# The Enhancement of Evolving Spiking Neural Network with Firefly Algorithm

Farezdzuan Roslan, Haza Nuzly Abdull Hamed and Mohd Adham Isa
*Faculty of Computing, Universiti Teknologi Malaysia, 81310 Johor Bahru, Johor, Malaysia.*
*farez92@gmail.com*

***Abstract*—This study presents the integration between Evolving Spiking Neural Network (ESNN) and Firefly Algorithm (FA) for parameter optimization of ESNN model. Since ESNN lacks the ability to automatically select the optimum parameters, Firefly Algorithm (FA), as one of nature inspired metaheuristic algorithms is used as a new parameter optimizer for ESNN. The proposed method, ESNN-FA is used to determine the optimum value of ESNN parameter which is modulation factor (Mod), similarity factor (Sim) and threshold factor (C). Five standard datasets from UCI machine learning are used to measure the effectiveness of the proposed work. The classification results obtained shown an increase in accuracy than standard ESNN for all dataset except for iris dataset. Classification accuracy for iris dataset is 84% which less than standard ESNN with 89.33%. The rest of datasets achieved higher classification accuracy than standard ESNN which for breast cancer with 92.12% than 66.18%, diabetes with 68.25% than 38.46%, heart with 78.15% than 66.3% and wine with 78.66% than 44.45%.**

***Index Terms*—Evolving Spiking Neural Network; Firefly Algorithm; Nature Inspired Algorithms; Parameter Optimization.**

## I. INTRODUCTION

The inspiration of structural system of human brain has inspired the researchers to develop artificial intelligent method, Artificial Neural Network (ANN) in classification solving method. ANN has three generations of neural network models: binary networks, real-valued networks and spiking neural network (SNN) [1]. SNN is more powerful tool of computation than other two neural network models [2], [3].

SNN have several types of models representing their own abstraction level. The first two models are prominent which are known as the conductance and threshold models [4]. The conductance models is invented by [5] and was then simplified by [6]. The threshold models represent high level of abstraction which are known as leaky integrate and fire model (LIF) and spike response model (SRM). There is one of the SNN models which is called as Evolving Spiking Neural Network (ESNN). The architecture of ESNN was first proposed by Wysoski et al. [7] that take Kasabov's [8] evolving model as the foundation through ECoS that evolve the structure of the network in the process of training. ESNN model shown to have the capabilities in simplicity and efficiency with its fast one-pass learning algorithm.

The issue in ESNN model is to determine the optimal value for parameters. ESNN has three important parameters which are modulation factor (Mod), threshold factor (C) and similarity value (Sim). The selection for each parameter value is done manually [9]. Thus, increases the computational time and rendered the model less efficient.

Swarm Intelligence (SI) has proven itself to have the adaptive ability in various environments and inspired many optimization techniques [10]. Firefly Algorithm (FA) is one of many SI algorithm that is used commonly in classification. However, the integration of FA as parameter optimizer in ESNN have not been done yet. The advantages of using these algorithm are the works of information sharing which can improve the algorithm to converge much faster and have higher probability to escape before trapped into local modes [11]. In [12], FA has been proven to have better search and adaptive ability. Therefore in this research, FA is proposed as the new optimizer for ESNN to optimize parameter values.

The explanation of the paper will be organized as follows: ESNN is explained in Section II, FA will be described in Section III. Section IV, the explanation of methods used. Section V shows the experimental results and the discussions and Section VI is the conclusion of the paper.

