International Journal of Psychosocial Rehabilitation, Vol. 24, Issue 02, 2020
ISSN: 1475-7192

# GENETICALLY OPTIMIZED GAIN RATIO FEED FORWARD NEURAL NETWORK ALGORITHM FOR REVIEW OPINION CLASSIFICATION

[1]Dr. Helen Josephine V L, [2]Dr. V.S Prakash, [3]Dr. Aruna Devi M

***ABSTRACT***-- *The opinion mining and sentiment analysis has become significant and vital area of web content mining and text mining and many researchers paying attention over the past decade . This research paper identified the importance Genetic optimization technique and used to optimize the neural network classifier algorithm Gain Ratio Feed forward Neural Network algorithm (GR_FFNN). The proposed Genetically Optimized GR_FFNNs algorithm used to optimize the vital parameter momentum and learning rate by using one of the leading soft computing approaches Genetic Optimization algorithms. Genetically optimized GR-FFNN algorithm was investigated with the mobile learning app reviews. It classifies the reviews into three different class based the opinion extracted along with their polarity. The methodology, architecture and the results of the proposed Genetically Optimization Gain Ratio FeedForward Neural Network (Genetically Optimized GR-FFNN) algorithm are discussed elaborately.*

***Keyword --*** *Genetic Algorithm, Classification, Machine Learning, Neural Network, Opinion Mining*

## I. INTRODUCTION

One of the major advantages of the internet in the recent years is that anyone can express their views about the products and services through reviews, blogs, feedback, and comments on the website and in social media. There are various algorithms and techniques existing to analyze the opinions, but no technique that can provide a perfect solution. In this research article, neural network based opinion classifier algorithm Gain Ratio Feed forward Neural network (GR-FFNN) algorithm [1] to be optimized in order to provide the best result. This research paper proposed an algorithm which would optimize the parameter of the GR-FFNN neural network classifier and provide best performance. This paper divided into 6 sections. Section II described about the motivation and background. Section III explained the Genetic Optimization Technique. Methodology and architecture of the proposed Genetically Optimzied GR-FFNN algorithm are discussed in Section IV. In Section V explained the impact of experimental results. Eventually, this research paper ended with Section VI.

[1]*Associate Professor, Department of Computer Applications, CMR Institute of Technology, Bengaluru, helenjose.cbe@gmail.com*
[2]*Assistant Professor, Department of Computer Applications, Kristu Jayanthi College, Bengaluru, prakashfuture@gmail.com*
[3]*Professor, Department of Computer Applications, Cambridge Institute of Technology, Bengaluru, arunamauriyan@rediffmail.com*

**Received: 02 Dec 2019 | Revised: 24 Jan 2020 | Accepted: 16 Feb 2020**

## II. MOTIVATION AND BACK GROUND

Multilayered feedforward neural networks acquire more number of properties which would suited and provide the optimal solution to complex pattern classification problems. In addition, neural networks are proficient to generalize, and generate perfect results for inputs which are not found in the training dataset. The efficiency of the neural network is determined by the number of neurons (size), how the neurons are connected (topology), the speed the neural network (learning rate), momentum and various other parameters. If the parameters are not chosen properly, two main problems namely over-fitting and unacceptable error might be taken place.

The Back propagation algorithm depends on the error gradient descent that the weight unavoidably falls into the local minimum points[2]. This algorithm some time even for larger dataset, gets stagnate and stuck into poor local minimum problem [3]. A global minimum is defined as the smallest value of all the local minima in feature space. Traditional optimization methods explore the neighborhood of a current solution and only choose new solutions which would firmly minimize the objective function value. Commonly this is the reason to trap in a local minimum, which would provide less chances of finding global minima. Hence, global optimization methods are required if one needs to search for a global minimum [4]. These methods are called meta-heuristics have turn into famous in for providing solution for computationally complex global optimization problems [5]. Genetic algorithms provides best solution for global optimization, where as neural networks are good for local optimization problem [6].

For optimization and learning task, there are two important techniques namely neural network and genetic algorithm. Each has its own pros and cons [7][8]. Genetic algorithms is one of the best optimization methods which finds smart way to discover global optimum value for outsized and complex space. So, GAs are well appropriate to train to get optimum value in feedforward and back propagation neural network [9][10]. The genetic optimization algorithm used to optimize the GR-FFNN algorithm parameter, which would avoid premature convergence of solution. GA is also very useful when the developer does not have accurate domain expertise, because GA acquires the ability to discover and learn from their domain.

