



Spiking Neural Network Learning Models for Spike Sequence Learning and Data Classification

Moses Apambila Agebure^{1*}, Edward Yellakuor Baagyere¹
and Elkanah Olaosebikan Oyetunji²

¹Computer Science Department, Faculty of Computing and Information Sciences, CK Tedom
University of Technology and Applied Sciences, Navrongo, Ghana.

²Department of Mechanical Engineering, Faculty of Engineering, Lagos State University, Epe
Campus, Nigeria.

Authors' contributions

Author AAM designed and implemented the algorithms, conducted experiments and prepared the first draft of the paper. EYB reviewed the results and conducted further analyses. EOY reviewed and fine-tuned the analysis. All authors read and approved the final paper.

Article Information

DOI: 10.9734/AJRCOS/2020/v6i430163

Editor(s):

(1) Dr. Xiao-Guang Lyu, Huaihai Institute of Technology, China.

Reviewers:

(1) C. K. Gomathy, SCSVMV Deemed to be University, India.

(2) Ali Rza Sögüt, Konya Technical University, Turkey.

(3) Shahid Mumtaz, University of Aveiro, Portugal.

Complete Peer review History: <http://www.sdiarticle4.com/review-history/62867>

Received: 26 September 2020

Accepted: 01 December 2020

Published: 09 December 2020

Original Research Article

ABSTRACT

Supervised learning in Spiking Neural Network (SNN) is a hotbed for researchers due to the advantages temporal coded networks provide over that of rate-coded networks with respect to efficiency in information processing and transfer rates. Supervised learning in rate-coded networks though well established, it is difficult to directly apply such models to SNN due to difference in information coding schemes. In this paper, we seek to exploit the advantages of spiking neural networks for spike sequence learning in order to establish two (2) models; batch and sequential learning models for solving data classification tasks. The models are built using the least squares approach leveraging on its approximation abilities. The first set of experiments are on spike sequence learning in which an extensive evaluation of the model is performed using different input-output firing rates and learning periods. Results from these experiments show that the proposed

*Corresponding author: E-mail: magebure@cktutas.edu.gh;

model for spike sequence learning produced better performance than some existing models derived for spike sequence learning, particularly, at higher learning periods. The proposed models for data classification are also tested on some selected benchmark datasets most of which had imbalance class distributions and also on real-world road condition datasets for anomaly classification collected by the authors as part of a larger study. While the proposed models generalised very well to all datasets including those with the class imbalance problem where F1 and Recall values above 0.90 were recorded, some well-know machine learning algorithms applied to the datasets yielded lower F1 and Recall values and in some cases recorded 0.0 Recall.

Keywords: Spiking neural network; Spike sequence learning; data classification; class imbalance learning; least squares method.

2010 Mathematics Subject Classification: 53G25, 83C05, 57N16

1 INTRODUCTION

Supervised learning in Spiking Neural Network (SNN) is an emerging area in machine learning and its application has been focused on spike sequence learning [1–3] and data classification [4–7] with much emphasis on the later because of its wide scale application. Information is encoded as temporal codes in SNN contrary to rate-based codes in classical Artificial Neural Network (ANN). This difference in information coding makes it difficult to directly apply learning algorithms for ANN to SNN. This, coupled with the prospects of better information processing and transfer in temporal codes increased the demand for efficient learning models for SNN.

Spike sequence learning involves training output neurons of SNNs to emit spike patterns at desired times and thus, require learning models capable of training networks with multi-spiking neurons. Experiments on spike sequence learning have been mainly conducted using single layer networks and the learning models derived using Spike Timing Dependant Plasticity (STDP) and anti-STDP mechanisms [1–3]. The learning model proposed in [1] dubbed the ReSuMe is considered a modification of the Widrow-Hoff (Delta) rule [8], derived originally for non-spiking linear units to incorporate the STDP processes to enable biologically plausible supervised learning. Though ReSuMe so far is considered one of the most biological plausible supervised learning models, it becomes exceedingly difficult for trained networks to converge when the number of desired output spikes is more than one. To improve on the convergence rate of ReSuMe, [2]

and [3], modified it to include delay learning (DL-ReSuMe) and extended delay learning (EDL-ReSuMe) respectively, which they reported to have improved convergence rates.

There are however, several supervised SNN models derived using different learning mechanism for different network architectures for data classification tasks. First among these is the Error Back propagation based learning model (SpikeProp) proposed by [4], which was tested using a two (2) layer network. Neurons in networks trained using this model could only emit single spikes. Following the fact that networks with multi-spiking neurons are required to efficiently solve complex classification tasks, the SpikeProp was modified by several researchers including [5, 6, 9], to enable multiple spike learning.

There have also been a significant attempt to derive learning models for data classification in which network parameter optimisation and training are governed by evolutionary processes and algorithms [10–12]. Though these models produced good classification performance, neurons in these networks could only be simulated using single spikes similar to the model proposed by [4].

The third set of SNN models for data classification are those that are derived as modifications to the ReSuMe model [7, 13]. The model proposed in [13], modified the ReSuMe to include some properties of error function definition used in gradient decent for training multi-layer networks. [7] on the other hand, extended the ReSuMe model into a multi-layer

learning model that allowed the output neurons to emit any number of spikes per class label. There have also been a few recent studies that sought to derive and investigate the suitability of deep learning SNN models for solving general machine learning tasks [14].

Contrary to the wide use of error back propagation, evolutionary algorithms, and STDP based methods in existing supervised SNN learning models, the models presented in this paper employ a training method that elicits the approximation capability of the least squares method to compute the change in synaptic weights required to enable output neurons emit spikes at desired times. The weight update scheme was originally presented as a conference paper [15]. In [15], the derivation of the model was presented and evaluated on spike sequence learning task using a single set of input/output spike frequencies and its performance compared to the ReSuMe model [1].

