



Evolving Spiking Neural Network as a Classifier: An Experimental Review

M. Saravanan¹(✉), Annushree Bablani², and Navyasai Rangisetty²

¹ Ericsson India Global Services Pvt. Ltd., Perungudi, India
m.saravanan@ericsson.com

² Indian Institute of Information Technology, Sri City, India
{annushree.bablani,navyasai.r17}@iiits.in

Abstract. The brain-inspired Spiking Neural Networks (SNNs) are considered as the third generation of neural networks for AI applications. The spiking neural network has been proved very efficient to predict and classify data with spatial and temporal information in the way of modeling the behavior and learning potential of the brain. The aim is to understand the working of the Evolving SNN (ESNN) as a classifier and how it is different from the existing neural models. Besides exploring the existing ESNN architecture, the results have been generated by tuning the various parameters of the ESNN model which may be contributing to provide better prediction accuracies. The tuned ESNN model is applied to various datasets and compared with the existing second-generation neural network model like LSTM. The results show comparable improvement in the classification accuracy using ESNN which concludes that the ESNN and its variants are the beginning of a new era of Neural Networks.

Keywords: Evolving spiking neural network · Reservoir · Gaussian receptive field · Spatial-temporal data

1 Introduction

In the past few years, Neural Networks have evolved a great deal in solving a lot of computationally complex problems. Being a decade-old algorithm with such flexibility to evolve into various generations, the increasing complexity of the problem has made neural networks, one of the most used algorithms by researchers. With this emergence, a new type of model, SNN has been introduced [1], which is computationally more powerful than the previous generations. In the conventional Artificial Neural Network, the neurons are continuous data variables, here in SNN (aka, Third Generation Neural Network), the neurons receive and transmit information in the form of spike trains (inspired from the biological brain).

One of the SNN architectures, called Evolving Spiking Neural Network, has been introduced in [2] which makes SNN more adaptive and faster by incrementally merging the evoked neurons to capture the pattern in the given problem. ESNN takes this property from Evolving Connectionist Systems (ECOS) [3], where the new neurons, new features,

new connections are generated during the execution of the system. This is a dynamic model, which adapts or evolves with time. In this paper, we are trying to explore the evolving nature of ESNN as a classifier using three case study applications to understand its effectiveness. Because of its dynamic and evolving nature, ESNN can capture data as and when available without the requirement of retraining the network. The ESNN learns the mapping from signal data vector to specified class label and hence this makes it a unique classifier.

To improve the performance of models on spatiotemporal data, an extension to ESNN was proposed in [4] called Recurrent ESNN, which follows the principle of probabilistic Liquid State Machine, by adding a layer to the architecture. This additional layer acts as the reservoir which transforms the input pattern into a single high-dimensional network. The high-dimensional network then can be trained using ESNN. The ESNN model is illustrated in this paper and applied on three different applications: 1) Mobility Dataset 2) 5G Data and 3) EEG data for emotion Recognition and these datasets exhibit spatiotemporal features as a common factor. The reason for choosing two communication datasets is because of their dynamic behavior and drift caused during various times and locations. There is no specific pattern in these changes. Also, we have used EEG data which is biological and highly dynamic data. In EEG data there underlies a pattern but this pattern changes from human to human, like communication data which changes with the type of user. The aim is to produce a more accurate, energy-efficient, and fast learning system to achieve better classification results on spatiotemporal datasets. To achieve that instead of learning from all the fired neurons we have selected the best k , neurons as the input to the ESNN. The detailed methodology and results achieved are explained in the rest of the paper.

2 Related Works

Researchers have applied ESNN to achieve better results for the limited size neuron repository [5]. The model presented in [5] classifies input after one presentation. These models are good for online learning, where the input changes dynamically and the model can classify the data without the requirement of retraining. The model followed a sliding window approach where the input samples within a particular window were encoded. In another work, the classification performance of ESNN was improved by selecting an optimal set of features using the wrapper and quantum-inspired evolutionary optimization approaches [7]. Using this combination relevant features were identified and finally, an optimal parameter setting was achieved giving better classification results. Parameter optimization is a very crucial part of any kind of network model and to achieve that like previous work using a quantum inspired SNN with Particle Swarm Optimization (PSO) approach was proposed [8]. This quantum inspired PSO (QiPSO) was designed for search in binary spaces, and it has shown comparable results to the formal QiSNN. Later Dynamic QiPSO [9], was proposed where the search strategy was performed in both binary and continuous search spaces.