## III. GENETIC OPTIMIZATION TECHNIQUE

Genetic algorithm was proposed by John Holland and it is a useful technique for search and optimization problems which is inspired from the natural genetics and natural selection. It is executed iteratively with the concept of evolutionary biology [11]. In genetic algorithm, genotype of an individual is set up with the values of its genes. Each and every gene is normally denoted as a binary number. The phenotype of a human being is the real meaning the genes represent. A population of individuals is primarily formed with all of their genotypes which are randomly selected. Once the primary population is formed, it is sorted by fitness.

Algorithm can be implemented to calculate the fitness value and choose the best fitness value to obtain best solution[12]. During each generation, the parents(two individuals) are randomly selected to reproduce, with the extremely fit individuals more possible to be selected. The genotypes of the parents are united to form a new offspring in a method known as crossover. In each iteration, genetic algorithm identifies a set of possible input which provides optimum solution in the problem search space. Fig. 1 shows the general structure of Genetic Algorithm.

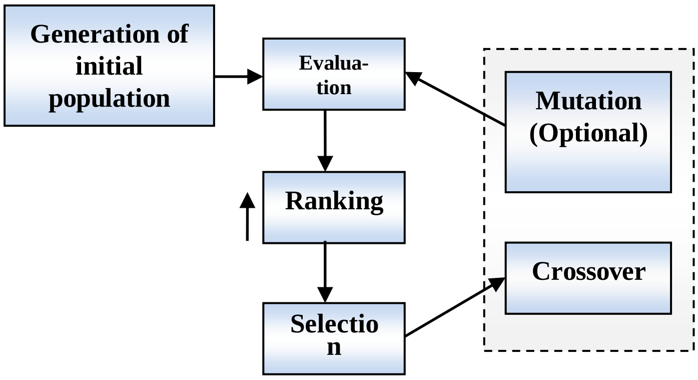


**Figure 1: Structure of Genetic Algorithm**

Generating of initial population, evaluation, ranking, selection and crossover are the various steps in genetic algorithm. The first step rivets arbitrarily generating the initial population. The second step is known as evaluation step; in this each entity in the set population (along with the newly created entity opening in the second iteration) is allotted a fitness value. In next step, all of the individuals are ranked depends on their fitness value. During the selection step, two different entities are chosen to reproduce dependent on their rank. The two special elected individuals' genotypes are used to create a new individual during the cross over step. At last, the new entity's genes may perhaps be mutated (bits are randomly flipped) and starts with next iteration.

ANN algorithm is the one of the best technique for supervised machine learning and regression based model in terms of computation and accuracy. The ANN encompasses various parameters and these parameters need to optimize with optimization algorithm to provide best result especially to improve the classification accuracy. The hybrid model, Genetic optimization technique combined with the ANN algorithm is the most reliable and computational intelligence technique is achieved better result than the traditional back propagation neural network.

Initial Population, Population size, Mutation probability and crossover probability are the most important parameter to implement GA. Usually population size is determined based on the problem's nature. First population is produced randomly is known as population initialization. After the initialization step, each and every solution is obtained by a fitness function which is associated to the objective function of the search and optimization problem. Three natural genetic operators namely selection or representation, crossover and mutation applied and the population of solutions is adapted to a new population. Each iteration is known as one generation. In each generation, these three operators are used successively to produce the next generation until the termination condition is fulfilled.

A Genetically optimized Algorithm has four basic steps:

1. Crete population (initial set of potential solutions).
2. Evaluation function to filter outcome solutions based on fitness.
3. Along with genetic composition of offspring apply Genetic operators (crossover, mutation and selection, etc.).
4. Parameter value s that genetic algorithm uses (population size, probabilities of applying genetic operators, etc.).

***A) SELECTION / REPRESENTATION***

Selection operator selects the individuals known as parents, which will be given to the population at next generation. It allows the random behavior to select next population. Selection associates to continued existence of the fittest [13]. Fitness is influenced by an objective function of the optimization problem or by a subjective judgment. According to the survival of the fittest principle, parents with superior fitness value have superior opportunity of being selected for subsequent reproduction generation. But the parents with lower fitness values have a lower opportunity of being selected for subsequent population. There are various methods are available for selection [14].

