A novel approach for recommendation using optimized bidirectional gated recurrent unit

Prakash Pandharinath Rokade[1], Swati Babasaheb Bhonde[2], Prashant Laxmanrao Paikrao[3], Umesh Baburao Pawar[4]

[1]Department of Information Technology, S.N.D. College of Engineering and Research Center Yeola, Savitribai Phule Pune University, Maharashtra, India
[2]Department of Computer Engineering, Amrutvahini College of Engineering, Maharashtra, India
[3]Department of Electronics and Telecommunication, Government COE Amravati, Maharashtra, India
[4]School of Computing, Sandip University Nashik, Maharashtra, India

Article Info

Article history:

Received Sep 26, 2024
Revised Jul 3, 2024
Accepted Jul 12, 2025

Keywords:

Bidirectional gated recurrent unit
Feedforward neural network
Gated recurrent unit
Genetic algorithm
Multimodal deep learning

ABSTRACT

In today's world, every one of us refreshes our mood and gets energy through entertainment and enjoyment. Human nature is to provide feedback through ratings or comments for products used, services received, or films viewed. The recommendation system serves the user with recommendations based on historical stored information of user preferences. These systems amass information about the user in order to provide personalized experiences. These systems put efforts into delivering personalized experiences by accumulating information about the user. Hybrid algorithms are necessary to address the issues recommendation systems confront, which include low prediction accuracy, output that exceeds range, and inadequate convergence speed. This study suggests building a movie recommendation system using the remora optimization algorithm (ROA) and the bidirectional gated recurrent unit (BiGRU), the most recent version of the recursive neural network (RNN). The proposed method's results are compared with those of the genetic algorithm (GA), feed forward neural network (FFNN), and multimodal deep learning (MMDL). In terms of movie recommendation, BiGRU with ROA performs better than GA, MMDL, and FFNN.

Corresponding Author:

Prakash Pandharinath Rokade
Department of Information Technology, S.N.D. College of Engineering and Research Center Yeola, Savitribai Phule Pune University
Nashik, Maharashtra, India - 423401
Email: prakashrokade2005@gmail.com

1. INTRODUCTION

Social media is heavily used by people to share their opinions and feelings on many websites. These feelings manifest as opinions or ratings for a good or service. The greatest difficulty facing someone who has to make an online purchase in the information era is not only finding enough options or information but also deciding how to use the wealth of available data to their advantage [1]–[3]. This leads to the generation and analysis of massive amounts of data in order to forecast and suggest a user any product or service based on his interests. Movie rating databases comprising user ratings and various movie parameters may be found on a number of well-known websites, such as Kaggle. In order to address a problem, decisions supported by several stronger past impressions are always preferable to those produced by a single user impression [4], [5]. Only those users whose ratings are more strongly relevant to one another are gathered, as opposed to gathering all reviews and ratings [6]. Using various machine learning techniques, such as support vector machines (SVM), neural networks (NN), and genetic algorithms (GA), numerous academics have worked to improve recommendation systems [7]–[9]. In this study, we suggested a bidirectional gated recurrent unit (BiGRU) optimized remora optimization algorithm (ROA) based movie recommendation system. The 100 kB IMDB dataset is processed using the BiGRU method. Userid, movieid, and rating are the only parameters taken into account prior to preprocessing [10]. Similarity between users' ratings of the same movies and their ratings of their differences from one another are obtained. Finally, a user receives recommendations for the top 10 movies based on his interest pattern after the weights used to discover user similarity are optimized using ROA. The outcomes of GA, multimodal deep learning (MMDL), and feed forward neural networks (FFNN) are contrasted with the results. BiGRU appears to have produced better outcomes for every testing parameter we compared taken into account prior to preprocessing. Similarity between users' ratings of the same movies and their ratings of their differences from one another are obtained [11], [12]. K-nearest neighbor algorithm with collaborative filtering is used for movie rating prediction [13]. The long short-term memory using cyclic learning rate (LSTM-CLR) framework is used to identify cyberbullying on social media, improving classification accuracy without numerous trials and adjustments [14]. Finally, a user receives recommendations for the top 10 movies based on his interest pattern after the weights used to discover user similarity are optimized using ROA. The outcomes of GA, MMDL, and FFNN are contrasted with the results. BiGRU appears to have produced better outcomes for every testing parameter we compared [15]. Traditional recommendation systems, such as collaborative filtering and content-based filtering, often suffer from limitations including low prediction accuracy, slow convergence speed, and ineffective weight optimization. While deep learning models like FFNN and MMDL have shown improvements, they still struggle with optimal weight assignment and real-time adaptability. This paper introduces an optimized recommendation system using BiGRU with ROA to enhance accuracy and convergence speed. The main contributions of this paper are to propose of a BiGRU-based recommendation model optimized using ROA to enhance convergence speed and reduce training loss, to compare BiGRU with FFNN, MMDL, and GA to demonstrate superior performance and to implement proposed system on the IMDB movie dataset, achieving a significant improvement in prediction accuracy. The proposed system is particularly useful in movie recommendation systems, personalized content filtering, and e-commerce platforms, where real-time and high-accuracy predictions are crucial for user engagement.

2. METHOD

2.1. Proposed system architecture

The system architecture comprises input file to various models and then applying Remora and BiGRU for recommendation. As shown in Figure 1, the system architecture contains input file, the matrix representation of input file, co-occurrence matrix representation, constrain model, rating independent model, Bidirectional GRU, remora optimization and finally the recommendations.

