Nat. Volatiles & Essent. Oils, 2021; 8(5): 4866 – 4782

NVEO
Natural Volatiles & Essential Oils

Protein Secondary Structure Prediction Of Qscr Protein Using Ffa Optimized Ann

[1]Saravanan K, [2]Sivakumar S ,[3] Sangeetha B and [4]Praveen H

[1]Department of Physics, AVS Engineering College, Salem, Tamil Nadu, INDIA
[2]Department of Physics, Government Arts College (Autonomous) Salem, Tamil Nadu, INDIA
[3]Department of EEE, AVS Engineering College, Salem, Tamil Nadu, INDIA
[4]Department of Physics, Nehru College of Engineering and Research Centre, Pampady, Thrissure, Kerala, INDIA.

ABSTRACT

Identifying Secondary structure of protein becomes more important in designing drugs. Presently, machine learning algorithms like Artificial Neural Network (ANN) and Support Vector Machine (SVM) has been utilized to identify the Protein Secondary Structure (PSS). Over the years, many hybrid methods are evolved for the optimization. In this work, to train a FFNN, a heuristic nature inspired algorithm namely firefly algorithm (FFA) is incorporated with BP algorithm to get a quick and enhanced convergence rate in training FFNN. Using this algorithm, enhanced convergence within a very few repetitions can be attained. It has also been observed that by this approach, higher accuracy in Protein Secondary Structure (PSS) of QscR can be achieved when compared to that of other existing techniques.

Keywords: Artificial Neural Network, Protein Secondary Structure, Firefly Algorithm, QscR.

INTRODUCTION

In a natural life, the calculation procedure for natural computation depends on the mechanisms which are streamlined and the functions present in it. Increased number of entities can be optimized and processed by applying this simplification. At the present time, the answer for many science and engineering problems are obtained from the ideas inspired by nature. To incorporate the natural process and the activities taking place in natural life, different computational methods are included. Thus, to arrive at the solution for more complex problems, computational methods are very much useful. By these methods many ideas which are naturally inspired are involved to form a hybrid process.

Based on the novel problem solving methodologies and standardized method, hybrid algorithm is evolved. Applying algorithm of this type are found to be specifically utilized in engineering and scientific development issues. Commonly, algorithm are evolved and altered in view to face the complexity which is increasing in the real time applications regularly.

The natural life acts itself as an optimization algorithm which paves the way for the development of GA. The inclusion of the BP method with GA gives better answer and it forms a hybrid method named as GA based BPNN. By using this algorithm, solutions for many scientific issues is obtained. Numerous comparative analyses have been executed on the GABPNN. This algorithm is found to be suitable find

suitable in any changing environment. For forecasting the stack rates also, this method can be used. In spite of its numerous successful applications, due to its large search space, the algorithm is suffered by its slow convergence.

Even though the BPNN has its own restrictions, this method finds its applications in various systems and this approach acts as an essential factor. The NN training, which is based on algorithm of BP depends on a few variables such as bias, learning rate and weight of algorithms. This method begins with initial values for weights and in accordance with each input and output, the weight gets updated. Based on the performance of capacity of learning, the parameters of learning are updated, but in many implementations the parameters are defined as static.

The typical BP uses steepest descent method; hence this method is called as SDBP algorithm. This method is also named as gradient method. In learning process, this algorithm adopts the gradient methodology. However, this process is generally confined by its minimum convergence time. The convergence is the principal matter in the case of BP learning algorithms, through which the FFNN is trained. With every single suggested algorithm, the change in convergence is tuned. The growth of this method is classified into two parts. Thus, the first part is executed using influence of numerical optimization technique. Levenberg-Marquardt (LM) BP and conjugate BP methods are used mostly. Even though the conjugate BP algorithm converges within a restricted number of iteration, this algorithm does not need any second order derivative. On the basis of batch mode, the parameters are modified. The standard topology adopted for non-linear least square algorithm is LM method. By using the iterative process to find local minimum, the LM algorithm decreases the issues of Gauss Newton method. The two methods namely the Steepest Descent and Gauss Newton methods are the constituents of LM algorithm. While the index performance is not converged, the LM behaves as Standard Propagation method and when it is converged, it behaves as Gauss Newton method. The performance of this algorithm takes place at a low speed and it needs more memory in development[1, 2]

With analytically instigate solution, the BP method was applied. One of the most well-known methods is the momentum BP algorithm. With each and every input and output, the weight parameter is updated in this method by the inclusion of filter. The variable learning back-propagation algorithm (VLBP) is another type of interested application. Performance index is the key factor through which the parameters for learning are modified. For the modification to BP algorithm, the updations should be made on the parameters in an improvised way.

The parameters such as weights and others are upgraded for each description of the input or output pair in the majority of the standard optimization procedures. In this method, the weight parameter is modified for each layer. Because of this, the search space for identifying the bias matrix and weight expands. Using the batch mode, the standard methods of BP algorithm are operated [3,4].

FFNN is trained by applying many types of BP approaches. Each type of approach has its own merits and demerits. The convergence happens firmly in the GA based NN method, but it needs more iterations to converge. In this method due to the large search space, there is no précised topology is followed to find it into local minimum. However, in this suggested method, iteration optimizing is less

when compared with other algorithms. The suggested FFA tuned NN is more stable due to the criteria of fixed convergence within short time [5,6]. The FFA tuned NN is not at all trapped in a local minimum and if the condition for convergence of data set is predefined, this method stops immediately.