1. Roulette Wheel Selection: This type of selection supported on the probability of parenthood and it is proportional to fitness. The wheel is spun until two parents are selected. This selection has disadvantage if the fitness vary by order of magnitude. Since the chance of parents to be selected is depended on the relative fitness of the population [15].

2. Rank selection: In this method, all parents in population are ranked according to the fitness. Each parent is assigned a weight inversely proportional to the rank.

3. Tournament Selection: four parents are arbitrarily selected from the group. Two are removed and two individual become the parents to the child in the next generation [16].

***B) CROSS-OVER***

Cross-over function merges two pair of bit strings (parents) to form children to the next generation. This function is divided into three steps. First step, identifies the pair of strings will be merged randomly to become the parents of two new strings. Second step discovers the crossover point. Based on the crossover algorithm random number between 0 to the length of the string is generated [17]. Several crossover techniques namely single point, two point, multi point crossover; three parent and uniform are existed. Single point crossover algorithm is recognized as superior than other algorithms. In third step, the bits of two strings lying to the right of the crossover point are swapped to generate two new strings. (ie) copy all the bits before the crossover point from the initial string and facsimile all the bits subsequent to the crossover point from the second string.

***C) MUTATION***

*Mutation function operator* pertains random changes to individual entity (parents) to form offspring. It helps to prevent the GA from hanging into local minimum. The vital restriction in the mutation function is the mutation rate or mutation probability (Pm). It identifies how frequent parts of bit will be mutated. If the mutation likelihood is zero, offspring are generated directly copied subsequent to crossover without any modification. . Mutation probability determines convergence of the algorithm. If the mutation probability is high, then the diversity of the algorithm that is regarded as a good way for keeping off premature convergence. At the same time, it leads to divert the algorithm from converging on a global solution. Therefore, optimum value of the mutation probability is needed to provide best result. Based on the mutation probability mutation points are identified and the bits are applied any one of the mutation function. Mutation functions are flipping a bit, interchanging bits and reversing bit[18].

Flipping : Identify a random mutation point and flip
Or change the bit 0 to 1 or 1 to 0.

Interchanging: Identify two random positions and swapped
the bit.

Reversing : Identify a random position and the bits
Subsequent to that position are inverted [19].

## IV. METHODOLOGY AND ARCHITECTURE

This part of section describes the methods of the planned work. In this study, 300 learning app reviews (positive, negative and neutral) were collected from the Android Market Website for scrutiny. The user opinions about mobile learning system are classified using Gain Ratio Feedforward Neural Network with genetic optimization approach.

The efficiency of Artificial Neural network based on the choice of basic parameters namely weight, bias value, learning rate, momentum value. It determines the success of the training process. An optimization method requires setting optimum value for basic parameters to yield good results, exclusively for setting the learning rate and momentum value. Bio-inspired optimization methods are computationally efficient alternatives to analytical methods. Genetic Algorithms (GA) is a standard multidimensional search and optimization technique. The strengths of GA are optimizes both continuous and discrete function, high quality of list of good solutions and speed of coverage. GA provides computational efficiency even though the search space is very large and number of parameters involved.[20]

Learning rate and training duration are indirectly proportioned. If it makes very little updates to the weights in the network, then the learning progress becomes very slow. But at the same time, if the learning rate is set too high, it can result drastic updates which leads to unwanted divergent behavior in momentum. Hence optimum learning rate is required to swiftly reach to the minimum point[21][22].

A local optimum of a function is defined as the point in the domain of a function that calculates evaluates to a optimum value at each and every point in the region of the local optima. On the other hand, a global optimum of a function optimizes the function on its entire domain. It is not just on a neighborhood of the optimum [23]. While implementing neural network, it can easily get clung in local optimum and it might thought it reaches the global optimum leading to sub optimal result. The global optimum point will be achieved only if no local optimum point. To prevent this situation, the momentum term is objective function is used. Momentum takes ranges of values 0 and 1 that helps to jump from the local optimum value. If the momentum is set too low, it cannot come out from the local optima and it can slow down the progress of the training process. If the momentum is set too high, the convergence will occur fast. But the learning rate should be kept smaller to achieve good result. The momentum value needs to be optimized to yield best result.