## II. EVOLVING SPIKING NEURAL NETWORK

One of the improved models of SNN is called Evolving Spiking Neural Network (ESNN). The improvement was made by Wysoski [13]. The ESNN model has been following two principles which are the likelihood of creating new classes and the merging for the same similar weight [1]. Same encoding method used for SNN which is the population as explained in [14]. An input neuron j firing times calculation can be done by using the point of intersect of Gaussian function. Equation 1 calculated the centre whereas Equation 2 calculated the width with the variable interval of [*Jmin, Jmax*]. Each receptive field width is controlled by parameter β.

$$R=J_{min}+(2j-3)/2*(J_{max}-J_{min})/(M-2) \tag{1}$$

$$\sigma=1/\beta\cdot(J_{max}-J_{min})/(M-2),\ 1\leq\beta\leq2 \tag{2}$$

where: R = The centre of Gaussian intersection
σ = The width of variable interval [*Jmin, Jmax*]

The Fast Integrate and Fire Model proposed by Thorpe [15] are based on a neuron receiving the spikes that arrived earliest will achieve a weight that is stronger compared to the spikes that arrived later. The neuron will fire and becomes disabled when the definite number of spikes is reached and threshold value is exceeded by Post-Synaptic Potential (PSP). Equation 3 presented the calculation of PSP of neuron j.

$$\mu_e=\begin{cases} 0 \text{ if fired} \\ \sum w_{fe}*Mod_e^{order\,(f)} \text{ otherwise} \end{cases} \tag{3}$$

where: $w_{fe}$ = Pre-synaptic neuron f weight
$Mod_e^{order\ (f)}$ = Modulation factor with interval [0, 1]
order (f) = The spike rank of neuron

The value for order (f) begins with zero according to the spikes which is first among all pre-synaptic neuron and increases by the firing time. New output neuron is created for each training sample as described in One-pass Learning algorithm. Further details regarding ESNN can be found in [16].

## III. Firefly Algorithm

Yang [17] proposed Firefly Algorithm (FA) that was inspired on how the flashes behavior of fireflies. Primarily the flashing of fireflies acts as a signaling system to attract the other fireflies and potential prey. FA simulates the flash pattern and characteristic of fireflies. In developing FA,[18] stated three idealized rules are as follows:

i. Firefly is an epicene.
ii. The less bright firefly is attracted to brighter firefly that moving randomly.
iii. he brightness of a firefly indicates the optimization of the object function.

In FA optimization, the light intensity of a firefly will determine the attraction of another fireflies. Therefore, firefly i and firefly j will have varied distance rij. Moreover, when the light density decreasing, its distance also decreases for its light is absorbed within the media. Given Equation 4, intensification of light, I(r) differ based on the law relation of square inverse.

$$I(r)=I_s/r^2 \tag{4}$$

where: $I_s$ = Light intensity at the source

$$x_i^{t+1} = x_i^t + \beta_0 e^{-\gamma r^2}\left(x_j^t - x_i^t\right) + \alpha_t \epsilon_i^t \tag{5}$$

For the second term, it is used to represent the attractiveness of the fireflies. The value for light absorption coefficient γ, is fixed and value $\beta_0$ is used to represent the initial light intensity when r = 0. For the firefly to move randomly, the third term used, $\alpha_t$ being the randomization parameter while for the vector of random numbers $\epsilon_i^t$ is taken from Gaussian or uniform distribution at time t. In most cases of implementations, $\beta_0 = 1$ and $\alpha \in [0, 1]$. On the other hand, the parameter γ varies from 0.01 to 100 and categorizes the variant attractiveness and using its value to determine the convergence speed and behavior of FA [19].

## IV. The Integration of ESNN-FA

The parameters of ESNN is optimized using FA to get optimal performance. Before optimizer is used, the parameters are adjusted manually and it is deemed to be infeasible since the process would be inefficient and unsystematic. Previous research implementing integration of optimization algorithm has shown promising result. Thus, we propose the optimizer algorithm that will be integrated with ESNN is Firefly Algorithm (FA).

The integration of ESNN-FA is conducted using the well-known Wrapper method. The method was initially inaugurated by [20]. Later, the method is thoroughly analyzed later by [21] and further analyze by [22]. The Wrapper method takes the classifier and melds with an optimization algorithm.