ESNN has been implemented by several researchers in a variety of applications. The ESNN classifier has been used for Spatial and Spectro-temporal pattern recognition problems. Various models with a single layer or with a layer of the reservoir have been

applied. In [9, 10] dynamic updates in the input neurons are added to the conventional Leaky Integrate-and-Fire (LIF) Model and updating of the weights was not only dependent on rank order learning, but on the time of the following incoming spikes to the post-synaptic neuron. This model was referred to as Dynamic ESNN, and they used a single layer of neurons [9,10]. When the size of the input neuron increases (temporal pattern), a single layer may not be sufficient and efficient to learn these patterns. So, a reservoir based ESNN has been proposed in [6]. Initially, the input is converted to a spike train using an encoding scheme and then these input spikes are passed through a filter to collect the temporal features. This filter acts as a liquid state or reservoir and ESNN acts as the readout or layer. The reservoir is constructed of LIF neurons with exponential synaptic currents. Here, the RESNN has been utilized to model Spatial-temporal patterns. The study used the real-world sign-language dataset LIBRAS on this model and it achieved encouraging classification results [6].

ESNN and its improved architectures have been used in applications like image recognition [11–13], speaker authentication system [14], audio-visual pattern recognition system [15], taste recognition [16, 17], sign language [6], object recognition [9], EEG pattern analysis [10] and many others. In this paper, we have attempted to understand how ESNN works as a classification approach and compared it with the earlier NN approaches. As ESNN has shown promising performance in earlier studies also, it motivated us to try to propose an ESNN by selecting the best-fired neurons and applying the model on three different use cases. Here, additionally we tried to tune the ESNN parameters suitably to get a best performance from the classifier. Detailed methodology is explained in the subsequent section.

3 Evolving Methodology

The data classification is based on prior knowledge or statistical information extracted from the given data. The data is usually classified as measurements or observations, defining points in an appropriate multidimensional space. Classification is dependent on the type of learning procedure that generates the output value based on template matching, statistical classification, syntactic or structural matching, and neural networks.

ESNN being a type of neural network classifies the data by creating spikes that contains the temporal information of data and hence are very powerful in discriminating the information. The implementation of the ESNN initially begins by creating an empty neuron repository and a new output neuron is generated and added to the empty repository. For every input sample, the numerical data is converted into a trail of spikes, this is called *encoding*. To perform encoding on numerical data, *Gaussian Receptive Field (GRF) population* [5] encoding scheme has been used. Every feature input is distributed over several neurons (called Gaussian Receptive Field Neurons or GRFN) and each of these neurons is fired only once during the time interval (T). After encoding each input feature is represented in the form spatiotemporal spike pattern. The center μ_j and the width σ_j of each GRF presynaptic neuron are computed as [5]:

$$\mu_j = I_{min}^n + \frac{2j - 3}{2} \left(\frac{I_{max}^n - I_{min}^n}{N - 2} \right) \quad (1)$$

$$\sigma_j = \frac{1}{\beta} \left(\frac{I_{max}^n - I_{min}^n}{N - 2} \right) \quad (2)$$

Here I_{max}^n and I_{min}^n are maximum and minimum values of n^{th} feature in given window size, N is the number of receptive fields (GRF neurons for each feature), β is a parameter in [1, 2]. The output of neuron j is defined as

$$out_j = \exp\left(\frac{(x - \mu_j)^2}{2\sigma_j^2}\right) \quad (3)$$

where x is the input value. The firing time of each presynaptic neuron j is defined as $T_j = T(1 - out_j)$, where T is the simulation time or spike interval.