The motivation of this paper is to investigate and sufficiently establish the viability of applying the proposed weight update scheme to training SNN for solving different machine learning tasks under different settings and learning principles. Furthermore to this, further investigations on the model proposed by [15] on spike sequence learning is conducted by subjecting it to simulations using different input/output mean firing rates and learning periods and its performance compared to some existing models derived for spike sequence learning. The weight update scheme used in the spike sequence learning model is adopted to formulate learning models for solving classification tasks in which two modes of learning; batch and sequential learning approaches are presented. The batch model, which we refer to as a normal SNN model (nSNN) is intended for learning from existing collection of datasets as happens in normal supervised learning while the sequential model dubbed sSNN, is proposed for application areas where it is difficult to get a bulk of data for learning at a go. The sequential model thus allows continuous learning and would be ideal for application in most real-world settings where data is generated sequentially. We present both models in this paper because we want to conduct a side-by-side comparison of the models'

performance under the same experimental conditions, which would serve as a guide for our future research decisions. The classification models are evaluated on some benchmark datasets with emphasis on datasets with class imbalance problem. This is to enable us assess the models' performance on datasets with known special challenges such as the class imbalance problem that militates against classification performance of machine learning algorithms. In addition, the proposed classification models are tested on real-world road condition datasets for road anomaly detection and classification and their performance compared to some existing classification algorithms.

The classification models proposed in this paper are tested with single layer networks but are extensible to deep learning networks, which would be investigated in our future research work. In the proposed model for batch learning, all instances in a training set are passed through the network in each learning iteration and the synaptic weights adjusted with respect to each instance and associated class label. With regards to the sequential learning model each instance in the training set is presented to the network one at a time for learning in that the weights are adjusted over several iterations until the network is able to correctly classify the given instance or a given number of iterations are exhausted before another instance is presented. No instance is presented to the network more than once for training in the sequential model.

In a summary this paper makes the following contributions:

- Conducted an extensive evaluation of the proposed model on spike sequence learning in order to validate the model's performance on different learning conditions.
- A batch and sequential learning models for data classification are proposed and implemented using the least squares weight update scheme.
- We validated the performance of the implemented models using some benchmark datasets and road surface anomaly detection and classification datasets collected by the authors.

$$\begin{bmatrix} (\Delta z_1 - 1) & \Delta z_1 & \dots & \Delta z_1 \\ \Delta z_2 & (\Delta z_2 - 1) & \dots & \Delta z_2 \\ \vdots & \vdots & \ddots & \vdots \\ \Delta z_I & \Delta z_I & \dots & (\Delta z_I - 1) \end{bmatrix} \begin{bmatrix} \Delta w_1 \\ \Delta w_2 \\ \vdots \\ \Delta w_I \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_I \end{bmatrix} \quad (2.4)$$

where $c_i = w_i - \Delta z_i (\sum_{i=1:I} w_i - 1)$.

Training a neural network to emit a sequence of spikes within a period T using the above scheme is quite straight forward and only requires updating the synaptic weights at two distinct event times; desired and undesired spike times. At each desired output spike time t_d , where there is no spike, the synaptic weights of input neurons that have spikes preceding the desired spike time are adjusted using equation (2.5).

$$w_{ji} = w_{ji} + \Delta w_{ji} \quad (2.5)$$

Increasing the weights of input neurons contributing to t_d is expected to cause the PSP of the output neuron to cross the threshold voltage at t_d . In the course of learning, there is a tendency that, the output neuron will emit spikes at undesired times, which have to be cancelled. To cancel undesired spikes, the synaptic weights of input neurons with spikes contributing to an undesired spike are reduced using equation (2.6).

$$w_{ji} = w_{ji} - \Delta w_{ji} \quad (2.6)$$

The weight updates using equation (2.5) and (2.6) are repeated over several learning iteration until a set value of a performance metric is obtained or the maximum number of iterations are completed.

The correlation-based metric [1], which measures the similarity between a network's actual output spike train and a desired spike train is used to measure the performance of the models in this paper as done in [15]. The correlation-based metric, C is defined by equation (2.7),

$$C = \frac{v_d \cdot v_o}{|v_d||v_o|}, \quad (2.7)$$

where v_d and v_o are vectors, which are convolutions of the desired and actual output spike trains respectively, using a symmetric Gaussian filter given by equation (2.8). The width of the function is determined by δ . The term

$v_d \cdot v_o$ is the inner product and $|v_d|$ and $|v_o|$ are the length of the vectors v_d and v_o , respectively.

$$f(t, \delta) = \frac{-t^2}{e^{2\delta^2}} \quad (2.8)$$

The values of C range from 0 to 1; 0 means no correlation and 1 high correlation between a desired and actual network spike trains. SNNs are therefore trained to obtain values of C equal to or closer to 1.

2.1.2 Proposed SNN Learning Models for Data Classification

The weight update scheme defined by equation (2.5) and (2.6) is adopted in the models for data classification tasks. Instead of having one desired spike train in the case of the spike sequence learning, there are two or more sets of distinct spike trains that a network for classification task must be trained to emit. Each spike train represents a class label in a dataset. The network can also be simulated with multiple output neurons, each neuron representing a class label in the dataset, but a single neuron emitting multiple spikes at different times for each class label is investigated in this paper.

The spike trains for class labels are obtain by generating a spike train with evenly spaced spikes whose size is twice the number of class labels in a given dataset within a learning period T . The size of the evenly spaced spike train is set to twice the number of class labels in a dataset because each label is encoded with two (2) spikes in this paper. As mentioned in Section 1, two modes of learning for data classification; the batch and sequential learning models are proposed and investigated in this paper. In both cases attributes in the datasets are encoded into populations of spikes using five (5) ($M = 5$) identical Gaussian Receptive Fields (GRF) defined by equation (2.9) [4]. Each field is

centred at μ_i with a spread σ defined respectively, by equations (2.10) and (2.11),

$$g_i(a) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(a - \mu_i)^2}{\sigma^2}\right), \quad (2.9)$$

$$\mu_i = a_{min} + \left(\frac{2i - 3}{2}\right) \left(\frac{a_{max}^n - a_{min}^n}{M - 2}\right), \quad (2.10)$$

$$\sigma = \frac{\left(\frac{1}{\gamma}\right)(a_{max} - a_{min})}{M - 2}, \quad (2.11)$$

where $i = \{1, 2, \dots, M\}$ and γ is an adjustment factor, which is set to 1.5 similar to what is used in previous studies [4, 6, 7].

Batch Learning Model

In the batch model, training is repeated over several iterations. In each iteration, the temporal codes for each instance in a training dataset is presented to the network and the weights are updated to allow spikes at the desired spike times of the class it belongs to and also to cancel all undesired spikes. Undesired spikes in relation to a given instance include spikes emitted at desired times of different classes. The training process is repeated until the maximum number of iterations are completed or a set value of classification metric is met. The maximum number of iterations is set to 100 and the mechanism used to classify instances and compute the classification accuracy in the course of training and testing is similar to what was used in [16].