By GA the learning rate and momentum values need to optimize to converge to the best solution. Hence, the genetic optimized technique along with neural network helps to enhance classification accuracy.

The proposed Genetically Optimized Gain Ratio Feedforward Neural algorithm find optimum value for learning rate and momentum using genetic optimization algorithm by applying selection, crossover and mutation function on the population. Table 1 describes the parameters of the Genetically Optimized GR-FFNN.

**Table 1** : Parameters' Description of Genetically Optimized GR-FFNN

| Parameters | Value |
|---|---|
| Population Size | 50 |
| Crossover | Single point crossover |
| Crossover Probability | 0.9 |
| Mutation | Uniform Mutation |
| Mutation Probability | 0.1 |
| Maximum Generations | 25 |
| Number of Epochs | 500 |
| Momentum Optimization | Lower bound : 0.4<br>Upper bound : 0.9 |
| Learning Rate Optimization | Lower bound : 0.1<br>Upper bound : 0.5 |
| Selection mechanism | Rank Selection |

The advantage of the Genetic optimized GR-FFNN is more efficient than the FFNN containing the same number of processing elements.

***Algorithm for Genetically Optimized GR-FFNN***

1. Initalization

Initialize weights $w_{ij}^{[1]}$ and $w_{ik}^{[2]}$ to small random values in the range of 0,1

Initialize $b_i^{[1]}$, $b_k^{[2]}$, learning rate α and momentum η

Assign values to the input layer $a^{[0]}$=X where X=$x_1,x_2,\ldots.x_j$

2. Check the stopping condition (x,t)

While condition is true, do

3. Receive the signal from neurons in the input layer and transmit them to the neurons in the hidden layer $Z^{[1]}$

. $Z_i^{[1]}$ is calculated by

$$Z_i^{[1]} = b_i^{[1]} + \sum_{i=0}^{n} w_{ij}^{[1]}\, a_j^{[0]}$$

4. Apply activation function $a_i^{[1]}$ over $Z_i^{[1]}$

$$a_i^{[1]} = \sigma(Z_i^{[1]}) = \frac{1}{1+e^{-Z_i^{[1]}}} \quad \text{(Sigmoid activation fn)}$$

5. Each neuron in the output layer $Z_i^{[2]}$ is evaluated, $Z_i^{[2]}$ is calculated by $Z_k^{[2]} = b_k^{[2]} + \sum_{i=0}^{n} w_{ik}^{[2]} a_i^{[1]} +$ (GRP)

Where GRP is the sum of topmost five positive and negative feature neuron values proceeding from the input to output layer. The top five values are calculated by the Gain Ratio.

$$Gain\ Ratio\,(S,A) = \frac{Information\ Gain\ (S,A)}{Splitinformation(S,A)}$$

$$Information\ Gain(S,A) = E(S) - \sum_{v \in value\ S(A)} \frac{|S_v|}{|S|} E(S_v)$$

$$E(s) = -\sum_{i=1}^{m} p_i \log p_i$$

$$SplitInformation(S,A) = -\sum_{v \in value\ S(A)} \frac{|S_v|}{|S|} \log_2 \frac{|S_v|}{|S|}$$

6. Calculate the output of hidden unit by applying its activation function over $Z_k^{[2]}$

$$a_{ki}^{[2]} = \sigma(Z_k^{[2]}) = \frac{1}{1+e^{-Z_k^{[2]}}} \quad \text{(Sigmoid activation fn)}$$

7. $a_{ki}^{[2]}$ value got from the output layer based on the feature. Loss function will be calculated based on the cross entropy function

$$\delta_k = -y \log \hat{y} - (1-y)\log(1-\hat{y})$$

where, $\hat{y}$ is the predicted target value and y is the actual target value.

8. The important parameter learning rate and momentum are optimized through the genetic optimization algorithm.

End-while

9. Based on Loss function value $\delta_k$, the weight of the neural network and bias values are updated. Each output neuron $a_{ki}^{[2]}$ updates weight and bias and optimized learning rate and momentum of the neural network.