The advantage of using the GRF encoding scheme over the mostly used Poisson encoding scheme is the time taken for encoding. In Poisson encoding, the encoder waits for all spikes to get fired and then encodes it, whereas in GRF each spike as it is fired is encoded. GRF does not wait for a full spike train, hence reducing the time of encoding in the model.

Here LIF model [1] is used to create the initial neurons. Each neuron fires at most once and a neuron fire only when its Postsynaptic Potential (PSP) reaches its threshold value. PSP of the i^{th} neuron is defined as:

$$PSP_i = \begin{cases} 0. & \text{if fired} \\ \sum_j w_{ji} \cdot mod^{order(j)}. & \text{Otherwise} \end{cases} \quad (4)$$

where w_{ji} represents the weight of the synaptic connection between presynaptic neuron j to output neuron i , mod is the modulation factor in [0, 1], and $order(j)$ defines the rank of the presynaptic neurons spike. The first rank is assigned as 0 and subsequently, the rank is increased by 1 based on the firing time of each presynaptic neuron.

Firstly, the model creates an empty repository for output neurons. For each pattern that belongs to the same given class, a new output neuron is created and connected to all presynaptic neurons in the previous layer, and weights are assigned using rank order. The weight w_{ji} is calculated as

$$w_{ji} = mod^{order(j)} \quad (5)$$

A numerical threshold γ_i is set for the newly created output neuron as the fraction c in (0, 1) of its maximum postsynaptic potential $PSP_{max,i}$ as

$$\gamma_i = PSP_{max,i} \cdot c \quad (6)$$

The weight vector of a newly created output neuron is then compared with the already present output neurons in the repository. If the Euclidean distance between the newly created output neuron weight vector and that of any of the already trained output neurons is smaller than a similarity parameter (SIM), they are considered to be similar. As a result, their thresholds and weight vectors are merged according to

$$w_{ji} = \frac{w_{new} + (w_{ji} \cdot M)}{M + 1} \quad (7)$$

$$\gamma_{ji} = \frac{\gamma_{new} + (\gamma_{ji} \cdot M)}{M + 1} \tag{8}$$

where M is the number of previous merges of similar neurons through the learning history of the ESNN. After merging, the weight vector of the newly created output neuron is discarded, and the new pattern is presented to the model. If none of the already trained neurons in the repository is found to be similar (as per the SIM parameter) to the newly produced output neuron, then it is added to the repository.

The testing phase is carried out by propagating the spikes that encode the test sample to all trained output neurons. The class label for the test sample is assigned according to the class label of the output neuron which has fired first after reaching its threshold value γ_i . The methodology explained above is given as a flow diagram in Fig. 1.

In this paper, the existing methodology is improved by tuning the ESNN parameters c , mod , and SIM . The parameter c determines that how many fractions of PSP will be used as the threshold value. The parameter mod is the modulation factor, and it determines the importance of the order of the first spike, and the parameter SIM , which measures the similarity between two neurons, and on basis of SIM , a neuron is either added to a repository or is replaced the already available similar neuron in a repository.

The synapses are dynamic; their values change over the timescale of training. To have that variability in the ESNN model the parameters c , mod , and SIM are utilized. ESNN model unlike any other neural network is quite sensitive to its parameters and they play a major role in defining the accuracy of the model. Determining an optimal set of parameters is crucial and challenging hence an attempt has been made to tune these parameters with various datasets and results are presented in the next section. All these variables lie in the range of (0,1) and varying them and tuning according to the model will improve the efficiency.

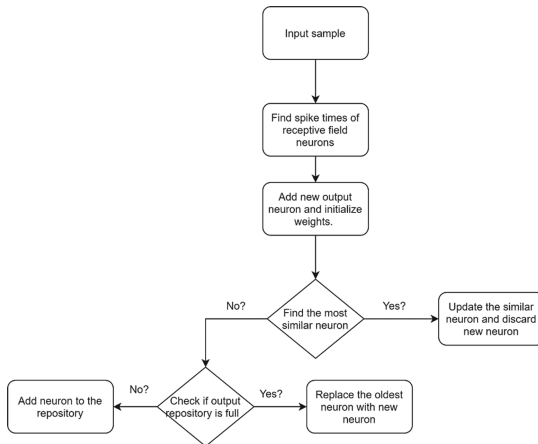


Fig. 1. ESNN workflow diagram

4 Dataset Description

Here, we first implement the ESNN for the classification tasks and to understand how the parameter tuning in ESNN can impact three different types of communication datasets, which will be analyzed. The motivation to use these three datasets is that the datasets are represented using their temporal information and ESNN's spikes manage the temporal information efficiently, which is expected to generate a better classification.