Before the start of training, a threshold correlation value, C_t is chosen arbitrarily but this value should not be so low that the training process is stopped prematurely while weights are not well trained. In this paper a $C_t = 0.65$ is used. Pairwise interclass correlations of the desired spike trains representing the various classes in a dataset are computed using equation (2.7), from which an average interclass correlation metric C_{av} is computed using equation (2.12).

$$C_{av} = \frac{\sum_{l=1}^L \sum C_{(l,l-r)}}{N_p} \quad (2.12)$$

where $l = \{1, 2, 3, \dots, L\}$, $r = \{1, 2, 3, \dots, l - 1\}$, N_p is the number of all possible pairs of desired output spike trains, and L is the number of class labels in a dataset.

The correlation metric of the network's actual output spike train in response to input spikes for each instance in the training set and the desired spike trains of the various classes in the dataset are computed to form a vector, C_d for each instance. The second highest value in C_d is compared to C_{av} and the highest among the two is added to C_t to form a determinant, C_{det} . An instance under consideration is classified as belonging to the class with the highest value C_{max} in C_d , if and only if $C_{max} \geq C_{det}$. In a given iteration, all instances in the training set are classified using this mechanism and a confusion matrix is form, from which the training accuracy at the given iteration is computed. The training process is terminated at a given iteration if 100% training accuracy is obtained or 100 iterations are completed.

After each training process the trained weights and network parameters are saved and used for testing. In the testing phase, the weight update function is turned-off and the test instances are passed through the network and the mechanism used to classify training instances outlined above is followed to classify the test instances and the required classification performance metrics computed from the confusion matrix.

Sequential Learning Model

The main difference between the batch and sequential models is the mode in which the network weights are trained with respect to each instance in a training set. In the sequential model, instances are presented to a network one at a time and the network is trained using the input spikes of the single instance until it is able to correctly classify or the maximum number of iteration is completed before another instance is presented. For each instance, the weights are trained for several iterations until a correlation metric ($C = 1$) of the network's actual output spike train and the desired output spike train of the class the instance belongs is obtained and $C \geq C_{det}$ or the maximum number of iterations is completed. C_{det} is obtained using the same method as in the batch model. The weight update scheme defined by equations (2.5) and (2.6) are also used to update the weights at desired and undesired spike times.

In this model, the network is fully trained using the first instances of the various classes in a training dataset, and subsequent instances are first passed through the network without updating the weights and the correlation metric vector, C_d of the resulting actual output spike train and the desired spike trains of all class labels in the dataset is computed. If the correlation between the actual output and the desired spike train of the instance's class, $C \geq 0.90$ and $C \geq C_{det}$, it means that the network is able to correctly classify the instance and it is assumed that it is already in the knowledge-base of the network and training is not performed with the instance. Else, the network is trained until the afore mentioned condition is met or the maximum number of iterations is completed. In the event that the maximum iterations is completed first and there is a $C_{max} \in C_d$ where $C_{max} \geq C_{det}$, the instance is classified as belonging to the class that produced C_{max} . This training procedure is carried out for one instance at a time and is stopped after the last instance is learned.

In both batch and sequential learning models, it is anticipated that the conditional mode of learning where new knowledge is only added to the network when it is not able to produce an expected desired spike train with respect to a given training instance has a limiting factor that enable uniform learning from all classes irrespective of the nature and distribution of instances in a dataset. This is expected to increase the models ability to efficiently learn and classify instances in datasets with inherent challenges such as the class imbalance problem. Because majority of instances from the same class in a dataset are most likely to have highly correlated input spike trains, the magnitude of weight change required for learning reduces as the number of instances with highly correlated input spikes are encountered by the network to a point that marginal or no updates are made to the weights since they induce similar PSP in the network. This implies that at a point in learning from an imbalance dataset, the network would stop updating or only perform corrective updates to the weights with regards to the majority class instances much earlier. This phenomenon reduces learning interference thereby allowing the network to efficiently learn from the minority class instances.

2.2 Existing Classification Models

To assess the performance of the proposed SNN classification models relative to existing classification algorithms, their performance are compared to the Support Vector Machine (SVM), Multilayer Perceptron (MLP), Logistic Regression (LR), and IBk classification models. Classification experiments involving these models are done using their implementation in the Waikato Environment for Knowledge Analysis (WEKA) Machine Learning tool [17]. With the exception of the IBk model where the number of k-Nearest Neighbors is change to three (3) from the default of 1, all other models are simulated using their default parameters provided in the WEKA tool. Also, five (5) fold cross-validation process is used to train and test all the existing models in order to reduce bias and mitigate against over-fitting scenarios.

2.3 Classification Datasets

Six (6) benchmark datasets and real-world road condition data collected as part of a larger research work are used to test the performance of the classification models. Brief details of these dataset are provided below.

2.3.1 Benchmark Datasets

The benchmark datasets used include four binary datasets; the Glass2, Haberman, New-Thyroid1, and Vowel0 datasets, and two (2) three class datasets, which are the Balance and Fisher IRIS datasets. All these datasets are available at UCI Machine Learning repository [18]. With the exception of Fisher IRIS dataset which has equal number of instances in all classes, the others have imbalance distributions. The IRIS dataset is considered in this paper because it presents a high level complexity and is widely used to evaluate classification models. The other datasets are considered because of their imbalance nature, which provides us the opportunity to assess the performance of the proposed models on datasets with known challenges that affect efficient learning in existing machine learning algorithms. A summary of the datasets is provided in Table 1.

The models are trained and tested on the original datasets and also on oversampled versions of the datasets. The WEKA implementation of the Synthetic Minority Oversampling Technique (SMOTE) [19], is used to oversample minority class instances prior to learning. In the case of the Balance dataset, which is a three (3) class dataset, only instances of the class with the least number are oversampled. The oversampling rate for all datasets is varied from 100% to 400% at an interval of 100% depending on the class distribution and in such a way that the size of the oversampled class does not exceed any of the other classes. The performance of the models on the percentage that yields the best results in each dataset is reported.

