshold regime. τ_m controls the rising of PSP curve towards threshold and τ_s controls decaying width of the PSP curve which is necessary for overlapping in case of multiple PSPs. Therefore, good selection of the values for τ_m and τ_s improves the spike firing capability of the neuron. A high value of τ_m, forces the neuron to fire a spike too early and a low value of τ_m does not allow a PSP to raise easily towards threshold. Since weights get multiplied with τ_m and τ_s indirectly, it becomes

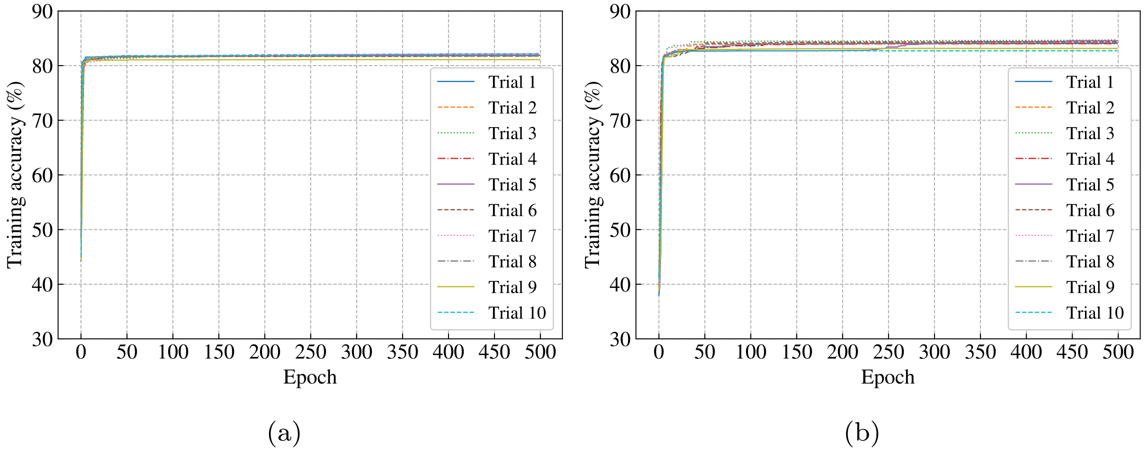
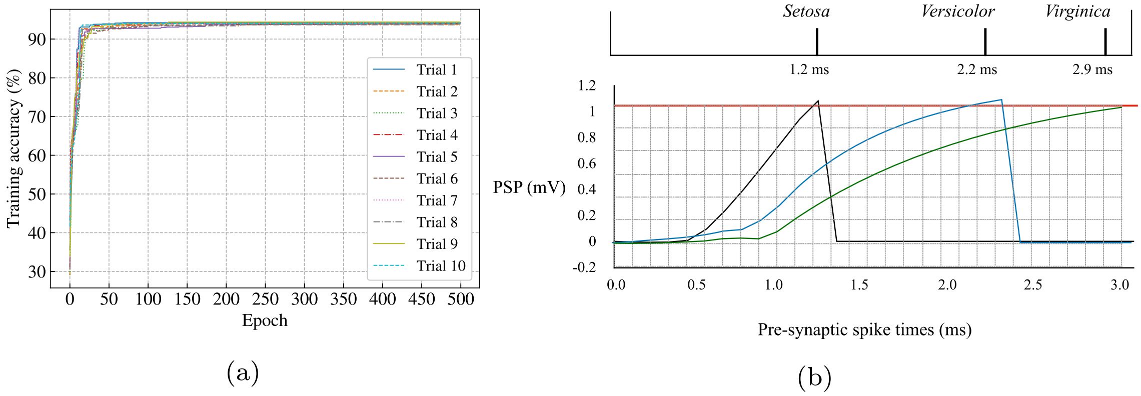


Fig. 5 **a** Training curve of Liver disorders dataset with all 10 set of random trials. **b** Training curve of Pima diabetes dataset with all 10 set of random trials



(a)

(b)

Fig. 6 **a** Training curve of Iris flower dataset with all 10 set of random trials. **b** Illustration of the classifying behaviour of WOLIF in case of Iris dataset after training for three samples of different classes