The first dataset is the telecom dataset [18] with 42 eNode B's and 10,000 UE's (unique id for each user mobile) for 5000 min. The following multiple mobility patterns were simulated: 1) Work Professional: Person travels from fixed home location to office location and comes back to home. 2) Sales Professional: Person travels from fixed home location and travels to 10 random locations. 3) Random waypoint: Person travels from random home location to random destinations until the day finishes off. In the simulator, one step corresponds to one minute, and also to introduce concept drift we switched off 8 of the eNode B's at a time of 3000 min and woke them up after 4000 min. For further details of the dataset, the reader is suggested to go through [18]. The second dataset utilized is a 5G dataset [19, 20] which identifies the incoming connection request and assigns it to the most optimal slice based on important KPIs. These KPIs can be captured from control packets between the UE and network. It has multiple types of input devices which include smartphones, general IoT, AR-VR, Industry 4.0 traffic, e911 or public safety communication, healthcare, smart city, or smart homes traffic, etc. It can even capture an unknown device requesting access to one or multiple services. Here, they have UE category values defined to them and the network also allocates a pre-defined QoS class identifier (QCI) value to each service request. In 5G, the packet delay budget and the packet loss rate are an integral part of the 5QI (5G QoS Identifier). DeepSlice will also observe what time and day of the week the request is received in the system [19].

To understand the variability in the results the third dataset utilized is a multimodal electroencephalogram (EEG) data set for the analysis of human affective states [21]. The EEG and peripheral physiological signals of 32 participants were recorded as each watched 40 one-minute-long excerpts of music videos. Participants rated each video in terms of the levels of arousal, valence, like/dislike, dominance, and familiarity. All three datasets were included for an experimental study to perform classification based on our proposed technique.

5 Results

The experimental results were performed using Intel(R) Xeon(R) CPU Model Name with 2.30 GHz frequency having 2 number of CPU Cores. The CPU Family Haswell with RAM 12 GB (upgrade to 26.75 GB) has been utilized for the implementation of the model on the three datasets (Sect. 4). From Eqs. 4 to 6, updating the weight and adding the right neuron in the repository using ESNN depends on parameters like c , mod , and SIM considered in the study.

To understand the best of the model, we performed various experiments. An attempt has been made to tune these parameters such that the accuracy of the model is improved.

We have applied the model and tuned these parameters c , mod , and SIM for the telecom dataset. The results of the same are given in Fig. 2. The variations in the accuracy with the varied value of parameters are shown. We can see that from Fig. 2(a), if the mod has lower values ranging in between 0.3 to 0.4 it gives good accuracy and provides comparable accuracy with values ranging between 0.6 to 0.7. This suggests that the modulation factors involved in updating the weight cannot be very less or exceedingly high. A moderate modulating factor achieves a weight that finally improves the network and achieves better accuracy. If we look at the graph of parameter c in Fig. 2(b), we can conclude that if the fraction lies in the range of 0.5 to 0.8 will be proportional to the threshold applied at the output neuron that will give better results. For Fig. 2(c) we can specify that the parameter SIM which tells us about the similarity between the two output neurons generated, should be less, less the similarity, better the learning, more accurately we can classify the data.