Table 1. Summary of benchmark classification datasets

Dataset	Instances/Class(Per. (%))			Total
Glass2	17(7.94)	197(92.06)	-	214
Haberman	81(26.47)	225(73.53)	-	306
New-Thyroid1	35(16.28)	180(83.72)	-	215
Vowel0	90(9.11)	898(90.89)	-	988
Balance	288(46.08)	49(7.84)	288(46.08)	625
Fisher IRIS	50(33.33)	50(33.33)	50(33.34)	150

2.3.2 Road Condition Datasets

The road condition datasets were collected in Ghana as part of a larger research work that is aimed at developing an automated road condition classification system. The datasets were collected using an Android Application that captured triaxial acceleration of inbuilt smart phone accelerometers on-board some cars and tricycles [20]. Experiments conducted in [20], using the datasets confirmed that data from both cars and tricycles are suitable for road condition classification. The features as presented in the datasets are statistical measures computed from windows of amplitudes extracted from the time series triaxial acceleration data.

Table 2. Summary of road condition datasets

Activity	Condition Class	Instances/Class	Percentage (%)
Road Type	Paved	1717	60.59
	Unpaved	1117	39.41
	<i>Total</i>	<i>2834</i>	<i>100</i>
Unpaved Road	Normal	1117	75.22
	Anomaly	368	24.78
	<i>Total</i>	<i>1485</i>	<i>100</i>
Paved Road	Normal	1717	44.45
	Speed Ramp	319	8.25
	Pothole	1240	32.10
	Patches	587	15.20
	<i>Total</i>	<i>3863</i>	<i>100</i>

The datasets are for three different road condition classification tasks; road type classification, unpaved road anomaly classification, and paved road anomaly classification. The road type dataset is a binary class dataset that comprise of normal paved and unpaved roads surface data. The main task on road type classification is therefore to train models to classify the instances as belonging to paved or unpaved road. The unpaved road anomaly classification dataset is also a binary dataset containing normal unpaved road surface data and data representing anomalies on unpaved roads including ditches, speed ramps and other obstructions that have the tendency of causing damage to vehicles or discomfort to passengers. All anomalies on unpaved roads are modelled into one (1) class making the dataset binary. The paved roads condition dataset is modelled as a four (4) class problem consisting of speed ramps, potholes, patches, and normal road data. A summary of the datasets is presented in Table 2. Experiments involving these datasets are done using five (5) fold-cross validation process.

3 RESULTS AND DISCUSSION

The results presented in this section are divided into three (3) subsections. The first is on the evaluation of the proposed SNN learning model on spike sequence learning task; the second presents the performance of the modified versions of the learning models on selected benchmark classification tasks; and the last is an evaluation of the performance of the classification

models on road surface condition classification, using real-world road condition data collected by the authors via an Android Application.

3.1 Spike Sequence Learning

A series of spike sequence learning experiments are conducted in this section to assess the performance of the proposed learning rule dubbed SuperLSQ under varying learning periods and input-output spike patterns. First, the convergence analysis of SuperLSQ is presented for learning periods varied from $T = 100ms$ to $1000ms$ at an interval of $100ms$ with input neurons emitting multiple spikes at a mean frequency, $F_{in} = 5Hz$ and the output neuron also firing at a mean frequency, $F_{out} = 100Hz$. The effects of input-output firing rates on the performance of the proposed model is also assessed by varying the input firing rate from $5Hz$ to $50Hz$ at an interval of $20Hz$ and the output firing rate from $50Hz$ to $150Hz$ at an interval of $50Hz$. The performance of the proposed method is also compared to two (2) existing models derived for spike sequence learning. All correlation-based metric reported in this section are medians of twenty (20) repetitions of each experiment.

3.1.1 Effects of Learning Period

The median of the correlation-based metric, c_m for the first 150 training epochs of SuperLSQ, simulated using learning periods varied from $100ms$ to $500ms$ and $600ms$ to $1000ms$ are shown in Fig. 1 (a) and (b), respectively.

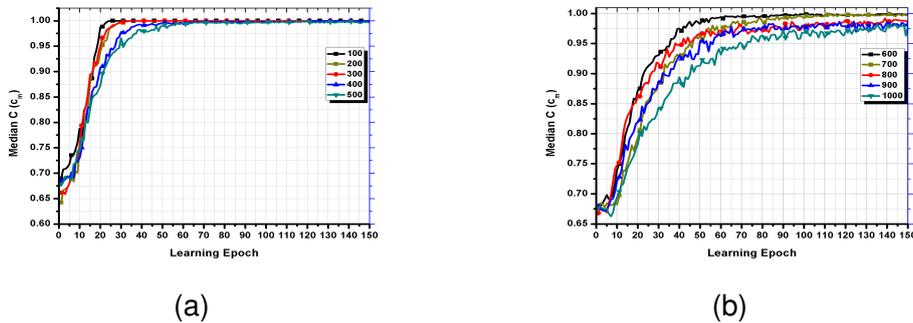


Fig. 1. Performance of SuperLSQ for (a) $T = 100ms - 500ms$ and (b) $T = 600ms - 1000ms$

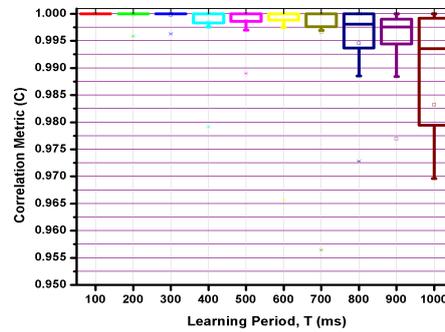


Fig. 2. Box Plot of Correlation Metric Measures of 20 trials for $T = 100ms - 1000ms$.

From Fig. 1(a), the proposed model, SuperLSQ, obtained the maximum value, $c_m = 1$ at epochs 23, 32, 33, and 120 respectively, for the learning periods $100ms$ to $400ms$. It obtained a median, $c_m = 0.9971$ for $500ms$ at the 150^{th} epoch and however, converged to $c_m = 1$ after 208 epochs when learning is allowed to continue. The median correlation metric c_m , for the learning periods from $600ms$ to $1000ms$ are shown in Fig. 1(b). As shown in Fig. 1(b), the proposed model could not attained $c_m = 1$ within the first 150 epochs for all learning periods from $600ms$ to $1000ms$. However, it converged to $c_m = 1$, after 152 and 317 epochs for the periods $600ms$ and $700ms$, respectively, when learning continues after the first 150 epochs. For the periods $800ms$, $900ms$, and $1000ms$, it attained maximum c_m values of 0.9939, 0.9910, and 0.9911 at epochs 591, 201, and 569 respectively, and hover around these values for the rest of the training epochs.