FIREFLY ALGORITHM (FFA)

The flashing light of the fire flies is based on the principle of bioluminescence. In the life cycle of the fire flies, there exist many different views towards the role of flashing light. In spite, most of these views are associated with mating phase. The base task of flashing light is to captivate the mating partner. At this stage the phenomena of luminescent emission takes place. The exclusive role of the flashing light is to indicate the acceptance for willingness on mating. During the process of luminescent emission, the couple involved in mating are brought together for intercourse. In a firefly species, the Plotinus is a category of species. In this category, the male firefly sends a signal to the female and the female responds to that signal after a certain gap of time. Species of other category of firefly reveals contrasting habits of mating based on the varying circumstances [7-12].

The salient features of this algorithm are

1. Either gender fireflies are engaged to one another, no matter what sex they belongs to.

2. During the phase of mating, the major role is played by the flashing light. The two parameters, luminosity and attractiveness, are responsible for mating. The degree of attraction during mating is proportional to the luminosity of the flashing light. The firefly which is less in brightness is engaged by the brighter one or the less shining firefly automatically move towards the highly dazzling firefly.

3. The attracting behavior during mating is affected due to the flashing light, it is considered as a main objective function. The brightness parameter is also influenced by the objective function.

The population of the FFA at the beginning is given by

$$P(f) = (f_1, f_2, f_3 \dots f_n)^T \quad (1)$$

For $f_i = (1,2,3 \dots, n)$ (2)

Using the above function, the intensity of their flashing light can be determined as

$$L_i = L_o e^{(-nd^2)} \quad (3)$$

Where,

L_i = Light intensity,

L_o = light intensity at Initial condition,

η = Light absorption coefficient,

d = Distance between flies.

Thus 'd' can be calculated as

$$d_{ij} = \|f_i - f_j\| = \sqrt{\Sigma(f_i - f_j)^2} \qquad (4)$$

A constant value is assigned to absorption parameter of the flashing light at the beginning and this value is changed in accordance with optimization process. By modifying the η, the algorithm can be converged. Almost in all the cases, the absorption parameter has a constant value. The execution of the algorithm starts and it continues till the end of the generation. Initially, out of the total population, one is assigned as brightest one and other flies are attracted towards it.

Under this condition, the two parameters 'distance' and 'attractiveness' is calculated for each firefly from brighter one to the rest and this affects each firefly differently. On completing this movement in a successful manner, based on their performance, the fireflies are ranked.

ANALYSIS OF PROPOSED METHOD

Fig 1 denotes the FFNN in a précised manner. By incorporating the FFA in the BPNN, the performance index can be optimized.

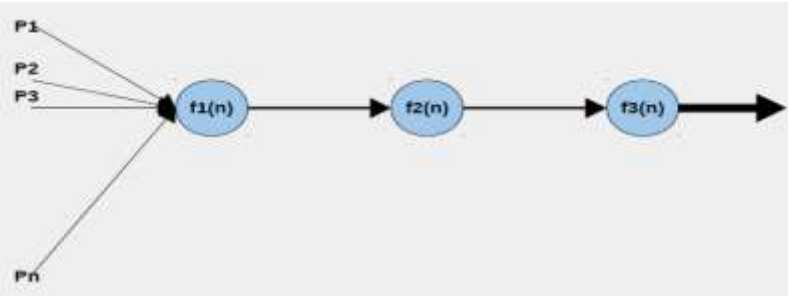

Figure 1. Design of FFNN for BP.

Using three layers as shown in figure 1, the BPNN is developed with one neuron in each layer. The outcome from every single neuron is computed as

$$P^{(n+1)}(r) = f^{(n+1)}\left(N^{(n+1)}(r)\right) \qquad (5)$$

Where,

p = output of a neuron

f = Transfer Function

N = Net output obtained from a neuron.

With the help of a group of weights which are generated randomly, the suggested FFA based BP is started. To proceed with additional action, each weight which is generated randomly is added to the BPNN. The sum of the squared error of weight matrix is generated from the input pattern matrices through the BPNN. In the suggested FFA, the performance index depends on the sum of squared error.

Hence, for a weight matrix, the weight values are calculated using the expression given below.

$$WV_m = \sum_{(m-1)}^{l} a.\left(rand - \frac{1}{2}\right) \quad (6)$$

WV_m = Weight value of a weight matrix (m -1,2,3,…,l)

rand = Ranges from 0 - 1.

a = Constant

Hence, the list of weight matrix can be written as

$$W^L = \left[WV_m^1, WV_m^2 WV_m^3, \ldots\ldots WV_m^{(q-1)}\right] \quad (7)$$

Using the above expression, the sum of the squared error can be easily computed from the process of BP.

$$V(x) = \sum_{(j=1)}^{m}\left(t_j - p_j\right)^T.\left(t_j - p_j\right) = \sum_{(j=1)}^{m} e^T.e \quad (8)$$

where,

t = Target (Input pattern),

p = input pattern matrix.