In Layer 2, output layer

$$w_{ik}^{[2]}(\text{new}) = w_{ik}^{[2]}(\text{old}) + \alpha\delta_k a_i^{[1]} + \eta$$

$$b_k^{[2]}(\text{new}) = b_k^{[2]}(\text{old}) + \alpha\delta_k + \eta$$

In Layer 1, hidden layer

$$w_{ij}^{[1]}(\text{new}) = w_{ij}^{[1]}(\text{old}) + \alpha\delta_k a_j^{[0]} + \eta$$

$$b_i^{[1]}(\text{new}) = b_i^{[1]}(\text{old}) + \alpha\delta_k + \eta$$

10. Evaluate the stopping condition.
    a. The maximum number of epochs has reached.
    b. Actual output equals to the target value

The important parameters momentum and learning rates are genetically optimized in order to improve the classification accuracy. The pictorial representation of series of steps pursued in yielding the genetic optimization method is shown in Figure 2.

**Received: 02 Dec 2019 | Revised: 24 Jan 2020 | Accepted: 16 Feb 2020** 

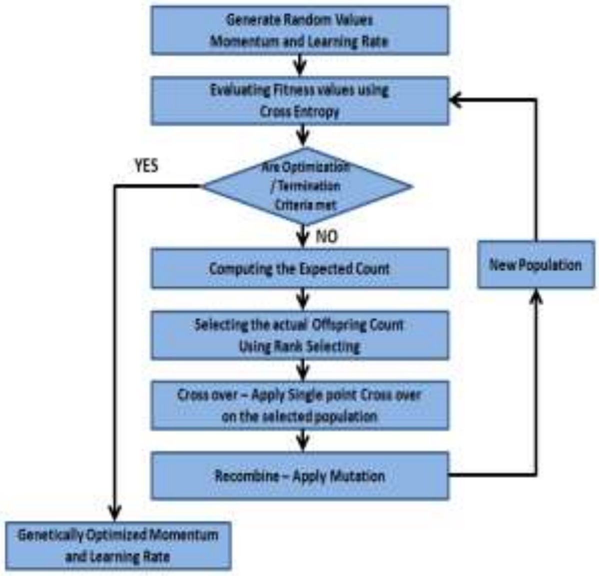


Figure 2: Genetic Optimization GR-FFNN Algorithm Steps

The procedure below resolves the generation of genetic optimized momentum and learning rates [24][25].

Procedure Genetic Optimization (Learning Rate, Momentum)

**t = 0 (t refers to the number of generations)**

**Initialize the population: P(0) = {a1(0), …, aμ(0)} Iμ**

**Evaluate: P(0): {Φ(a1(0)), …, Φ(aμ(0))} (Φ is the fitness function)**

**While (P(t)) true //Reproductive loop**

**Select: P’(t) = sθz {P(t)} (selects only the best fitness values)**

Recombine: P’’(t) = θz {P’(t)} (Cross over)

Mutate: P’’’(t) = mθm {P’’(t)}

Evaluate: P’’’(t): {Φ(a1’’’(t)), …, Φ(aλ’’’(t))}

Replace: P(t+1) = rθr (P’’’(t))

t = t + 1

End While

## V. RESULT AND DISCUSSION

Mobile learning app positive, negative and neutral reviews were gathered from the Android Market Website for experiment. The cleaned Mobile learning app review dataset has been obtained through SWSF (Significant Word, SVD, and Filtering) pre-processed algorithm [26][27]. This cleaned dataset has been fed into the proposed Genetically Optimized GR-FFNN algorithm. The classification accuracy achieved from the experiment is 88.33%. The experiments were compared with the existing ANN classifiers LVQ, Elman, FFNN and GR-FFNN. The classification accuracy and other classification metrics namely precision, recall and F-

International Journal of Psychosocial Rehabilitation, Vol. 24, Issue 02, 2020
ISSN: 1475-7192

Measure has evaluated and the values were compared. The GR-FFNN algorithm achieved the classification accuracy of 85.33%, whereas the other ANN classifiers LVQ gained 60.67%, Elman neural network obtained 79.67% and FFNN gained 82.33%. The classification accuracies are tabulated in Table 2.