To buttress the convergence capability of the model within these periods, Box plots of the correlation metrics at the point of convergence for the 20 trials of each learning period is shown in Fig. 2. It is observed that for the learning periods $100ms$ to $300ms$, the method achieved maximum correlation metric of 1 for all the 20 trials, a minimum of 0.9972 within the periods $400ms$ to $700ms$, and a minimum and maximum of 0.9682 and 1 respectively at $1000ms$.

A key notable trend exhibited by the model is that the number of epochs required to converge increases with an increase in the learning period, which also comes with marginal decrease in median correlation metric. This trend is

attributed to an increase in the number of spikes in the desired output spike train that comes with increased learning periods. These results demonstrate that the proposed method though recorded marginal drop in performance with increased learning periods it can produce stable performance for any learning period within $100ms$ to $1000ms$.

3.1.2 Effects of Firing Rate

The effects of input-output firing rates on the proposed models is assessed and presented in this subsection. The input firing rate is varied from $5Hz$ to $50Hz$ at an interval of $20Hz$ and the output rate also varied from $50Hz$ to $150Hz$ at an interval of $50Hz$. The possible combinations of input-output rates are simulated using learning periods between $100ms$ and $1000ms$ increased at an interval of $400ms$. The results for different combinations of input-output firing rates and learning periods are shown in Fig. 3.

It is observed from Fig. 3, that when the network is simulated with $100ms$, all possible combinations of input-output firing rates considered produced median correlation metric, $c_m = 1$, except when the input and output firing rates are set to the maximum, $45Hz$ and $150Hz$ respectively, where, $c_m = 0.9706$ is recorded. When the learning period is increased to $500ms$, the input-output firing rate combinations 5/50, 5/100, 25/50, 25/100, and 45/50Hz all achieved $c_m = 1$, while the combinations with relatively higher rates, particularly in the output neuron; 5/150/, 25/150, 45/100 and 45/150 yielded 0.9995, 0.99067, 0.93773, and 0.94045

respectively. Further increasing the period to 900ms, $c_m = 1$ is only recorded when input-output firing rates are set to 25Hz and 50Hz respectively, with the minimum c_m of 0.65425

occurring at 25/150Hz followed by 45/150Hz with 0.84713 c_m measures. All other combinations of input-output rates and learning periods recorded c_m values above 0.9564.

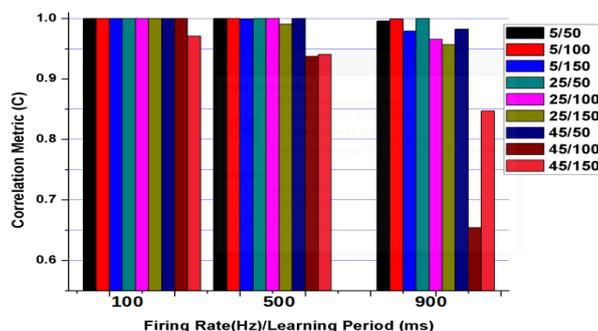


Fig. 3. Effects of firing rate/learning period on superLSQ

These results suggest that a combination of higher firing rates and learning periods do have a negative impact on the learning capability of the proposed model. This is however, expected, since increase in both results to an increase in the size of the spike trains which translates to an increase in the computational complexity of the network. However, the impact can be considered very minimal since over 92% of the cases yielded c_m values above 0.93 which marks a good learning performance.

3.1.3 Comparison with Existing Methods on Spike Sequence Learning

In this section we present and compare the experimental results of SuperLSQ with Delay Learning (DL) ReSuMe [2] and Extended Delay Learning (EDL) ReSuMe [3]. Fig. 4 shows the median correlation metric (c_m) of the three models for learning periods varied from 100ms to 1000ms at an interval of 100ms. From Fig. 4 all the models obtained c_m values of 1 for periods from 100ms to 500ms. For periods after 500ms, the performance of DL and EDL dropped continuously to 0.925 and 0.952 at 1000ms respectively, while the proposed model obtained c_m values of 1 at 600ms and 700ms and dropped marginally to 0.9967 at 1000ms. It is evident from these results that the proposed model is more stable than the DL and EDL at higher Learning periods.

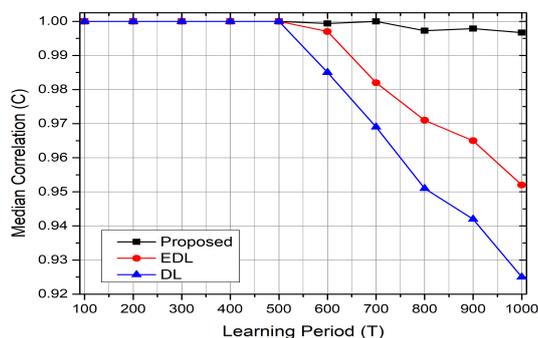


Fig. 4. Performance of superLSQ and resume for $T = 100 - 1000ms$

3.2 Evaluation on Benchmark Classification Datasets

The performance of the proposed SNN classification models dubbed normal SNN (nSNN) and sequential SNN (sSNN) vis-à-vis the performance of four existing classification algorithms including SVM, MLP, IBk, and LR on six (6) benchmark classification datasets is presented in this section. The mean F1-score and Recall of these models on the benchmark datasets are shown in Fig 5 (a) through (d) for the binary datasets and Figs 6 (a) and (b) for the two three class datasets. As shown in Table 1, four (4) of the datasets are binary with the class imbalance problem and the remaining two are three class datasets out of which the Balance dataset is also imbalance. Two sets of experiments are conducted on the imbalance datasets; first the original datasets are used to train and test the models in which the F1 and Recall values are labelled Normal.F1 and Normal.Recall respectively in the figures. Secondly, the SMOTE is applied to the datasets at percentages ranging from 100% to 400% at a 100% interval until both Recall and Precision as well as the F1-score of the models are maximised. The mean F1 and Recall values recorded in the later are labelled SMOTE.F1 and SMOTE.Recall in Figs 5 and 6. The F1-score and Recall values reported are means of twenty (20) trials of the models; this is done to minimise biasness that might be introduced by the random split of datasets for training and testing. We opt to report the Recall values because we want to demonstrate how the existing models record good F1-scores while wrongly classifying most or all minority instances as belonging to the majority class.