Using the formula given below, the performance index for the suggested method under study can be arrived

$$F(v) = \sum_{(i=1)}^{q} v(x)^T.v(x) \quad (9)$$

For the performance index, the gradient is the first order derivative and that can be expressed as

$$\nabla F(v) = \left[\frac{\delta(F(v))}{\delta v_1}, \frac{\delta(F(v))}{\delta v_2}, \ldots \frac{\delta(F(v))}{\delta v_n}\right] \quad (10)$$

$$\nabla F(v) = 2\sum_{(i=1)}^{(q)} v_i(x).\frac{\delta v_i(x)}{\delta x} \quad (11)$$

The parameters of BPNN, weight and bias values are determined in equations 12 and 13 which are given below

$$W_{(r,j)}^{(n+1)} = W_{(r,j)}^{n} - \lambda.s^n\left(a^{(n-1)}\right)^T \quad (12)$$

And

$$B_{(r,j)}^{(n+1)} = B_{(r,j)}^{n} - \lambda.s^n \quad (13)$$

Here

λ = learning parameter

S^n = sensitivity of n^{th} layer.

A sensitivity of the current layer can be determined by knowing the sensitivity of the preceding layer. Hence, in a repetitive pattern, the sensitivity can be estimated from NN.

The sensitivity can be computed as follows

$$S^n = f^n(N^n).\left(W^{(n+1)}\right)^T.S^{(n+1)} \qquad (14)$$

Similarly the sensitivity for the input layer is computed as

$$S^n = -2.f^n(N^n).\left(t_j - p_j\right) \qquad (15)$$

Hence from the equation (5.9), the value of the performance index of the suggested method is estimated. This estimated value is then stored in the index list.

$$F^L(x) = [f_1(x), f_2(x), f_3(x), \dots \dots f_n(x)] \qquad (16)$$

The values of $F^L(X)$ are taken as firefly, and these fireflies are in the contest of mating. To incorporate FFA into BPNN training method the facts mentioned below are taken into account.

1. Reducing the performance index is the ultimate task of the BP training algorithm. The performance index list with a minimum error is taken as a more captivating firefly.

2.When the error is more the attraction is low and vice versa.

3.The search process converges when the light absorption co-efficient (y) increases with iteration process.

It is noticed, that the equation (1) and (16) are nearly identical logically. The identification of the ultimate encaging firefly can be easily found using the performance index list (Eqn.16). In this suggested method, the highly intense firefly is denoted by f_j and remaining of less intense by f_i. Now by using eqn.(4) and (3) , the gap between f_i and f_j and the power of the flashing light is determined. At once, the migration of the fireflies from f_i to f_j is found by using the formula given below

$$\Delta f_i = f_i + L_o e^{\left(-\eta d^2\right)}\left(f_j - f_i\right) + \alpha\left(rand - \frac{1}{2}\right) \qquad (17)$$

In eqn.(5.7), the second term arises based on the attraction and the final term is arbitrary accompanied by an persistent parameter ‘d’. By adjusting the weight matrix of the concerned weight list, the learning process can be made more effective and is denoted as

$$W_{(r,j)}^{(n+1)} = W_{(r,j)}^{n} - \Delta F_i \qquad (18)$$

The bias for the respective weight matrix can be adjusted based on the formula mentioned here

$$B_{(r,j)}^{(n+1)} = B_{(r,j)}^{n} - \Delta F_i \qquad (19)$$

The OF_i gives the adjusted space between f_i and f_j. There exist two convergence principles in the suggested algorithm. The first criterion is the classification, followed by the second, which is the aggregate sum of the square error. After the complete delegation of the entire weight matrix in the weight list, the two criteria are determined by using this suggested method. On completion of every iteration, the average of sum of the squared error is computed. Thus the matching percentage of the input and the output gives the correct classification. When the input is equal to the output, then it will produce a matching with high percentage else the matching percentage, will be low. Hence at the end

of the iteration, the average rate of correct classification is computed. The method under this study begins with the predetermined average correct classification and by the aid of keeping the average of sum of the squared error value as a one threshold value. Henceforth, the algorithm under study is as follows.

FFA Based BPNN

Input:

Non-linear input pattern and its corresponding target, light absorption coefficient and learning parameter ,.

Output:

Modified Weight and Bias matrix, Correct rate of Classification and SSE.

Begin:

Formulate different weight using Eqn. 6.

Calculate SSE for an individual weight. This SSE is considered as a performance index and the error value as firefly.

IF True:

Minimum error of SSE is calculated and assign it to f_j

while n < (length of SSE list):

Find next value of f_j and assign it to f_i

If (fi > fj) :

Calculate the distance.

Move Fi towards F_j using Eqn.(17).

Modify weight and bias value using Eqn.(18) and Eqn.(19)

else

End If

Recalculate SSE using modified weight and bias.

Calculate correct classification rate.

Increase n.

End While

If (avg. correct classification >threshold):

Stop the process

Else

Continue the process

End While

The suggested FFA which is based on BPNN training method converges in less iterations. When the population of the fireflies at the beginning is increased, the iteration number and the measure of correct classification can be speeded up.

RESULTS AND DISCUSSION

In this work, 100 proteins set for training and 5 protein set for testing were used. The pertaining sets and the testing set combine all the three classes of α-helix, β-strand and coil of QscR

protein SS. Each set was utilized as testing and validating set. The parameter configuration used for FFA is depicted in table 1

Table 1. The parameter configuration used in FFA

S.No	Parameters	Values
1.	No of optimized parameter	2
2.	Range of optimized parameter value	-10 to 10
3.	Number of flies	50
4.	Max. Iteration	200
5.	α	0.8
6.	γ	1
7.	β	0.97

Among the training data set, 47% were about coil, 31% were strand and 21% were Helix. In the testing data set, it was about 48% of C, 31% of E and 21% of H.