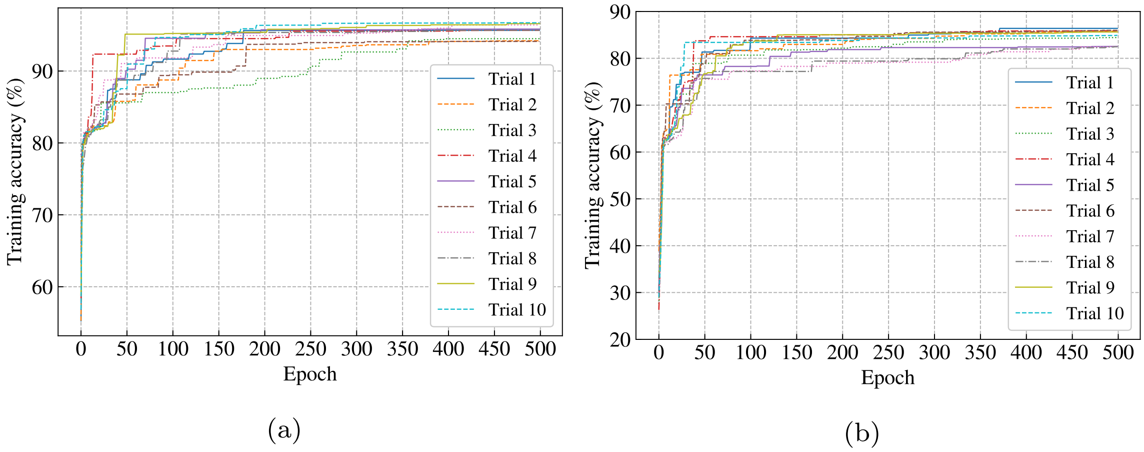
more important to make a balance between the selection of values for τ_m and τ_s. In addition, a higher value of τ_s produces wide shaped PSPs and therefore less overlapping happens among multiple PSPs. It is recommended to select the value of τ_m slightly greater than the encoding interval ΔT i.e., 1 ms in our case. The value of τ_m is set to 1.1 ms that is just one time step ahead (since we have used time step δt as 0.1 ms). Note that, the value of τ_s is set to half of τ_m and found better experimentally. Figure 8a and b shows the shape of PSPs upon varying τ_m while keeping τ_s half of τ_m and the shape of PSPs upon varying τ_s while keeping τ_m 1.1 ms respectively. From Fig. 8a and b, the effect of τ_m and τ_s on the rise and decay of PSP is clearly interpretable. Moreover, Fig. 9a shows the shape of PSP by varying both τ_m and τ_s where the need of overlapping is clearly presented.

5.3 Effect of weights initialisation range

SNN is very sensitive to synaptic weights which is a very important parameter that affects the spike firing behaviour of a neuron directly. Therefore, initialisation of weights to some random value has to be done very carefully. We applied heuristic rule for the selection of weights initialisation range. The rule is to set the upper limit less or equal to threshold and lower limit to some small negative values. We have selected the range as [-0.25, 1] where the negative lower limit allows 20% negative weights from [-0.25, 0) which corresponds to the inhibitory synapse and the positive upper limit allows 80% positive weights from [0, 1] which corresponds to the excitatory synapse. Figure 9b shows the shape of an excitatory PSP and an inhibitory PSP where shapes clearly explain their role towards the updating process of PSPs. Although many researchers claim that a mixture of inhibitory and excitatory synapse does not allow a classifier to converge easily, we have used the same in an efficient way with better convergence rate.