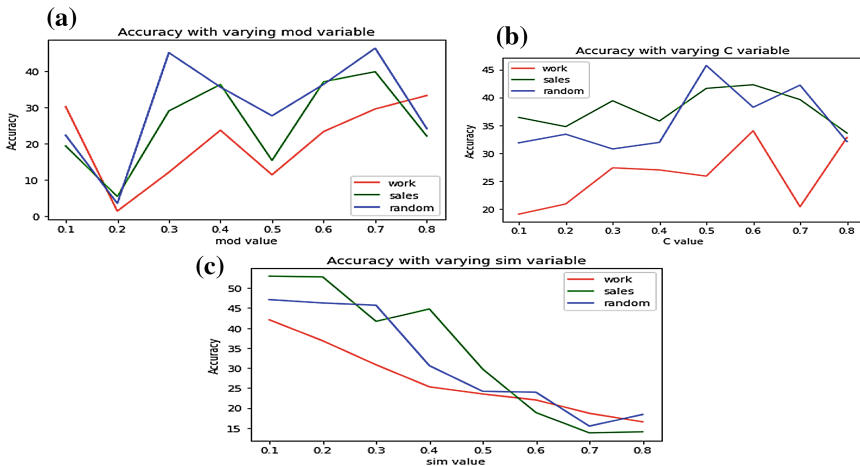


Fig. 2. ESNN Parameters mod , c , SIM in range (0,1) are tuned. (a) Accuracy with varying mod . (b) Accuracy with varying C . (c) Accuracy with varying SIM

The ESNN model results have been compared with Long Short-Term Memory (LSTM) [22] and the conventional SNN to demonstrate the improvement in the performance of the system at various levels using different parameters. The LSTM also belongs to the family of Neural Network, where instead of having only forward connections it has feedback connections. The LSTM has been developed to be used on time series data and to solve the problem of vanishing gradient descent. In LSTM, each block is made up of memory cells with input, forget, and output gates. The gates allow the memory cells to store and access information for longer periods to improve performance. The training time taken by the telecom dataset to perform LSTM is 208.57 s for the work-professional dataset, 420.52 for the sales-professional dataset, and 323.75 s for the random waypoint. Whereas, the ESNN is much faster taking 12.88 s for the work-professional dataset, 27.43 for the sales-professional dataset, and 25.09 s for the random waypoint. ESNN takes comparatively very less time than LSTM to complete

the training. LSTM requires more time due to its architecture, whereas ESNN completes this task faster. Not only faster training times, but also better accuracy on the dataset has been achieved using ESNN.

A comparison of the accuracy of the ESNN model with the LSTM model is given in Table 1. Not only training and the testing of the similar variant of mobility dataset, but we have also experimented with cross-training and testing within the mobility dataset, to understand the working of the model. For example, the ESNN model is trained with work professional telecom data and tested on sales professional telecom data and vice versa. The aim is to reach a single universal model which can perform well on similar datasets collected at or for different scenarios.

Table 1. Accuracy of the ESNN model, compared with LSTM.

Training	Dataset	work-professional	Sales-professional	random waypoint
Testing with ESNN	Work-professional	99.31	73.42	73.09
	Sales-professional	99.25	99.95	99.63
	random waypoint	99.27	99.97	99.97
Testing with LSTM	Work-professional	43.95	8.52	9.23
	sales-professional	35.56	65.34	63.75
	random waypoint	34.53	65.81	66.14

A comparative analysis has also been performed by considering the best 5 or top 5 results while testing. In the telecom dataset, there is a possibility that if more than one service devices are present in that location then our mobile can connect to any of the service devices. Since the output can be any of the service devices, we need to consider more than one output neuron, so we are using *top k* to overcome this problem. Figure 3 shows the results achieved using various combinations of training and testing models. A comparison is performed using LSTM and ESNN approaches. While testing using LSTM, a prediction on the probability of each class using ‘model.predict_proba’ is made and we have considered top-5 classes (classes with top-5 probabilities) to check for true class. If the true class is present in these 5 classes, then the prediction is considered true. However, while testing with ESNN we considered the first 5 neurons that are fired (whose PSP value reached threshold value γ_i). If a true is present in any of the classes of these 5 neurons, our prediction is true. More experimental analysis has been performed on the telecom dataset. As shown in Fig. 3, the y-axis represents the accuracy achieved by each model. Here,

- Model-1 represents training with work-professional data and testing on work-professional data.
- Model-2 represents training with work-professional data and testing on sales-professional data.