The performance of the models on the Glass2 dataset shown in Figure 5(a) shows that the proposed SNN models; nSNN and sSNN recorded higher F1 values of 0.964 and 0.935 respectively, and that of the existing models ranged between 0.868 and 0.882. While the F1 scores of the existing models appear to be relatively good, the SVM and LR failed to correctly classify all positive instances and hence recorded Recall values of 0.0 and the MLP and IBk only had 0.059 and 0.235 Recall values. The nSNN and sSNN models on the other

hand recorded Recall values of 0.933 and 0.895 indicating mean positive class detection rates of about 93.30% and 89.50% as compared to the almost 0.0% rates recorded by the existing models. However, when the minority class instances are oversampled using the SMOTE, the Recall values of the existing models MLP, LR, and IBk increased significantly to 0.871, 0.788, and 0.871 respectively, while the SVM maintained a 0.0 Recall value. The Recall values of the nSNN and sSNN varied marginally from 0.933 to 0.941 and 0.895 to 0.926 respectively on the oversampled Glass2 datasets. The F1 scores of all models, however, did not record significant improvement on the oversampled datasets.

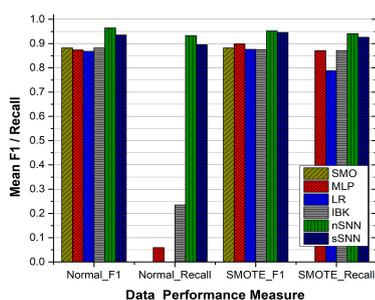
The performance of the models on the Haberman dataset is shown in Fig. 5 (b). Which shows that the existing models recorded F1 scores between 0.623 and 0.690 and Recall values of 0.0 in the case of the SVM model and 0.333 for MLP on the original dataset. The proposed nSNN and sSNN models however, recorded significantly higher F1 scores of 0.984 and 0.967 and Recall values of 0.923 and 0.931 respectively on the original dataset. Contrary to the significant increase in Recall values of the existing models in the Glass2 dataset after oversampling, the Recall values of the models only increased marginally from 0.0 to 0.412 for the SVM, 0.333 to 0.621 for the MLP, 0.148 to 0.547 for the LR, and 0.309 to 0.794 for the IBk model. The SVM, MLP, and LR, however, recorded marginal drop in F1 scores from 0.623 to 0.593, 0.735 to 0.650 and 0.683 to 0.644 respectively, while the IBk recorded a marginal increase from 0.69 to 0.709 after oversampling. The proposed nSNN and sSNN models on the other hand recorded F1 scores of 0.996 and 0.952 and Recall values of 0.977 and 0.916 on the oversampled dataset.

The performance of the models on the New-Thyroid1 and Vowels0 datasets are shown in Figs 5 (c) and (d) respectively. The results show that the MLP and the proposed models performed fairly well on both the original and oversampled datasets with F1 and Recall values above 0.962 and 0.958 on the New-Thyroid1 dataset and 0.933 and 0.917 on the Vowel0 dataset. The SVM however, recorded lower Recall values of 0.543 on the original New-Thyroid1 dataset and 0.622 and 0.833 on the original and oversampled

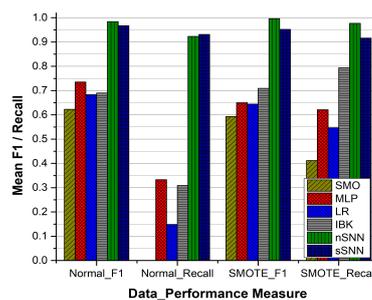
Vowel0 datasets. The IBK and LR also, recorded Recall values of 0.829 and 0.867 respectively, on the original New-Thyroid1 and vowel0 datasets. It is however, worth noting that, all the existing models obtained marginally better F1 and Recall values than the proposed SNN models on the oversampled New Thyroid dataset, in which the SVM and IBK recorded Recall of 1.0 and the MLP and IBK also obtained higher F1 and Recall of 1.0 on the Vowel0 dataset.

The performance of the models on the three class datasets are shown in Fig. 6 (a) and (b). As shown in Table 1, the Balance dataset is imbalance and as a result we trained and tested the models on the original dataset and also on an oversampled version of it where instances of the class with fewer instances were oversampled using SMOTE. With respect to the

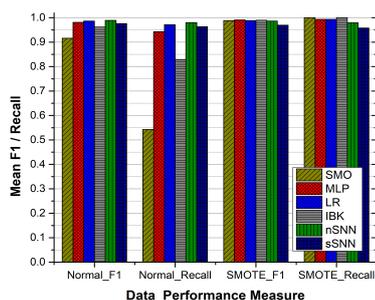
Balance dataset, the proposed nSNN and sSNN models obtained the best F1 and Recall values on the original dataset. nSNN recorded F1 and Recall of 0.976 and 0.942, while sSNN recorded 0.987 as F1 and 0.953 as Recall. Among the existing models, MLP recorded F1 and Recall of 0.913 and 0.714 followed by LR with 0.890 as F1 and 0.531 as Recall. IBk and SVM yielded 0.837 and 0.835 as F1 measures and however, failed to correctly classify all the minority class instances and thus recorded Recall of 0.0. Though the F1 scores of all models did not change significantly on the oversampled dataset, a good increase in Recall values were recorded. Notable among these are the increase in Recall from 0.0 to 0.821 and 0.932 for IBk and SVM. However, the Recall of the proposed models, nSNN and sSNN increased marginally from 0.942 to 0.967 and 0.953 to 0.984.