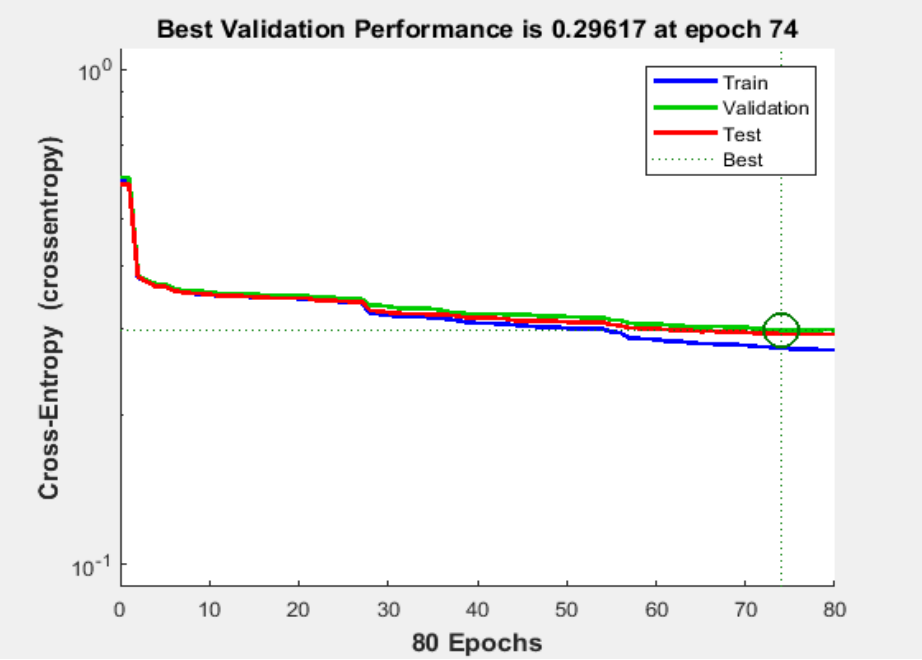

Figure 2. Best validation performance curve

Table 2. Predicted secondary structure of QscR under different topology

Methods		Secondary structure				
Sequence (1-50)		MHDEREGYLE	ILSRITTEEE	**FFSLVLEICG**	**NYGFEFFSFG**	ARAPFPLTAP
Structure	**DSSP**	******SHHH	HHHH** SHHH	HHHHHHHHHH	HTT*SEEEEE	EE***STTS*
	MLNN	CHHHHHHHHH	HHHHCCCHHH	HHHHHHHHHH	HHCCCEEEEE	EECCCCCCCC
	PSONN	CHHHHHHHHH	HHHHCCCHHH	HHHHHHHHHH	HHCCCEEEEE	EECCCCCCCC
	Proposed FFA-ANN	CHHHHHHHHH	HHHHCHCHHH	HHHHHHHHHH	HHCHCEEEEE	EECCCEECCC
Sequence(51-100)		KYHFLSNYPG	EWKSRYISED	YTSIDPIVRH	GLLEYTPLIW	NGEDFQENRF
Structure	**DSSP**	*EEEEE*** H	HHHHHHHHTT	GGGT*HHHHH	HHHS*S* EEE	ETTT*SS*HH
	MLNN	CEEEECCCCH	HHHHHHHHHC	CHHHCHHHHH	HHHCCCCEEE	CCCCCHHHHH
	PSONN	HEEEECCCCH	HHHHHHHHHC	CHHHHHHHHH	HHHCCCHEEE	CCCCHHHHHH
	Proposed FFA-ANN	HEEEECCCCH	HHHHHHHHHC	CHHHHHHHHH	HHHCCCCEEE	CCCCCHHHHH
Sequence(101-150)		FWEEALHHGI	RHGWSIPVRG	KYGLISMLSL	VRSSESIAAT	EILEKESFLL
Structure	**DSSP**	HHHHHHHTT*	*EEEEEEEE*	GGG*EEEEEE	EESSS*** HH	HHHHHHHHHH
	MLNN	HHHHHHHHCC	CCEEEEEEEC	CCCCEEEEEE	ECCCCCCCHH	HHHHHHHHHH
	PSONN	HHHHHHHHCC	CCEEEEEEEC	CCCCEEEEEE	ECCCCCCCHH	HHHHHHHHHH
	Proposed FFA-ANN	HHHHHHHHCH	CCEEEEEEEH	CCCCEEEEEE	ECCCCCCHHH	HHHHHHHHHH
Sequence(151-200)		WITSMLQATF	GDLLAPRIVP	ESNVRLTARE	TEMLKWTAVG	KTYGEIGLIL
Structure	**DSSP**	HHHHHHHHHH	HHHHHHHHSG	GGG**** HHH	HHHHHHHHTT	**HHHHHHHH
	MLNN	HHHHHHHHHH	HHHHCCCCCC	CCCCCCCHHH	HHHHHHHHHC	CCHHHHHHHH
	PSONN	HHHHHHHHHH	HHHHCCCCCH	CCCCCCCHHH	HHHHHHHHHE	CCHHHHHHHH
	Proposed FFA-ANN	HHHHHHHHHH	HHHHCCCCCH	CCCCCCCHHH	HHHHHHHHHC	CCHHHHHHHH