5.4 Role of bias neuron

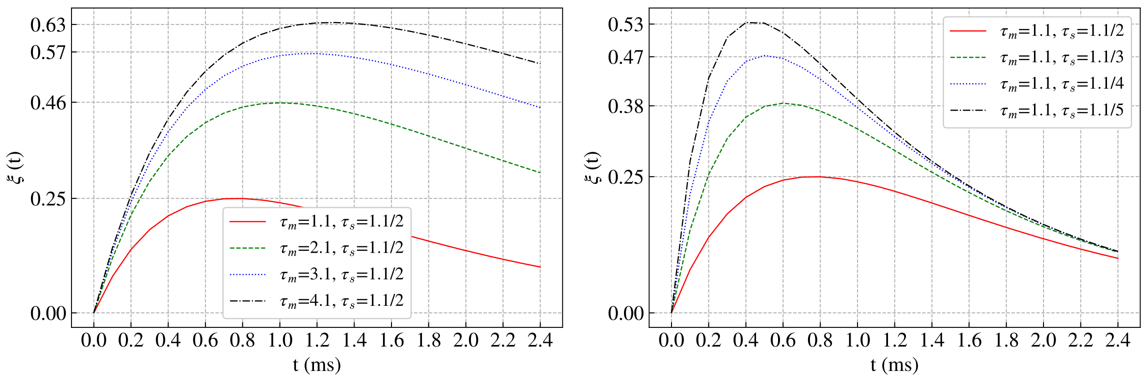
The bias neuron initially starts the membrane potential updating process if there are more number of lateral



(a)

(b)

Fig. 7 **a** Training curve of Banknote authentication dataset with all 10 set of random trials. **b** Training curve of Wireless indoor localization dataset with all 10 set of random trials



(a) (b)

Fig. 8 **a** The value of τ_m varies keeping τ_s same i.e., half of the first value of τ_m. **b** The value of τ_s varies keeping τ_m same i.e., 1.1 ms

pre-synaptic spike times. Therefore, the spike time for the bias neuron is set to a very early pre-synaptic spike time i.e., 0 ms.

5.5 Role of time step

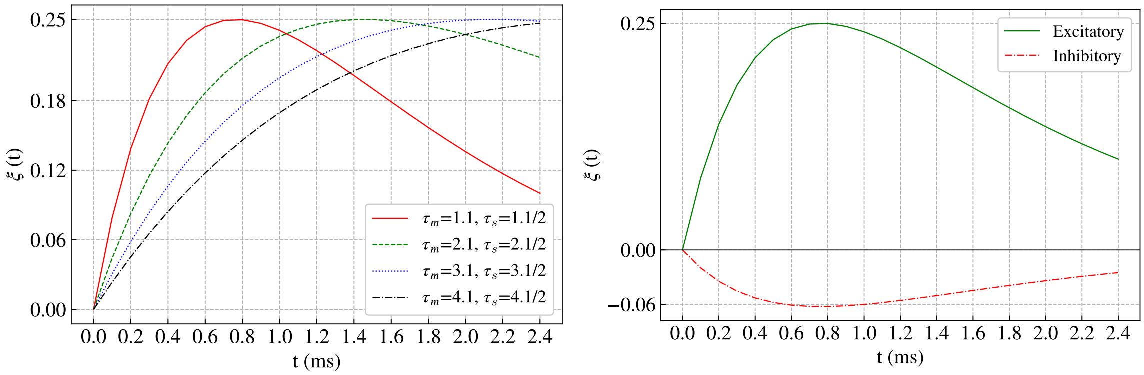
A small value of time step δt takes more iteration over total simulation time T thereby increases computational cost. We have used T as 2 ms for binary classification problem, 3 ms for 3-class classification problem, and 4 ms for 4-class classification problem. The value of δt is set to 0.1 ms, therefore it takes at most 20 iterations to produce a spike for binary classification problem, at most 30 iterations to produce a spike in case of 3-class classification problem, and at most 40 iterations to produce a spike in case of 4-class classification problem. However, a large value of δt over T does not allow SNN to learn form non-linear data properly and thus affects the training of the classifier. In SEFRON, the value of T was taken as 4 ms and δt was taken as 0.01 ms i.e., computationally costlier than that of WOLIF.

5.6 Effect of encoding neurons

The selection of the number of encoding neuron η directly affects the computational cost. A higher value of η means higher is the number of input neurons since population encoding is used in this research. Therefore, it has to be selected in a very careful manner. We set the value of η as 3 therefore we successfully minimise the total network load in terms of synaptic connections to an optimum level and it is clearly visible in the Tables 2 and 5.