- Model-3 represents training with work-professional data and testing on random-waypoint data.
- Model-4 represents training with sales-professional data and testing on work-professional data.
- Model-5 training with sales-professional data and testing on sales-professional data.
- Model-6 represents training with sales-professional data and testing on random-waypoint data.

From Fig. 3 we can conclude that using top-5 max fired PSP neurons for testing using ESNN (in blue line) achieves the highest results among all types of models. Considering top-5 LSTM (in yellow line) probabilities for testing represents a comparable result for all types of models. While comparing between various models, model -5 and model-6 give us good accuracy results. This suggests that when trained with one type of data and tested with the same data gives us better accuracy.

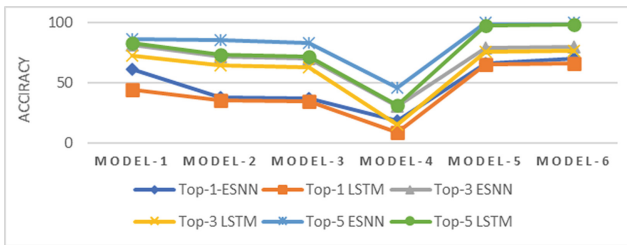


Fig. 3. Accuracy achieved on different combinations of the model whilst considering the testing on top-k probabilities.

However, model-6 where training is performed on sales professional data and tested on random waypoint has also achieved better accuracy. When you compare it with the result of model-3, model-6 shows promising results that is because the sales professional data has data of various locations a salesman visits whereas in the work professional dataset a specific set of locations are only captured. When these datasets are tested with a random waypoint where locations are random hence the model trained on random locations of sales professionals achieves better results.

We have also performed experiments and calculated the accuracy of the model by initially training it on sales professional data and testing it on work professional data. Then we add a few samples from work professional data to sales professional data and trained it again and tested on work professional. We keep on adding the samples of work professionals and train the model and then test. The results achieved are shown in Fig. 4(a) for both LSTM and ESNN models. A similar experiment is performed by adding samples of work professional dataset on random waypoint dataset and the results are shown in Fig. 4(b). In Fig. 4, the y-axis gives the accuracy achieved and the x-axis represents the number of work professional datasets added while training the model. An attempt is made to compare the improved ESNN model with the state-of-the-art neural network and it is shown in Table 2. On comparing ANN, SNN, and RESNN with our improved ESNN, it is explicit that the ESNN outperforms all other approaches. The experiment results have

also been computed by training the 5G data (second dataset) using LSTM an accuracy of 96.14% has been achieved in 289.05 s, training with RESNN an accuracy of 82.89% has been achieved with a training time of 588.31 s. Whereas on using the ESNN model, an accuracy of 100% is achieved with a training time of 89.35 s.

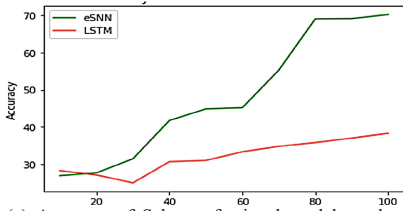
Table 2. Accuracy and training time for ANN (top-5), SNN (top-5), ESNN and RESNN when training on work-professional and testing on work-professional dataset.

Model (top-5)	Training time	Accuracy
ANN	208.57	83.27
SNN	222.55	83.87
ESNN	12.88	99.31
RESNN	80.28	99.05