In this study, ESNN as the classifier is combined with FA as an optimizer. FA integrated with the classifier ESNN to optimize the parameters, modulation factor (Mod), threshold factor (C) and similarity value (Sim). A set of random values is used to initialize all the candidates. Subsequently, the candidates interact with each other based on the classification accuracy.

For each candidate, holds definite parameter values and make up the structure of ESNN. Then, the input samples will go through ESNN. This process will classify the input samples according to their targeted classes. The evaluation of candidate's performance is done by calculating the fitness function based on the classification accuracy. In the integration approach, the fitness function is a necessary element [22]. In other words, all input samples will be encoded into spikes and pass through ESNN model to find the current fitness.

The FA_Best is assigned with the best classification accuracy hold by the candidates. If the FA candidates encounter a better result compared to the FA_Best, the new created FA_Best will replace the FA_Best. The iteration will be repeated until the termination criterion is reached. Figure 1 shows the ESNN-FA framework and Figure 2 explains the algorithm 1 of the proposed integration.

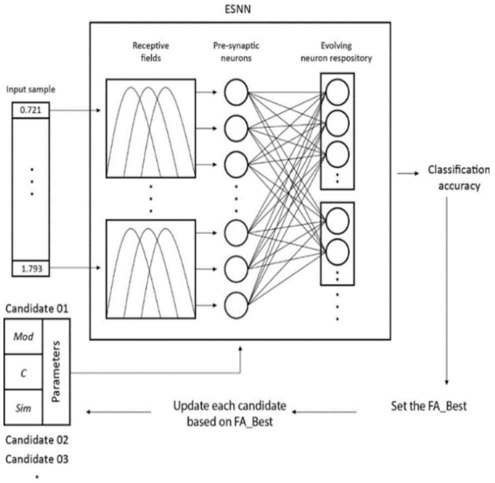

Figure 1: The Proposed ESNN-FA Framework

```
Initialize candidates
Read input dataset
Initialize ESNN parameters:
Mod= [0, 1], C= [0, 1] and Sim = [0, 1]
Initialize neuron repository R
while not reaching maximum iteration do
    for all candidates do
            Encode input sample into firing time of pre-synaptic neuron f
            for all input sample e belong to the same
            output class do
                Set weight for all pre-synaptic neuron
                where $wf = Mod_e^{order(f)}$
                Calculate $PSP_{max}(e) = \sum wfe * Mod_e^{order(f)}$
                Get PSP threshold value, $\theta = PSP_{max}(e) * C$
                if the trained weight vector <= Sim of trained weight in R then
                    Merge weight and threshold value with most similar neuron
                    Calculate $w = \frac{w(new)+w*N}{N+1}$
                    Calculate $\theta = \frac{\theta(new)+\theta*N}{N+1}$ where N is the number of merge before
                else Add new neuron to output neuron repository R
                end if
    end for (Repeat to all input samples for other output class)
    Calculate fitness
    Update FA individuals
    Do testing according the best fitness value
end while
```

Figure 2: Algorithm 1 of ESNN-FA

### A. Experiment Setup

There are five datasets taken for this study from UCI machine learning repository [23]. The datasets used are Iris, Wisconsin breast cancer, Pima Indians Diabetes, Heart and Wine.

All datasets undergone pre-processing before it is suitable for training. The paper points out that training and testing datasets are normalized before the experiment is started. The range of normalization is [0, 1] by Equation 6.

$$\bar{m}=\frac{m\text{-}min_m}{max_m\text{-}min_m} \tag{6}$$

where: m = Original attribute value
$\bar{m}$ = The normalized attribute value
$max_m$ = Maximum values of the attribute m
$min_m$ = Minimum values of the attribute m

Moreover, FA currently has its own parameters. The selected values for FA used parameters are shown in Table 1.