5.7 Stability and generalisation

From Tables 2, 5, and 7 when accuracies are analysed it is observed that WOLIF is more stable in case of random set of trial since the standard deviation does not differ very much from the mean value of accuracies. Moreover, the capability of handling versatile dataset with the minimal synaptic load and without hidden layer in case of non-linear



(a) (b)

Fig. 9 **a** The value of τ_m varies by 1 ms each and accordingly value of τ_s varies. In all cases the value of τ_s is exactly half of the value of τ_m. **b** Unweighed excitatory and inhibitory synapse temporal patterns, inclines WOLIF towards the property of generalisation.

6 Conclusion

In this paper, an efficient classifier WOLIF along with its learning rule in order to classify non-linear temporal patterns has been presented. WOLIF shows very impressive training and testing accuracy as well as computational cost which uses GWO algorithm for weights optimisation and LIF neuron having double decaying synapse model for the generation of temporal spikes. It is both biologically plausible and computationally efficient. The usage of static long-term synaptic weights that is a combination of both inhibitory and excitatory synapses justifies the biological plausibility. WOLIF outperforms state-of-the-art algorithms in case of binary classification and almost equally performed in case of multi class classification problem. The total simulation time is also reduced to improve computational cost compared to the state-of-the-art algorithms. In addition, the stability and generalisation of WOLIF classifier is also mentionable.

In future work, WOLIF can be improved further by allowing to fire multiple spikes from the same neuron.

Compliance with Ethical Standards

Conflict of interests The first author and second author declare no conflict of interest.

References

1. Maass W (1997) Networks of spiking neurons: the third generation of neural network models. Neural Networks 10(9):1659–1671
2. Ahmadian S, Khanteymoori AR (2015) Training back propagation neural networks using asexual reproduction optimization. In: 2015 7Th conference on information and knowledge technology (IKT), IEEE, pp 1–6
3. Jalali SMJ, Ahmadian S, Kebria PM, Khosravi A, Lim CheeP, Nahavandi S (2019) Evolving artificial neural networks using butterfly optimization algorithm for data classification. In: International conference on neural information processing, Springer, pp 596–607
4. Jalali SMJ, Ahmadian S, Khosravi A, Mirjalili S, Mahmoudi MR, Nahavandi S (2020) Neuroevolution-based autonomous robot navigation: A comparative study. Cognitive Systems Research
5. Cariani PA (2004) Temporal codes and computations for sensory representation and scene analysis. IEEE Trans Neural Netw 15(5):1100–1111
6. Gautrais J, Thorpe S (1998) Rate coding versus temporal order coding: a theoretical approach. Biosystems 48(1-3):57–65
7. Maas W (1997) Noisy spiking neurons with temporal coding have more computational power than sigmoidal neurons. Adv Neural Inform Process Syst 9:211–217
8. Gerstner W, Kistler WM (2002) Spiking neuron models: Single neurons, populations, plasticity. Cambridge University Press, Cambridge
9. Bialek W, Rieke F, De Ruyter Van Steveninck RR, Warland D (1991) Reading a neural code. Science 252(5014):1854–1857
10. Natschläger T, Ruf B (1998) Spatial and temporal pattern analysis via spiking neurons. Network: Computation in Neural Systems 9(3):319–332
11. Bohte SM, Kok JN, La Poutre H (2002) Error-backpropagation in temporally encoded networks of spiking neurons. Neurocomputing 48(1-4):17–37
12. Kistler WM, Gerstner W, Leo van Hemmen J (1997) Reduction of the hodgkin-huxley equations to a single-variable threshold model. Neural Comput 9(5):1015–1045
13. Gerstner W (1995) Time structure of the activity in neural network models. Phys Rev E 51(1):738
14. Hodgkin AL, Huxley AF (1952) A quantitative description of membrane current and its application to conduction and excitation in nerve. The Journal of Physiology 117(4):500–544
15. Stein RB (1965) A theoretical analysis of neuronal variability. Biophys J 5(2):173–194
16. Stein RB (1967) Some models of neuronal variability. Biophysical J 7(1):37–68
17. Vazquez RA, Cachón A (2010) Integrate and fire neurons and their application in pattern recognition. In: 2010 7Th international conference on electrical engineering computing science and automatic control, IEEE, pp 424–428
18. Gütig R, Sompolinsky H (2006) The tempotron: a neuron that learns spike timing–based decisions. Nature Neuroscience 9(3):420–428
19. Wade JJ, McDaid LJ, Santos JA, Sayers HM (2010) Swat: a spiking neural network training algorithm for classification problems. IEEE Trans Neural Networks 21(11):1817–1830
20. Tsodyks MV, Markram H (1996) Plasticity of neocortical synapses enables transitions between rate and temporal coding. In: International conference on artificial neural networks, Springer, pp 445–450
21. Ponulak F, Kasiński A (2010) Supervised learning in spiking neural networks with resume: sequence learning, classification, and spike shifting. Neural Computation 22(2):467–510
22. Mohemmed A, Schliebs S, Matsuda S, Kasabov N (2012) Span: Spike pattern association neuron for learning spatio-temporal spike patterns. Int J Neural Syst 22(04):1250012
23. Florian RV (2012) The chronotron: A neuron that learns to fire temporally precise spike patterns. PloS One 7(8):1–27
24. Mostafa H (2017) Supervised learning based on temporal coding in spiking neural networks. IEEE Trans Neural Netw Learn Syst 29(7):3227–3235
25. Dora S, Subramanian K, Suresh S, Sundararajan N (2016) Development of a self-regulating evolving spiking neural network for classification problem. Neurocomputing 171:1216–1229
26. Wang J, Belatreche A, Maguire L, Mcginnity TM (2014) An online supervised learning method for spiking neural networks with adaptive structure. Neurocomputing 144:526–536
27. Jeyasothy A, Sundaram S, Sundararajan N (2018) Sefron: a new spiking neuron model with time-varying synaptic efficacy function for pattern classification. IEEE Trans Neural Netw Learn Syst 30(4):1231–1240
28. Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. Advances Eng Soft 69:46–61
29. Wolberg WH, Mangasarian OL (1990) Multisurface method of pattern separation for medical diagnosis applied to breast cytology. Proceedings of the National Academy of Sciences 87(23):9193–9196
30. Wolberg WH (1992) Breast cancer wisconsin dataset. http://archive.ics.uci.edu/ml/datasets/