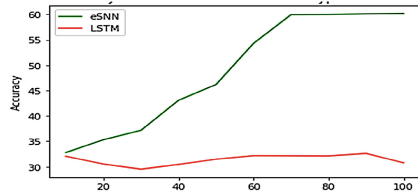
We can compare and say that ESNN has not only shown efficient performance in terms of accuracy but also in terms of training time. We have also used the spatiotemporal data i.e., EEG data for the implementation in this paper. The EEG data used is recorded while presenting a set of music video stimuli to the users [21]. EEG data were recorded from 32 channels or electrodes placed on the scalp following the 10–20 international placement protocol. The total recording was done for 2 min where the stimuli were presented to the user on a PC screen. Before performing experiments, the participants were trained and consented [21]. The video presented to participants were designed in such a way that they induce different types of emotions in the participants’ brains and then can be captured by the EEG. These recorded signals after being downsampled to 128 Hz, are utilized for implementing ESNN. We have compared the results with LSTM, where LSTM achieved an accuracy of 57.81 with a training time of 295.81 s and ESNN has achieved an accuracy of 99.6 but has taken more time of EEG data of 1234.16 s due to large data samples and multiple features of EEG data during training. On analyzing the three datasets it was found that the proposed ESNN model with tuned parameters and top-5 fired neurons achieve better accuracy for all three datasets as compared to LSTM.

6 Conclusion

The paper presented a detailed experimental review of Evolving Spiking Neural Networks as a unique classifier. We tried to understand how ESNN architecture behaves with different kinds of inputs. The ESNN architecture was fed with three different datasets: one mobility data having locations and time details of various professionals, a 5G network slicing data, and EEG data recorded for emotion analysis for 32 users. We had also performed experimental analysis by tuning *c*, *mod*, and *SIM* parameters of the ESNN model. After using the tuned parameters, it was found that the ESNN as a classifier gives more accurate results. The ESNN results have also been compared with the most used recurrent neural network i.e., LSTM. We had also used the GRF encoding scheme



(a) Accuracy of Sales-professional model on the work-professional data by adding samples from the work-professional data. The Sales professional model (model trained on sales professional dataset) is tested on work-professional dataset by adding samples from work-professional dataset to check which model can learn in a smaller number of epochs.



(b) Accuracy of the random waypoint model on work-professional data by adding samples from work-professional data. The random-waypoint professional model (model trained on random-waypoint dataset) is tested on random waypoint dataset by adding samples from random waypoint dataset to check which model can learn in a smaller number of epochs.

Fig. 4. (a) and (b) Accuracy of ESNN model on various combinations of the telecom datasets for the small number of samples and epochs.

which has significantly improved the time of the model. When compared to RESNN, ESNN takes very less time because of the simple spike generation process, whereas RESNN has a complex reservoir structure that requires more computational time. With these improvements and comparative analysis, we concluded ESNN, the third-generation neural network is going to bring the dawn in classification algorithms. With more experiments and improvements one can achieve better and efficient performance using ESNN architecture for different tasks.

References

1. Maass, W.: Networks of spiking neurons: the third generation of neural network models. *Neural Netw.* **10**(9), 1659–1671 (1997)
2. Kasabov, N.K.: Evolving spiking neural networks. In: *Time-space, spiking neural networks and brain-inspired artificial intelligence*, pp. 169–199. Springer, Heidelberg (2019)
3. Kasabov, N.K.: *ECOS: Evolving Connectionist Systems and the ECO Learning Paradigm*. Iconip, vol. 98 (1998). s
4. Schliebs, S., Nuntalid, N., Kasabov, N.: Towards spatio-temporal pattern recognition using evolving spiking neural networks. In: Wong, K.W., Mendis, B.S.U., Bouzerdoum, A. (eds.) *ICONIP 2010*. LNCS, vol. 6443, pp. 163–170. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-17537-4_21
5. Lobo, J.L., Laña, I., Del Ser, J., Bilbao, M.N., Kasabov, N.: Evolving spiking neural networks for online learning over drifting data streams. *Neural Netw.* **108**, 1–19 (2018)
6. Schliebs, S., Hamed, H.N.A., Kasabov, N.: Reservoir-based evolving spiking neural network for spatio-temporal pattern recognition. In the *International Conference on Neural Information Processing*, pp. 160–168. Springer, Heidelberg. https://doi.org/10.1007/978-3-642-24958-7_19
7. Schliebs, S., Defoin-Platel, M., Kasabov, N.: Integrated feature and parameter optimization for an evolving spiking neural network. In: Köppen, M., Kasabov, N., Coghill, G. (eds.) *ICONIP 2008*. LNCS, vol. 5506, pp. 1229–1236. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-02490-0_149