Table 1
Setting Value for FA Parameters in ESNN-FA

| Parameter | Value |
|---|---|
| Randomness (□) | 0.25 |
| Absorption (γ) | 1.0 |
| Initial attractiveness ($\beta_0$) | 1.0 |
| Population size | 50 |

## V. Experiment Results and Discussion

In this section, the paper presents the experimental results of ESNN-FA integration and the comparison to standard ESNN, and DE-ESNN [1]. First, the dataset is separated into two different parts; testing and training. By using 10-fold cross validation, 90% of total sample makes the training dataset and the rest is testing dataset.

Next, after the dataset is normalized and prepared, the integration of ESNN-FA will be trained based on the final datasets prepared to start classifying the dataset according to their classes. From these classification, the performance of the proposed method is calculated with the accuracy of the correct classification. The three important parameters can determine the result of training in ESNN. The value of Mod, C and Sim are changing between 0 and 1.

Table 2 shows the classification accuracy result using manually selected value of ESNN parameters. The experiments are conducted for ESNN without the use of optimizer. The value used shown that by changing the ESNN parameter, the results can achieve better accuracy for all datasets.

Table 2
Classification Result Obtained Using Manually Selected Value of ESNN Parameters

| Dataset | Manual Tuning ESNN Parameter | | | Classification accuracy % |
|---|---|---|---|---|
| | Mod | C | Sim | |
| Iris | 0.9 | 0.9 | 0.1 | 92.00 |
| Breast cancer | 0.9 | 0.35 | 0.1 | 94.41 |
| Diabetes | 0.9 | 0.9 | 0.1 | 56.41 |
| Heart | 0.85 | 0.75 | 0.1 | 74.44 |
| Wine | 0.9 | 0.95 | 0.1 | 66.67 |

Table 3 shows the parameter combination and accuracy result for ESNN-FA from all datasets. From the table, it is clear that for each dataset, FA search for the optimum value of ESNN parameters in regards with the best fitness accuracy.

Table 3
ESNN Parameter Optimized by FA and Testing Accuracy Obtained for All Dataset

| Dataset | ESNN Parameter | | | Classification accuracy % |
|---|---|---|---|---|
| | Mod | C | Sim | |
| Iris | 0.94 | 0.92 | 0.94 | 84.00 |
| Breast cancer | 0.98 | 0.60 | 0.26 | 92.12 |
| Diabetes | 0.89 | 0.28 | 0.49 | 68.25 |
| Heart | 0.59 | 0.28 | 0.79 | 78.15 |
| Wine | 0.89 | 0.98 | 0.14 | 78.66 |

From these results, it can be observed that the optimum parameter values are different for all five datasets. The proposed method shows that there is no specific combination of parameter value for all datasets. This means that the best accuracy results can be achieved when the optimum parameter is found.

The result of the experiment has been compared and analysed based on the classification accuracy achieved. Table 4 shows the results with comparison between ESNN-FA, standard ESNN and DE-ESNN for all datasets in training and testing data. The result for standard ESNN parameter is tested from [24] and the manually tuned ESNN parameter is from Table 1. The comparison for standard ESNN is represented by these results.

From Table 4, the results reported the accuracy achieved by ESNN-FA in the training phase was better except for heart dataset with accuracy of 84.16%. In addition, the findings show that the accuracy in the testing phase of ESNN-FA have demonstrated better performance for diabetes, heart and wine datasets when compared with standard ESNN.

On the other hand, when comparing the overall classification of the proposed methods, the accuracy for heart has a promising result with an accuracy of 78.15%. However, the accuracy of ESNN-FA for iris dataset is not satisfied with values of 84%, yet the proposed methods still outperforms the standard ESNN. On the other hand, at least heart dataset have better classification accuracy compared to MLP.

Furthermore, as it can be seen in the testing phase, the accuracy values indicate that ESNN-FA has resulted in better convergence compared to DE-ESNN. For breast cancer dataset, the accuracy result shown better than DE-ESNN with 92.12%. In addition, for heart dataset the accuracy achieved is 78.15%. The heart dataset shown an increase in 15.44% than DE-ESNN.

This shows FA have better adaptive ability and generalization performance [12]. These reflects to FA searching algorithm which improved the classification accuracy with ESNN.