**Table 2 : Comparing Classification Accuracy of GR- FFNN with existing Neural Network classifier**

| ***Algorithm*** | ***Classification Accuracy (in %)*** |
|---|---|
| ***LVQ*** | ***60.67*** |
| ***Elman NN*** | ***79.67*** |
| ***Feedforward NN*** | ***82.33*** |
| ***Proposed GR-FFNN*** | ***85.33*** |
| ***Proposed Genetically Optimized GR-FFNN*** | ***88.33*** |

The classification accuracy obtained from the GR-FFNN is 3% more than the existing FFNN. The classification accuracy has been improved because of the additional weight. The top five positive and negative features obtained by gain ratio formula fed to the output layer from the input layer.

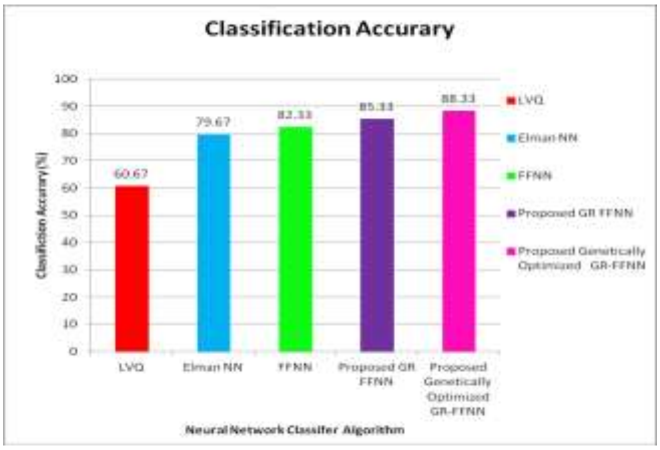


**Figure 3 :** Comparison of Genetically Optimized GR-FFNN with Existing Neural Network classifier

The comparative analysis it is evident to note that the proposed genetically optimized GR-FFNN classifier supersedes the existing ANN classifiers. The results are shown graphically in Figure 6.2. The classification accuracy is authenticated using Precision, Recall, and F-Measure. Precision reveals the chance that the retrieved document is relevant. Recall shows the likelihood that a relevant document in a search. Whereas, F-Measure is the harmonic mean of Precision and Recall.

The authenticate measures precision, recall, and f-measure are shown in Table 3 for FFNN and the proposed GR-FFNN classifiers. The measure of validation, precision, recall and f-measure are shown graphically in Figure.3. This tabulated value reveals the progress in the accuracy rate.

**Table 3 : Precision, Recall and F-Measure ofMobile learning app review dataset**

| Algorithm | Precision (in % ) | Recall (in % ) | F-Measure (in %) |
|---|---|---|---|
| **LVQ** | 62.135 | 61.375 | 61.753 |
| **Elman NN** | 78.855 | 80.333 | 79.587 |
| **FFNN** | 81.721 | 82.416 | 82.067 |
| **Proposed GR FFNN** | 84.909 | 85.666 | 85.083 |
| **Proposed Genetically Optimized GR-FFNN** | 87.506 | 88 | 87.752 |

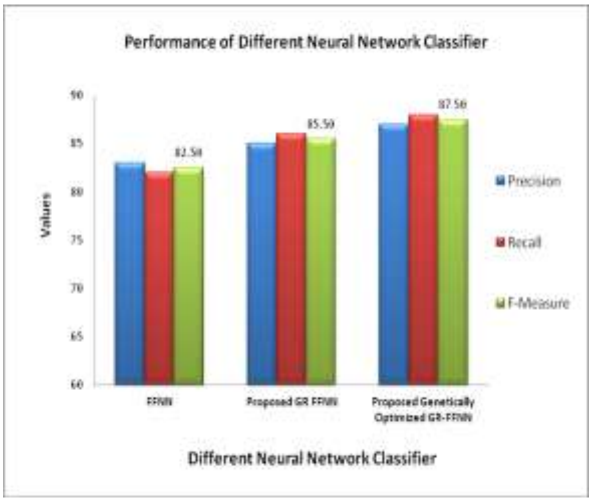


**Fig. 4 : Precision, Recall, and F –Measure ofMobile Learning App Review Dataset**

Thus, the proposed algorithm genetically optimized GR-FFNN is scrutinized against the start of art ANN classifiers along with the classification accuracy measures Precision, Recall and F-Measure. This result reveals that the proposed algorithm yields a better result than the existing ANN algorithm.