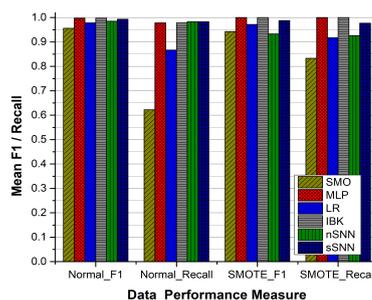
(a) Glass Dataset



(b) Haberman Dataset



(c) New-Thyroid1 Dataset



(d) Vowel0 Dataset

Fig. 5. Plot of mean F1 and recall of classification models on benchmark datasets

The performance of the models on the IRIS dataset is shown in Fig. 6 (b). Because the IRIS dataset is balanced the Recall values reported are the weighted averages over all classes in the dataset in which the values are identical to the F1 scores. It is observed from Fig. 6 (b) that all models obtained relatively good results with the IBk obtaining the minimum F1 score of 0.947. The proposed sSNN model however, recorded the maximum F1 score of 0.991 followed by nSNN with 0.970 and 0.967 for SVM.

From the performance of the models presented above, particularly, on the imbalance datasets, the performance of the proposed SNN models did not vary significantly on the original and oversampled versions of the datasets as compared to the existing classification models. This is more revealing on the Recall values of the models, which suggest that the existing models in the presence of imbalance data have a higher tendency of erroneously classifying positive class instances as belonging to the

negative class due to their rarity. This is however, not the case with the proposed models since the learning mechanisms employed in the models in a way limits the rate at which they learn from the majority class instances, thus, promoting proportional learning from both class. The good performance of the proposed models is further established on the oversampled datasets, which showed that as the existing models require alteration of the data distribution to obtain improved performance, which comes with increased requirements for computational resources in the case of oversampling. The performance of the proposed models are not influenced positively by modifying dataset distributions and thus the additional computational resources required to learn the additional data is avoided. The results obtained by the proposed models on the three (3) class datasets which are relatively complex to learn, further confirm that the models can produce relatively better results on multi-class datasets as compared to the existing models.

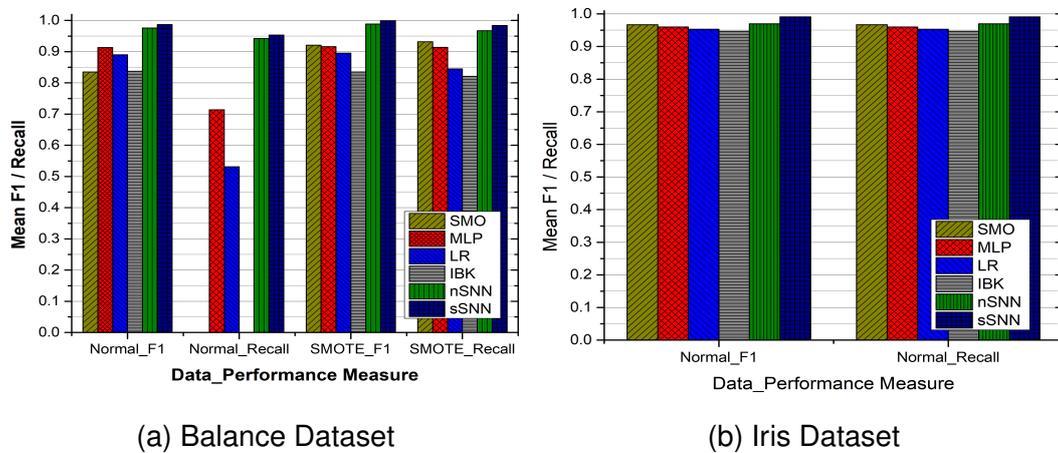


Fig. 6. Plot of mean F1 and recall of classification models on three class benchmark datasets

3.3 Evaluation on Road Condition Classification

The aim of this section is to conduct further assessment of the proposed models to establish their suitability in solving real-world classification tasks. First, the performance of the models on classification of anomalies on paved roads is presented followed by their performance on unpaved road anomaly classification and finally, on road type classification.

Table 3 shows the mean F1-scores of the proposed SNN (nSNN and sSNN) and the four existing classification algorithms (SVM, MLP, LR, and IBk) on paved road anomaly classification. Boldfaced numeric entries indicate highest F1-score and italics represents lowest score. As shown in the first three rows of Table 3, each anomaly type is paired with normal road condition data to form binary class problems and used to train and test the models mainly to determine their ability to discriminate between the anomaly types and normal road data. Results from these binary classification tasks show that the proposed models obtained higher F1-scores than all existing classification models. sSNN obtained the highest scores on the speed ramps and normal (Ramp vs. Normal) and patches and normal (Patch vs. Normal) pairs and the IBk yielded the least scores on both pairs. Also, the nSNN obtained the highest F1-scores on the potholes and normal pair with the LR classifiers producing the least scores. However, MLP and

SVM obtained relatively better scores on the patches and normal pair than the nSNN model.

The fourth row of Table 3 shows the performance of the models when data of the three (3) paved road anomalies considered are combined into one class and paired with normal road condition data for training and testing as a binary problem. Results for these experiments show that the nSNN model obtained the highest F1-score with the MLP performing marginally better than the sSNN model (0.9957 against 0.9947). The IBk, however recorded the least score of 0.9633.

The last experiments on the paved roads, involved a multi-class problem where data of the three anomaly types and normal road data are put together to form a four class problem. The mean F1-scores of the models on this dataset is presented as the last row of Table 3, where MLP recorded the highest performance followed by sSNN, SVM, and nSNN respectively, and the LR classifier obtained the least score.

Table 3. Mean F1-scores of models on paved road condition classification

Task	Testing Results (Mean F1-Scores)					
	nSNN	sSNN	SVM	MLP	LR	IBk
Ramp vs. Normal	0.9969	0.9979	0.9610	0.9848	0.9550	<i>0.9433</i>
Pothole vs. Normal	0.9966	0.9925	0.9538	0.9754	<i>0.9381</i>	0.9514
Patch vs. Normal	0.9632	0.9975	0.9727	0.9940	0.9630	<i>0.9518</i>
Anomalies vs. Normal	0.9994	0.9947	0.9864	0.9957	0.9700	<i>0.9633</i>
All 4 Classes	0.9625	0.9677	0.9641	0.9877	<i>0.9221</i>	0.9437

The performance of the models on unpaved road anomaly classification and road type classification are shown in Table 4. It is evident from Table 4 that the proposed models, nSNN and sSNN obtained higher F1-scores than all the existing models on both unpaved road anomaly classification and road type classification tasks. On unpaved road anomaly classification, the MLP model came third with an F1-score of 0.9900 and SVM obtained the least score, 0.9735. However, on the road type task, the LR model obtained the third highest score and the IBk obtained the least.