Table 4
Comparisons between the Proposed Method and Existing Optimization Methods

| Dataset | Algorithm | Classification accuracy % | |
|---|---|---|---|
| | | Training | Testing |
| Iris | Standard ESNN Parameter | 96.23 | 89.33 |
| | Manually Tuned ESNN | 95.56 | 92.00 |
| | ESNN-FA | 100 | 84.00 |
| | DE-ESNN | 97.62 | 93.33 |
| Breast cancer | Standard ESNN Parameter | 95.02 | 66.18 |
| | Manually Tuned ESNN | 99.74 | 94.41 |

| | | | |
|---|---|---|---|
| | ESNN-FA | 100 | 92.12 |
| | DE-ESNN | 97.62 | 91.18 |
| Diabetes | Standard ESNN Parameter | 44.39 | 38.46 |
| | Manually Tuned ESNN | 70.03 | 56.41 |
| | ESNN-FA | 76.78 | 68.25 |
| | DE-ESNN | - | - |
| Heart | Standard ESNN Parameter | 73.83 | 66.30 |
| | Manually Tuned ESNN | 80.04 | 74.44 |
| | ESNN-FA | 84.16 | 78.15 |
| | DE-ESNN | 97.47 | 62.71 |
| Wine | Standard ESNN Parameter | 44.75 | 44.45 |
| | Manually Tuned ESNN | 66.81 | 66.67 |
| | ESNN-FA | 85.95 | 78.66 |
| | DE-ESNN | - | - |

## VI. Conclusion

In this paper, an integration of ESNN-FA was proposed to search for optimal parameter values of ESNN. The comparison study between the ESNN-FA, standard ESNN and DE-ESNN have been done to illustrate the performance of ESNN-FA. As a result, ESNN-FA is proven with capability to classify most dataset better compared to standard ESNN and DE-ESNN. The results have shown promising results because of the improved classification accuracy with ESNN-FA. In future work, further modification to improve the performance of ESNN-FA will be conducted.

## Acknowledgment

This research is supported and funded by Universiti Teknologi Malaysia under the Research University Grant with vot. Q.J130000.2528.11H80.

## References

[1] A. Saleh, H. Hameed, M. Najib, and M. Salleh, “A novel hybrid algorithm of differential evolution with evolving spiking neural network for pre-synaptic neurons optimization,” *Int. J. Adv. Soft Computing and Its Applications*, vol 6, no. 1, pp. 1-16, Mar. 2014.

[2] W. Maass, “Noisy spiking neurons with temporal coding have more computational power than sigmoidal neurons,” *Inst. Theor. Comput. Sci. Tech. Univ. Graz. Graz, Austria, Tech. Rep. TR-1999-037*, 1999. [Online]. Available at http//www. igi. tugraz. at/psfiles/90. pdf.

[3] B. Schrauwen and J. Van Campenhout, “Backpropagation for population-temporal coded spiking neural networks,” in *Neural Networks, 2006. IJCNN’06. International Joint Conference on*, 2006, pp. 1797–1804.

[4] W. Gerstner and W. M. Kistler, *Spiking Neuron Models: Single Neurons, Populations, Plasticity*. Cambridge university press, 2002.

[5] A. L. Hodgkin and A. F. Huxley, “A quantitative description of membrane current and its application to conduction and excitation in nerve,” *J. Physiol.*, vol. 117, no. 4, pp. 500-544, Aug. 1952.

[6] E. M. Izhikevich, “Simple model of spiking neurons,” *IEEE Trans. Neural Networks*, vol. 14, no. 6, pp. 1569–1572, 2003.

[7] Wysoski, Benuskova, and Kasabov, “Adaptive learning procedure for a network of spiking neurons and visual pattern recognition,” in *Advanced Concepts for Intelligent Vision Systems*, vol. 4179, J. Blanc-Talon, W. Philips, D. Popescu, and P. Scheunders, Eds. Springer Berlin Heidelberg, 2006, pp. 1133–1142.