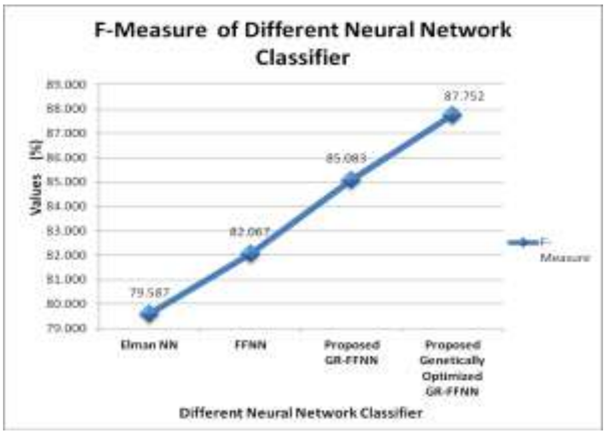


**Fig. 5 F –Measure Comparison Mobile Learning App Review Dataset**

The results of proposed genetically optimized GR-FFNN are compared with the GR-FFNN and other neural network classification algorithm. The proposed GR-FFNN vital parameter momentum and learning rate were genetically optimized which would bring the enhanced and the best result. The updated weights are again fed to the neural network to get the superior learning for the neural network. Genetically optimized GR-FFNN algorithm was investigated with the mobile learning app reviews. It classifies the reviews into three different class based the opinion extracted along with their polarity. The experimental results have shown that high accuracy and best classification performance metrics gained in genetically optimized GR-FFNN algorithm over the other existing ANN algorithms.

## VI. CONCLUSION

This research summarizes the proposed Optimized GR-FFNN algorithm to enhance the classification accuracy by optimizing its parameters with the help Genetic algorithm. The results of the proposed algorithms compared with the existing stat of art algorithms. The performance metrics classification accuracy, precision, recall and F-measure are evaluated for the proposed and existing algorithm. GR-FFNN algorithm has two vital parameter momentum and learning rate. These two parameters were genetically optimized which would bring the enhanced and the best result. The updated weights are again fed to the neural network to get the superior learning for the neural network. The experimental results have shown that high accuracy and best classification performance metrics gained in genetically optimized GR-FFNN algorithm over the other existing ANN algorithms.

## REFERENCES

1. Helen Josephine V.L and Dr. S. Duraisamy, “Gain Ratio Feed Forward Neural Network Algorithm to Improve Classification Accuracy”, International Journal Engineering & Technology (UAE) (IJET), ISSN : 2227-524X, Vol. 07, Issue - 4, pp.3579-3582, 2018.
2. N. Abdul Hamid, N. Mohd Nawi and R. Ghazali. (2011), The Effect of Adaptive Gain and Adaptive Momentum in Improving Training Time of Gradient Descent Back Propagation Algorithm on Classification problems, in Proceeding of the International Conference on Advanced Science, Engineering and Information Technology, pp. 178-184.
3. Haruna Chiroma, Ahmad Shukri Mohd Noor, Sameem Abdulkareem, Adamu I. Abubakar, Arief Hermawan, Hongwu Qin, Mukhtar Fatihu Hamza and Tutut Herawan, (2017), Neural Networks Optimization through Genetic Algorithm Searches: A Review, Applied Mathematics & Information Sciences An International Journal, vol.11, no. 6, pp,1543-1564
4. S. Vofl, S. Martello, I.H. Osman and C. Roucairol (1999), Advances and Trends in Local Search Paradigms for Optimization, Kluwer Academic, Boston, MA.
5. H. Maaranen, Miettinen, M. Makela, (2004), Quasi-Random Initial Population for Genetic Algorithms, Computer with Applications, vol. 47, pp. 1885-1895.