Sequence(201-237)		SIDQRTVKFH	IVNAMRKLNS	SNKAEATMKA	YAIGLLN
Structure	**DSSP**	TS*HHHHHHH	HHHHHHHTT*	SSHHHHHHHH	HHTT***
	MLNN	CCCHHHHHHH	HHHHHHHHCC	CCHHHHHHHH	HHHCCCC
	PSONN	CCCHHHHHHH	HHHHHHHHCC	CCHHHHHHHH	HHHCCCC
	Proposed FFA-ANN	CCHHHHHHHH	HHHHHHHHCC	CCHHHHHHHH	HHHCCCC

Fig 2 depicts the MSE of ANN for training, validation and test. With reference to this figure, the MSE is obtained at epoch 74 and it was the outstanding validation performance which is equivalent to 0.29.

Thus, the predicted secondary structure for the sequence of chain A of QscR using FFA tuned NN is given in the Table 2..

From the Table 2. the predicted H is about 66.35 and E is about 11.81, hence the proposed QscR is of mixed class rather than all helix or all beta.

Apart from the detection of SS of QscR, the impact of the arrangement of amino acids in predicted SS is investigated.

The results given by the maximum number of SS forecasting methods are based on the three familiar secondary structures.

For determining the secondary structure, the impact of the configuration and the physicochemical properties of amino acids are investigated here.

From SS, it is observed that constant amount of amino acids are not present. There exists some amino acids like Glutamine, Leucine and Isoleucine which are having highest count of helix residues. On the other hand Cysteine and Proline are having exact number of helix residues in it. Serine, Leucine and Phenylalanine look to bear highest content of strand. At the same time most, of the acids like Asparagine, Aspartic acid do not exist strand. Similarly, Cysteine and Glutamine do not exhibit coil property and huge count of debris in Serine and Proline which have coil structures, while Asparagine and Alanine seem to have the same content in the coil and the same is depicted in figure 3