[8] N. Kasabov, *Evolving Connectionist Systems: The Knowledge Engineering Approach*. Springer Science & Business Media, 2007.

[9] S. G. Wysoski, L. Benuskova, and N. Kasabov, “Adaptive Spiking Neural Networks for Audiovisual Pattern Recognition,” in *Neural Information Processing*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 406–415.

[10] N. F. Johari, A. M. Zain, N. H. Mustaffa, and A. Udin, “Firefly algorithm for optimization problem,” *Appl. Mech. Mater.*, vol. 421, pp. 512–517, Sep. 2013.

[11] L. Zhang, L. Liu, X.-S. Yang, Y. Dai, V. Zumer, and H. Zhang, “A novel hybrid firefly algorithm for global optimization,” *PLoS One*, vol. 11, no. 9, p. e0163230, Sep. 2016.

[12] H. Su, Y. Cai, and Q. Du, “Firefly-algorithm-inspired framework with band selection and extreme learning machine for hyperspectral image classification,” *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 10, no. 1, pp. 309–320, Jan. 2017.

[13] S. G. Wysoski, L. Benuskova, and N. Kasabov, “On-Line Learning with Structural Adaptation in a Network of Spiking Neurons for Visual Pattern Recognition,” in *Artificial Neural Networks – ICANN 2006: 16th International Conference, Athens, Greece, September 10-14, 2006. Proceedings, Part I*, S. D. Kollias, A. Stafylopatis, W. Duch, and E. Oja, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 61–70.

[14] S. M. Bohte, H. La Poutré, J. N. Kok, and H. La Poutre, “Error-backpropagation in temporally encoded networks of spiking neurons,” *Neurocomputing*, vol. 48, no. 1, pp. 17–37, 2002.

[15] S. THORPE and J. Gautrais, “How can the visual system process a natural scene in under 150 ms? On the role of asynchronous spike propagation,” in *European symposium on artificial neural networks*, 1997, pp. 79–84.

[16] H. N. A. Hamed, N. Kasabov, and S. M. Shamsuddin, “Integrated feature selection and parameter optimization for evolving spiking neural networks using quantum inspired particle swarm optimization,” in *Soft Computing and Pattern Recognition, 2009. SOCPAR’09. International Conference of*, 2009, pp. 695–698.

[17] X.-S. Yang, “Firefly algorithm,” *Nature-Inspired Metaheuristic Algorithms*, vol. 20, pp. 79–90, 2008.

[18] X.-S. Yang and X. He, “Firefly algorithm: recent advances and applications,” *Int. J. Swarm Intell.*, vol. 1, no. 1, pp. 36–50, 2013.

[19] T. Hassanzadeh and M. R. Meybodi, “A new hybrid approach for data clustering using firefly algorithm and K-means,” in *Artificial Intelligence and Signal Processing (AISP), 2012 16th CSI International Symposium on*, 2012, pp. 7–11.

[20] G. H. John, R. Kohavi, and K. Pfleger, “Irrelevant features and the subset selection problem,” in *Machine learning: Proceedings of the Eleventh International Conference*, 1994, pp. 121–129.

[21] R. Kohavi and G. H. John, “Wrappers for feature subset selection,” *Artif. Intell.*, vol. 97, no. 1–2, pp. 273–324, Dec. 1997.

[22] M. Valko, N. C. Marques, and M. Castellani, “Evolutionary feature selection for spiking neural network pattern classifiers,” in *2005 Purtuguese Conf. Artif. Intell.*, 2005, pp. 181–187.

[23] K. Bache and M. Lichman, *UCI Machine Learning Repository*. Irvine, CA: University of California, School of Information and Computer Science, 2013. [Online]. Available at http//archive. ics. uci. edu/ML.

[24] H. N. A. Hamed, *Novel Integrated Methods of Evolving Spiking Neural Network and Particle Swarm Optimisation*, Auckland University of Technology, 2012.