8. Hamed, H.N.A., Kasabov, N., Shamsuddin, S.M.: Integrated feature selection and parameter optimization for evolving spiking neural networks using quantum inspired particle swarm optimization. In: 2009 International Conference of Soft Computing and Pattern Recognition, pp. 695–698. IEEE, December 2009
9. Kasabov, N., Dhoble, K., Nuntalid, N., Indiveri, G.: On-line spatio-and spectro-temporal pattern recognition with evolving spiking neural networks utilizing integrated rank order-and spike-time learning. *Neural Networks* (2011)
10. Dhoble, K.: Spatio-/spectro-temporal pattern recognition using evolving probabilistic spiking neural networks (Doctoral dissertation, Auckland University of Technology) (2013)
11. Wysoski, S.G., Benuskova, L., Kasabov, N.: On-line learning with structural adaptation in a network of spiking neurons for visual pattern recognition. In: Kollias, S.D., Stafylopatis, A., Duch, W., Oja, E. (eds.) ICANN 2006. LNCS, vol. 4131, pp. 61–70. Springer, Heidelberg (2006). https://doi.org/10.1007/11840817_7
12. Wysoski, S.G., Benuskova, L., Kasabov, N.: Fast and adaptive network of spiking neurons for multi-view visual pattern recognition. *Neurocomputing* **71**(13–15), 2563–2575 (2008)
13. Wysoski, S.G., Benuskova, L., Kasabov, N.: Adaptive learning procedure for a network of spiking neurons and visual pattern recognition. In: Blanc-Talon, J., Philips, W., Popescu, D., Scheunders, P. (eds.) ACIVS 2006. LNCS, vol. 4179, pp. 1133–1142. Springer, Heidelberg (2006). https://doi.org/10.1007/11864349_103
14. Wysoski, S.G., Benuskova, L., Kasabov, N.: Text-independent speaker authentication with spiking neural networks. In: de Sá, J.M., Alexandre, L.A., Duch, W., Mandic, D. (eds.) ICANN 2007. LNCS, vol. 4669, pp. 758–767. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-74695-9_78
15. Wysoski, S.G., Benuskova, L., Kasabov, N.: Adaptive spiking neural networks for audiovisual pattern recognition. In: International Conference on Neural Information Processing, pp. 406–415. Springer, Heidelberg (2017). https://doi.org/10.1007/978-3-540-69162-4_42
16. Soltic, S., Wysoski, S.G., Kasabov, N.K.: Evolving spiking neural networks for taste recognition. In 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence) (pp. 2091–2097). IEEE, June 2008
17. Soltic, S., Kasabov, N.: Knowledge extraction from evolving spiking neural networks with rank order population coding. *Int. J. Neural Syst.* **20**(06), 437–445 (2010)
18. Perepu, S.K., Dey, K.: CDDM: a method to detect and handle Concept Drift in Dynamic Mobility Model for seamless 5G Services. In: 2020 IEEE Globecom Workshops (GC Wkshps), pp. 1–6. IEEE, December 2020
19. Thantharate, A., Paropkari, R., Walunj, V., Beard, C.: DeepSlice: a deep learning approach towards an efficient and reliable network slicing in 5G networks. In: 2019 IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), New York City, NY, USA, 2019, pp. 0762–0767, doi: <https://doi.org/10.1109/UEMCON47517.2019.8993066>
20. Thantharate, A., Paropkari, R., Walunj, V., Beard, C., Kankariya, P.: Secure5G: a deep learning framework towards a secure network slicing in 5g and beyond. In: 2020 10th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 2020, pp. 0852–0857. doi: <https://doi.org/10.1109/CCWC47524.2020.9031158>
21. Koelstra, S., et al.: DEAP: a database for emotion analysis; using physiological signals. *IEEE Trans. Affective Comput.* **3**(1), 18–31 (2012). <https://doi.org/10.1109/T-AFFC.2011.15>
22. Hochreiter, S., Schmidhuber, J.: Long short-term memory. *Neural Comput.* **9**(8), 1735–1780 (1997)