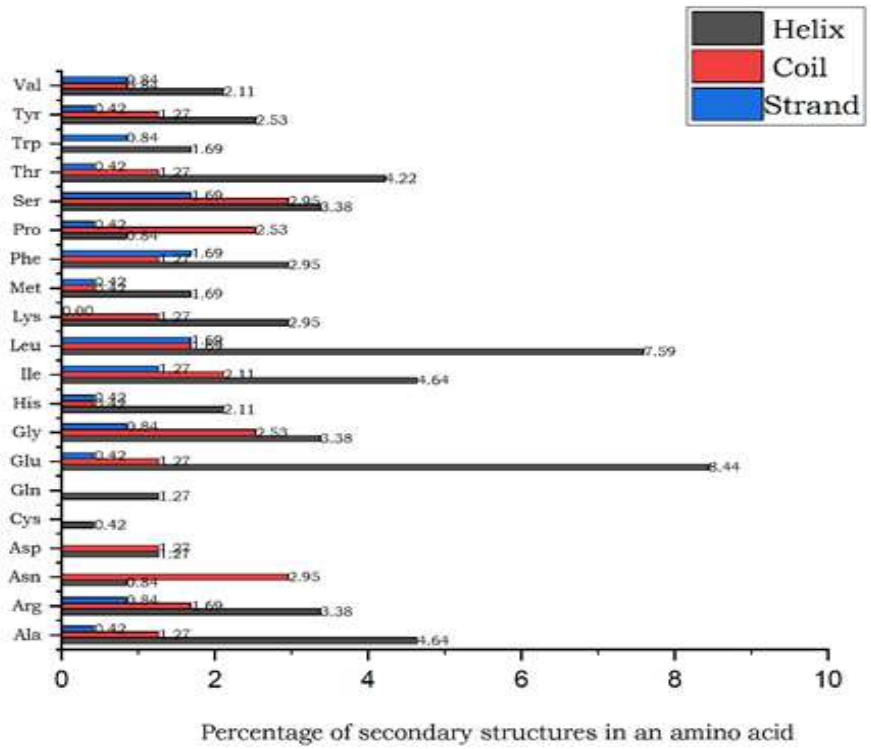

Figure 3a. AA contents presents in SS of QscR

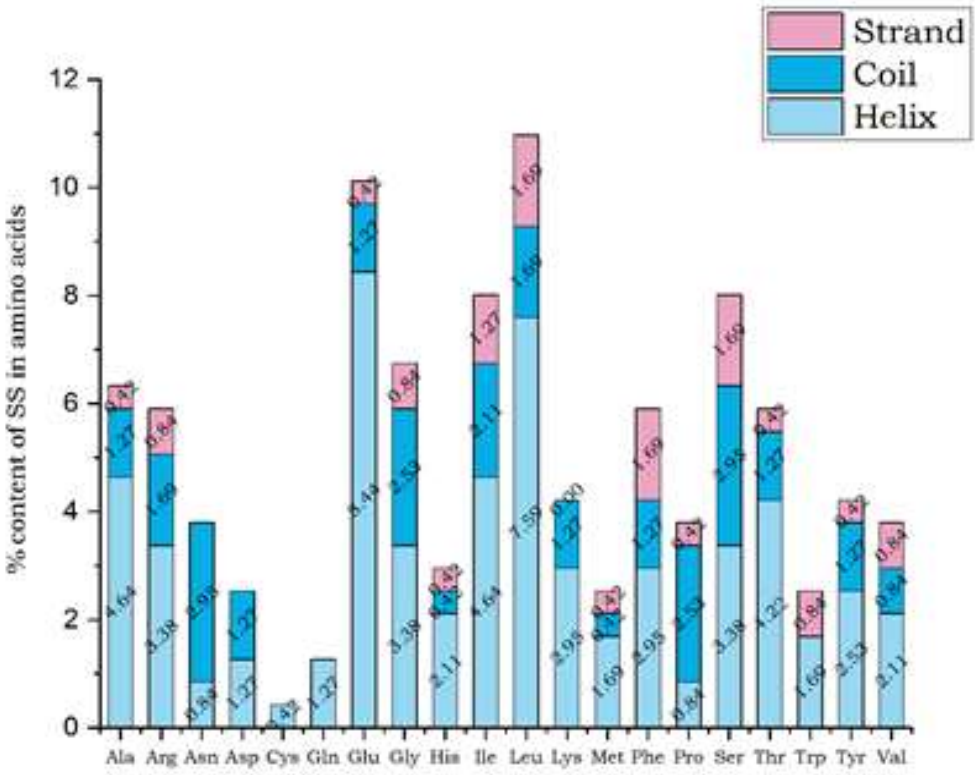

Figure 3b. Percentage of secondary structures in an amino acid

From the above graph, it is concluded that cys and gln are helix in nature and amino acids such as asn, glu and thr have not exhibit strand nature. From the analysis, it is concluded that α helix plays a vital role in QscR protein.

For determining the prediction accuracy, several methods have been used in this approach. The Q3 accuracy identifies the percentage of the residues which are predicted correctly. The percentage of accuracy of the commonly predicted residue in every type are represented by QH, QE, QC. SOV denotes the overlap of AA and is depicted in Table 3

Table 3. Performance analysis of Q3 and SOV of the proposed topology

	Q overall (%)	Q_H (%)	Q_E (%)	Q_C (%)
Q3	81.2	85.2	59.3	85.0
SOV	78.6	84.9	73.8	52.7

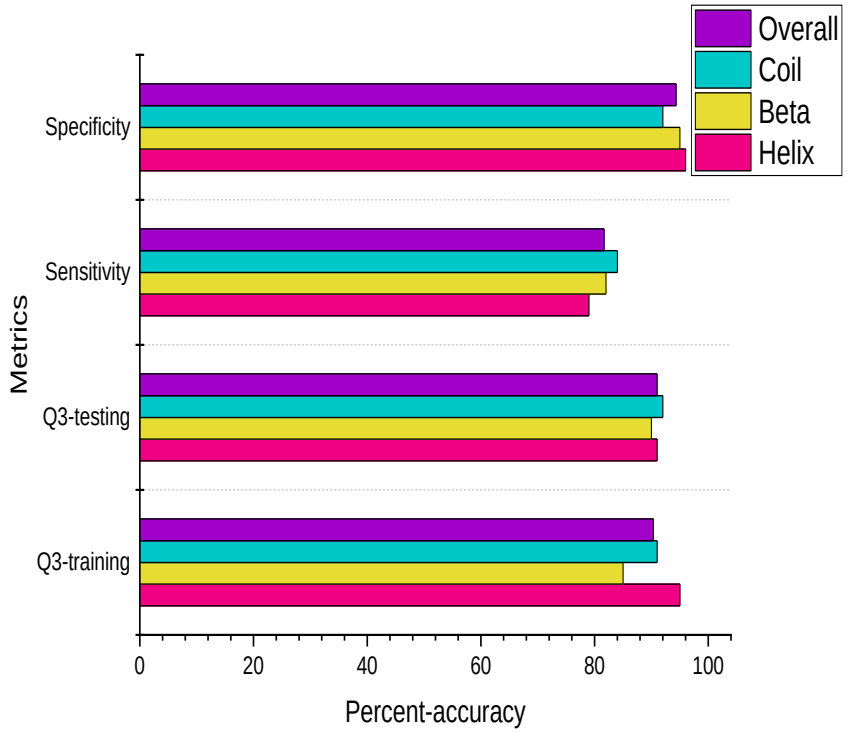

Figure 4. Accuracy of the proposed topology

For a set of protein which is tested, Q3 accuracy/training, PPVC, Mathew's correlation coefficient, specificity and sensitivity is shown in the figure 4. The classification performance can be assessed on the basis of TP, FP, TN and FN. Conforming to a pre-set threshold, the predicted class is determined using the estimated probabilities provided by the output of a classification. The ROC is formed by drawing a graph with TP rate and FP rate as co-ordinate pairs. The region below the ROC curve represents the cumulative performance of testing results. When a value is near 1.00, it indicates the absolute performance.

Analogy with other methods

Finally, results obtained using proposed methodology is compared with the performance of other networks which depicted the secondary structure of QscR.