31. Sigillito VG, Wing SP, Hutton LV, Baker KB (1989) Classification of radar returns from the ionosphere using neural networks. Johns Hopkins APL Technical Digest 10(3):262–266
32. Dua D, Graff C (2017) UCI machine learning repository
33. McDermott J, Forsyth RS (2016) Diagnosing a disorder in a classification benchmark. Pattern Recogn Lett 73:41–43
34. Fisher RA (1936) The use of multiple measurements in taxonomic problems. Annals of Eugenics 7(2):179–188
35. Bhatt R (2005) Fuzzy-rough approaches for pattern classification: Hybrid measures, mathematical analysis, feature selection algorithms, decision tree algorithms, neural learning, and applications. In: Decision tree algorithms, neural learning, and applications. Amazon books
36. Rohra JG, Perumal B, Narayanan SJ, Thakur P, Bhatt RB (2017) User localization in an indoor environment using fuzzy hybrid of particle swarm optimization & gravitational search algorithm with neural networks. In: Proceedings of Sixth international conference on soft computing for problem solving, Springer, pp 286–295

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Irshed Hussain was born in Assam, India and received the M.Tech. degree in Computer Science and Engineering from National Institute of Technology Silchar, Silchar-788010, Assam, India in 2016. He is currently pursuing Ph.D. in Computer Science and Engineering at National Institute of Technology Silchar, Silchar-788010, Assam, India. His current research interests include artificial neural networks, spiking neural networks, machine learning, metaheuristics, neuro-fuzzy system, EEG signal processing, and speech processing.

Dalton Meitei Thounaojam was born in Manipur, India. He did M.E. in Computer Science and Engineering and Ph.D. in Information Technology from Anna University Coimbatore, India and Assam University Silchar, India in 2009 and 2017 respectively. He is an assistant professor in the department of Computer Science and Engineering at National Institute of Technology Silchar, Silchar-788010, Assam, India. His research interests include artificial neural network, image processing, video processing, fuzzy system, and genetic algorithm.