6. Y. Du and Y. Li, (2008), Sonar array azimuth control system based on genetic neural network, Proceedings of the 7th World Congress on Intelligent Control and Automation, pp. 6123-612.
7. D. Montana and L. Davis, "Training feedforward neural networks using genetic algorithms", published in "IJCAI'89 Proceedings of the 11th international joint conference on Artificial intelligence", pp 762-767 vol. 1, 1989.
8. H.Shi, Evolving Artificial Neural Networks Using GA and Momentum,2009 Second International Symposium on pp.475-478D.
9. Whitley, "Genetic Algorithms and Neural Networks", book title "Genetic Algorithms in Engineering and Computer Science", publisher "John Wiley, pp 191- 201", august 1995.
10. R.Mahajan, Gaganpreet Kaur(2013), Neural Networks using Genetic Algorithms, International Journal of Computer Applications, Vol 77, No.14, pp. 0975-8887.
11. K.M.Saridakis , A.J.Dentsoras Integration of fuzzy logic, genetic algorithms and neural networks in collaborative parametric design. Advanced Engineering Informatics 20 (2006) 379-399.
12. GoldbergD. Genetic algorithms in search optimization and machine learning. Massachusetts: Addison-Wesley;1989.
13. R. Akbari and Koorush Ziarati(2011), A multilevel evolutionary algorithm for optimizing numerical functions, International Journal of Industrial Engineering Computations, Vol.2, pp.419-430.
14. H. Chiroma, A.Shukri, M. Noor et.al.(2017), Neural Networks Optimization through Genetic Algorithm Searches : A Review, Vol.1564 Issue.6 pp 1543-1564.
15. Z. Michalewicz, Genetic Algorithms + Data Structures -- Evolution Programs, Springer-Verlag, (1994).
16. Goldberg, D.E., "A note on Boltzmann tournament selection for genetic algorithms and population-oriented simulated annealing" Complex Systems, 4 (1990) 445-460.
17. Eremeev, Anton V(2017), Hitting Times Of Local And Global Optima In Genetic Algorithms With Very High Selection Pressure, Yugoslav Journal of Operations Research 27, Number 3, pp. 323–339 DOI: 10.2298/YJOR160318016E.
18. Randall S. Sexton, Robert E. Dorsey, et. al., Optimization of Neural Networks: A Comparative Analysis of the Genetic Algorithm and Simulated Annealing, European Journal of Operational Research, Vol. 114, Issue 3, 1 May 1999, pp 589-601, https://doi.org/10.1016/S0377-2217(98)00114-3
19. Dorsey, R. E., Johnson, J. D., & Mayer, W. J., "A Genetic Algorithm for the Training of Feedforward Neural Networks," Advances in Artificial Intelligence in Economics, Finance and Management (J. D. Johnson and A. B. Whinston, eds., 93-111). Vol. 1. Greenwich, CT: JAI Press Inc., 1994.
20. Dharmistha D Vishwakarma (2012), Genetic Algorithm based Weights Optimization of Artificial Neural Network, International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering ISSN 2278 – 8875, Vol. 1, Issue 3, pp.206-211.
21. Randall S.Sexton, et. al.,(1998), Toward global optimization of neural networks: A comparison of the genetic algorithm and backpropagation, Decision Support Systems Vol 22, Issue 2, pp. 171-185. https://doi.org/10.1016/S0167-9236(97)00040-7.

22. Bumghi Choi,Ju-Hong Lee,Deok-Hwan Kim, (2008) Solving local minima problem with large number of hidden nodes on two-layer feed forward artificial neural network, Neuro computing vol. 71, issues 16–18, October 2008,pp. 3640-3643.
23. Norhamreeza Abdul Hamid, Nazri Mohd Nawi, Rozaida Ghazali, Mohd Najib Mohd Salleh, (2011), Solving Local Minima Problem in Back Propagation Algorithm Using Adaptive Gain, Adaptive Momentum and Adaptive Learning Rate on Classification Problems, International Conference Mathematical and Computational Biology, pp. 448-455.
24. O. Kaynar, S. Tastan andF. Demirkoparan(2010), Crude oil price forecasting with artificial neural networks, Ege Acad. Rev. 10, 2, 559-573.
25. D.T. Pham and D. Karaboga(1999), Training Elman and Jordan Networks for system identification using genetic algorithms. Artif. Intell. Eng. 13, 10-117.
26. Helen Josephine V L, S. Duraisamy, "Novel pre-processing framework to improve classification accuracy in opinion mining", International journal of computing, vol. 17, no. 4, pp. 234-242, 2018.
27. Richa Mahajan and Gaganpreet Kaur. “Neural Networks using Genetic Algorithms”, International Journal of Computer Applications 77(14):6-11, September 2013.