Table 4. Comparative analysis of performance of the other methods in secondary structure prediction

Method	Alpha (%)	Beta sheet(%)	Coil
DSSP	54	13	-

STRIDE	54	11	-
MLNN	56.54	11.81	31.64
PSO-NN	59.07	12.23	28.69
FFA-NN	60.33	12.65	27

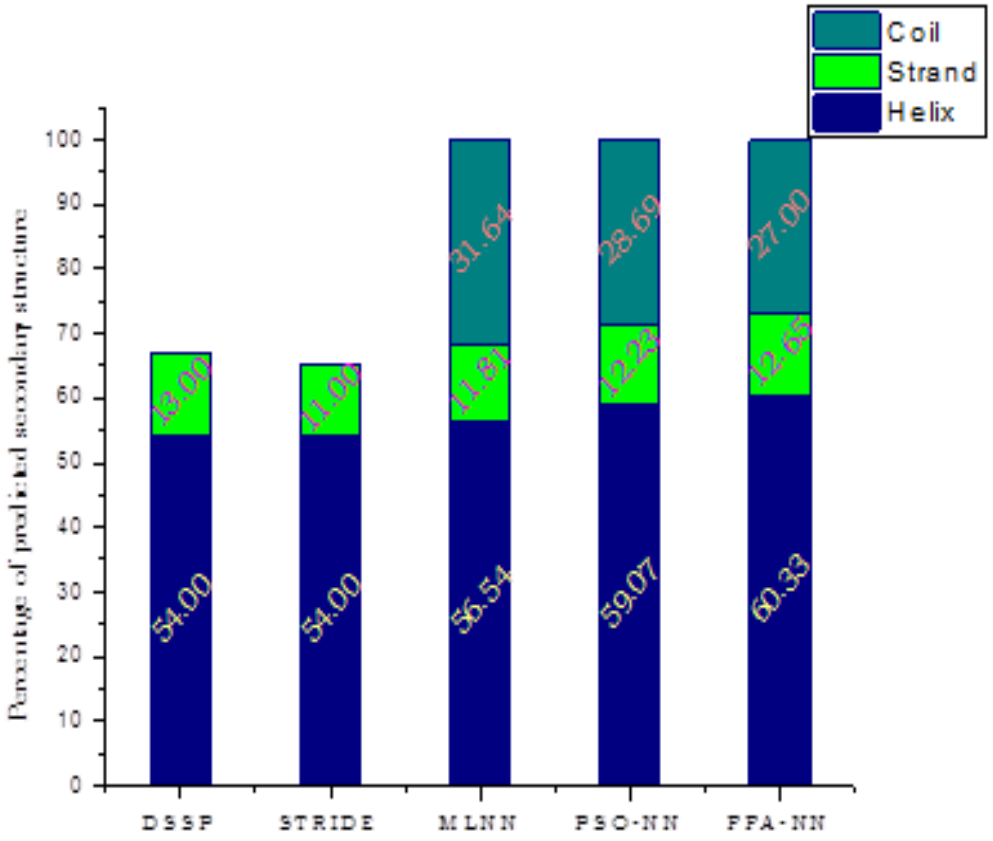

Figure 5. Comparative analysis of performance of the other methods in secondary structure prediction

From the tabulation 4 and from figure 5 it is concluded that the suggested PSO-NN gives more and better prediction of secondary structure than the other topologies.

Table 5. Performance comparison of Q3 and SOV of the proposed topology

	Qoverall (%)	QH (%)	QE (%)	QC (%)	Methodology
Q3	75.74	83.89	58.33	85	MLNN
SOV	75.6	85	72.9	42.6	
Q3	77.71	85.23	60.41	87.5	PSOANN
SOV	77.6	85.4	72.9	52.7	