Table 4. Mean F1-scores of models on road type and unpaved road condition classification

Task	Testing Results (Mean F1-Scores)					
	nSNN	sSNN	SVM	MLP	LR	IBk
Road Type	0.9966	0.9998	0.9624	0.9753	0.9821	<i>0.9574</i>
Unpaved Road	0.9969	0.9964	<i>0.9735</i>	0.9900	0.9748	0.9881

The performance of the models on all categories of tasks; paved road and unpaved road anomaly classification and road type classification as presented, confirm that the proposed models generally performed better than the existing models especially the LR and IBk in all cases. The SVM and MLP performed comparably well particularly, on the multi-class datasets. This suggest that the proposed models are comparatively, better than the existing models on binary classification tasks and can achieve performance comparable to or better than SVM and MLP on multi-class tasks. This also, further establishes the suitability of the proposed SNN models for solving real-world classification task and thus, provides a foundation for further investigations to determine their suitability of integration into a mobile road surface anomaly detection and classification system that is currently been developed.

4 CONCLUSION

A learning model for spike sequence learning based on the least squares weight update scheme is first investigated in this paper followed by proposition of classification models for batch and sequential learning using the least squares weight update scheme.

Extended spike sequence learning task were conducted using different input-output spike firing rates under varying learning periods. The results from these experiments demonstrated that the model can produce stable performance under different scenarios within periods of 100ms and 1000ms. It recorded median correlation metric of 1 for the periods ranging from 100ms to 500ms after 150 learning iterations with its worse value of 0.9967 at 1000ms. The model also showed stable performance in terms of median correlation metrics than two existing models for spike sequence learning at periods from 600ms to 1000ms.

Classification experiments conducted with the proposed SNN models also showed that the models can generalise well to datasets with inherent challenges such as the class imbalance problem than some well-known classification algorithms such as the SVM, MLP, LR, and IBk. Results from some benchmark datasets showed

that the proposed models obtained a minimum F1 and Recall values above 0.9 on the original imbalance datasets while some of the well-known machine learning algorithms recorded 0.0 Recall. Classification results of the models on real-world road condition datasets also revealed that the proposed models produced results that are better than most of the existing classification algorithms.

Though the single layer models produced comparably better results than some existing algorithms, it would be interesting to extend and investigate the performance of the weight update scheme on deeper networks.

COMPETING INTEREST

The Authors have no competing interest.

REFERENCES

- [1] Ponulak F, Kasiński A. Supervised learning in spiking neural networks with ReSuMe: Sequence learning, classification, and spike shifting. *Neural Computation*. 2010;22(2):467–510.
- [2] Taherkhani A, Belatreche A, Li Y, Maguire LP. DL-ReSuMe: A delay learning-based remote supervised method for spiking neurons. *IEEE Transactions on Neural Networks and Learning Systems*. 2015;26(12):3137–3149.
- [3] Taherkhani A, Belatreche A, Li Y, Maguire L. EDL: An Extended Delay Learning Based Remote Supervised Method for Spiking Neurons. *Neural Information Processing. Lecture Notes in Computer Science*. 2015;9490(11):190–197.
- [4] Bohte SM, Kok JN, La Poutre H. Error-backpropagation in temporally encoded networks of spiking neurons. *Neurocomputing*. 2002;48(1-4):17–37.
- [5] Ghosh-Dastidar S, Adeli H. A new supervised learning algorithm for multiple spiking neural networks with application in epilepsy and seizure detection. *Neural Networks*. 2009;22(10):1419–1431.
- [6] Xu Y, Zeng X, Han L, Yang J. A supervised multi-spike learning algorithm based on gradient descent for spiking neural

- networks. *Neural networks*. 2013;43C:99–113.
- [7] Taherkhani A, Belatreche A, Li Y, Maguire LP. A supervised learning algorithm for learning precise timing of multiple spikes in multilayer spiking neural networks. *IEEE Transactions on Neural Networks and Learning Systems*. 2018;29(11):5394–5407.
- [8] Widrow B, Hoff EM. Adaptive Switching Circuits. 1960;4:96–104.
- [9] Booi O, Tat Nguyen H. A gradient descent rule for spiking neurons emitting multiple spikes. *Information Processing Letters*. 2005;95(6):552–558.
- [10] Belatreche A, Maguire LP, McGinnity M, Wu QX. Evolutionary Design of Spiking Neural Networks. *New Mathematics and Natural Computation*. 2006;02(03):237–253.
- [11] Silva M, Vellasco MM, Cataldo E. Evolving Spiking Neural Networks for Recognition of Aged Voices. *Journal of Voice*. 2017;31(1):24–33.
- [12] Lobo JL, Laña I, Del Ser J, Bilbao MN, Kasabov N. Evolving Spiking Neural Networks for online learning over drifting data streams. *Neural Networks*. 2018;108:1–19.
- [13] Sporea I, Grüning A. Supervised learning in multilayer spiking neural networks. *Neural Computation*. 2013;25(2):473–509.
- [14] Tavanaei A, Ghodrati M, Kheradpisheh SR, Masquelier T, Maida A. Deep learning in spiking neural networks. *Neural Networks*; 2018.
- [15] Agebure MA, Oyetunji EO, Baagyere EY. A supervised method for learning precise timing of multiple spikes in spiking neural networks. 2019 IEEE AFRICON; 2019:1–7.
- [16] Yellakuor BE, Moses AA, Zhen Q, Olaosebikan OE, Qin Z. A multi-spiking neural network learning model for data classification. *IEEE Access*. 2020;8(72):360–372.
- [17] Frank E, Hall MA, Witten IH. The WEKA Workbench. Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques". 4th ed. Morgan Kaufmann. 2016.
- [18] Dua D, Graff C. UCI machine learning repository; 2017. [Online]. Available:<http://archive.ics.uci.edu/ml>
- [19] Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP. SMOTE: Synthetic minority over-sampling technique. *J. Artif. Int. Res*. 2002;16(1):321–357.
- [20] Agebure MA, Oyetunji EO, Baagyere EY. A three-tier road condition classification system using a spiking neural network model. *Journal of King Saud University - Computer and Information Sciences*; 2020.

© 2020 Agebure et al.; This is an Open Access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>) which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Peer-review history:

The peer review history for this paper can be accessed here:
<http://www.sdiarticle4.com/review-history/62867>