Q3	78.66	86	62.5	87.5	FFAANN
SOV	78.6	84.9	73.8	52.7	

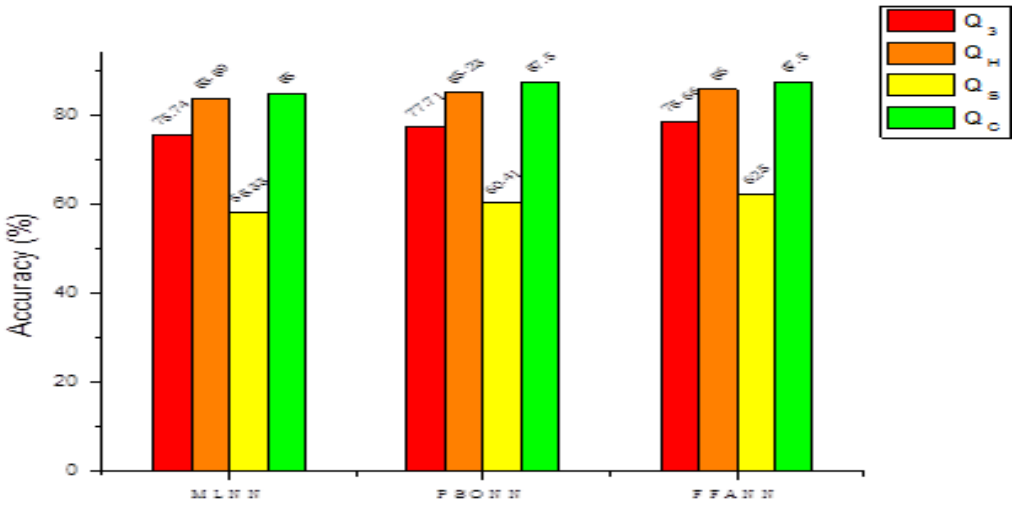

Figure 6. Q3 comparison with other topologies

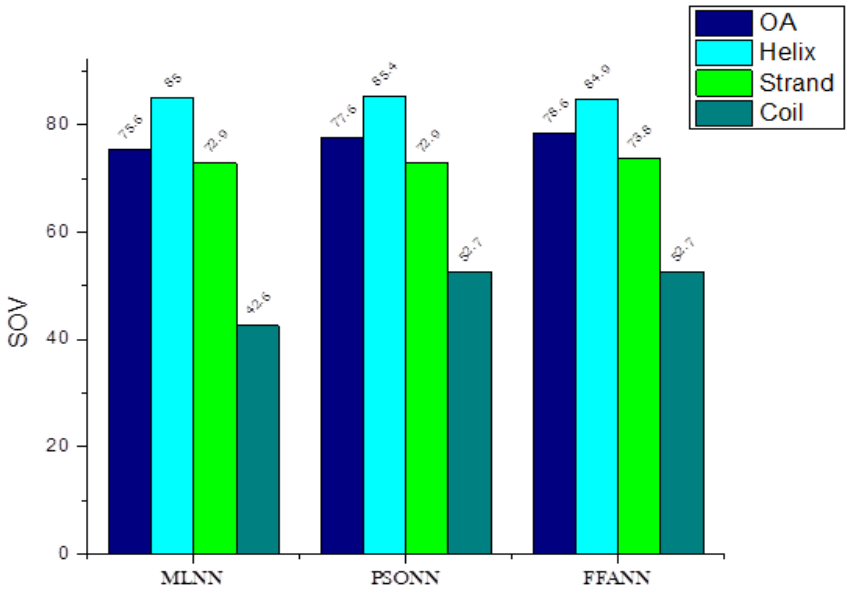

Figure 7. Overall SOV comparison with other topologies

From the overall comparison study, it is concluded that the proposed FFANN gives more and better prediction of secondary structure than the other topologies. Similarly, it exhibits greater accuracy and high per-residue accuracy than MLNN and PSONN topologies.

CONCLUSION

A new topology called firefly trained Neural Fields which can tune NN automatically is designed for protein SS prediction. In this topology, a novel method FFA based ANN is implemented to identify secondary structure of a protein. The recommended forecasting method revealed more favourable outcomes and this method surpasses several other state-of-the-art forecasting methods This scheme automatically tunes the neural network using optimization topology called firefly. It has achieved the accuracy about 97% on the independent dataset. The results obtained after the execution of the experiments have showed that the suggested method is a convenient one for the exploration of essential protein modifications and this method would be a dominant research domain in forecasting PSS.

REFERENCES

1. Bui, Dac-Khuong, Tuan Nguyen, Jui-Sheng Chou, H. Nguyen-Xuan, and Tuan Duc Ngo, "A modified firefly algorithm-artificial neural network expert system for predicting compressive and tensile strength of high-performance concrete", *Construction and Building Materials* 2018, 180, 320-333.

2. Samad, T.,"Back-propagation improvements based on heuristic arguments", Proceedings of International Joint Conference on Neural Networks, Washington, 1990, 1, pp. 565-568.

3. Ghorbani, M.A. , Shamshirband, S. , Zare Haghi, D. , Azani, A. , Bonakdari, H. , & Ebtehaj, I., "Application of firefly algorithm-based support vector machines for prediction of field capacity and permanent wilting point", *Soil and Tillage Research* , 2017, *172* , 32–38.

4. Moazenzadeh, R. , Mohammadi, B. , Shamshirband, S. ,& Chau, K. W., "Coupling a firefly algorithm with support vector regression to predict evaporation in northern Iran", *Engineering Applications of Computational Fluid Mechanics* , 2018, *12* (1), 584–597.

5. X.-S. Yang, "Firefly algorithms for multimodal optimization", in: Stochastic Algorithms: Foundations and Applications, SAGA 2009, Lecture Notes in Computer Sciences, 2009, Vol. 5792, pp. 169-178.

6. X.-S. Yang, "Firefly algorithm, L´evy flights and global optimization", in: Research and Development in Intelligent Systems XXVI (Eds M. Bramer, R. Ellis, M. Petridis), Springer London, 2010, pp. 209-218.

7. T. Apostolopoulos and A. Vlachos, "Application of the Firefly Algorithm for Solving the Economic Emissions Load Dispatch Problem", in: International Journal of Combinatorics, 2011, pp. 1-23.

8. F. Tangour and P. Borne, "Presentation of some Meta-heuristic for the Optimization of complex system", in: Studies in Informatics and Control , 2008, Vol.17, No(2), pp. 169-180.

9. H. Banati and M. Bajaj, "Fire Fly Based Feature Selection Approach", IJCSI International Journal of Computer Science Issues, 2011, Vol. 8(4), No 2, pp. 473- 480.

10. N. Chai-ead, P. Aungkulanon, and P. Luangpaiboon, "Bees and Firefly Algorithms for Noisy Non-Linear Optimisation Problems", International MultiConference of Engineers and Scientists (IMECS) Hong Kong, 2011, Vol. II.

11. A. A. Abshouri and M. R. Meybodi, “New Firefly Algorithm based On Multi swarm & Learning Automata in Dynamic Environments”, IEEE proceedings,2011.

12. A. Rajini, V. k. David, “A Hybrid Metaheuristic Algorithm for Classification using Micro array Data”, International Journal of Scientific & Engineering Research, 2012, Vol. 3(2), pp. 1-9.