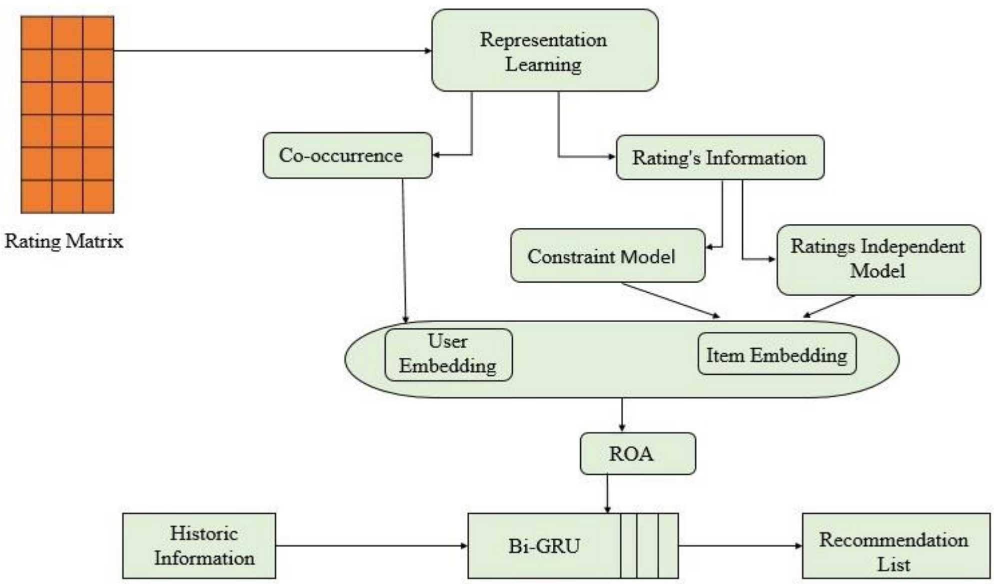

Figure 1. Architecture of proposed system [16]

2.2. Bidirectional gated recurrent unit

Bidirectional gated recurrent unit (BiGRU), a more recent RNN version than LSTM, is becoming increasingly popular these days. Because RNN performs the same operation at every time point, its calculation graph is extraordinarily deep [17]. A neural network's long- and short term memory strategy is suggested to address RNN problems, although it has a more complex structure and struggles to converge more quickly [18], [19]. The BiGRU outperforms the LSTM in terms of speed. The network configuration of BiGRU is the following: there are 2 GRU units, 2 dense layers and 2 dropout layers (*i.e.* totally 6 layers). Inside each GRU unit 1 reset and 1 update unit will be there [20].

As shown in Figure 2, GRU1 is a forward GRU, and Figure 2 displays its internal features, while Figure 3 displays the internal details of GRU2, a reverse GRU. The forward calculation shown in Figure 3 is carried out as follows. Suppose at time t, $\overrightarrow{r_t}$ is the reset gate of the positive input GRU. Here is the formula:

$$\overrightarrow{r_t} = \sigma(\overrightarrow{W_r}\overrightarrow{x_t} + \overrightarrow{U_r}\overrightarrow{h_{t-1}} \tag{1}$$

In the formula, σ is the sigmoid function, $\overrightarrow{x_t}$ and $\overrightarrow{h_{t-1}}$ are the values of the most recent activation and the current input, correspondingly. W_r is the input weight matrix. $\overrightarrow{U_r}$ is the weight matrix for cyclic connections. Similarly, suppose $\overrightarrow{z_t}$ is the update gate of the forward GRU at time t; the formula is as:

$$\overrightarrow{z_t} = \sigma(\overrightarrow{W_z}\overrightarrow{x_t} + \overrightarrow{U_z}\overrightarrow{h_{t-1}} \tag{2}$$

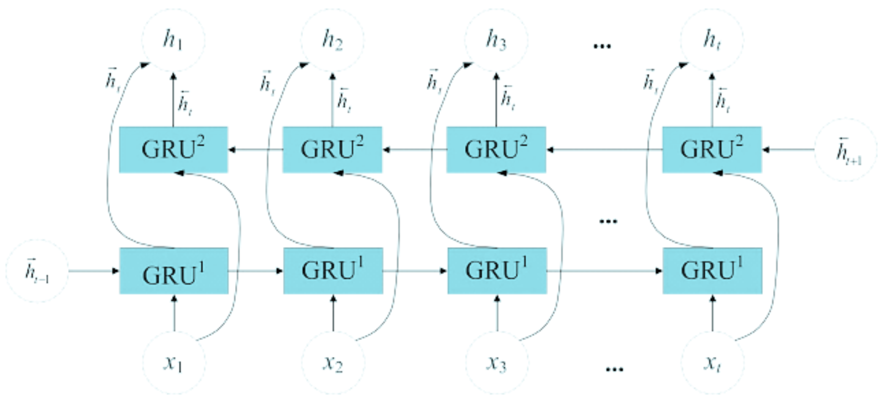

Figure 2. BiGRU working architecture

Suppose at time t, $\overrightarrow{h_t}$ is the activation value of the positive input GRU, which is a compromise between the candidate activation value $\overrightarrow{h_{t-1}}$ and the last activation value $\overrightarrow{h_{t-}}$.

$$\overrightarrow{h_t} = (1-\overrightarrow{z_t})\cdot \overrightarrow{h_{t-1}} + \overrightarrow{z_t}.h_{t-} \tag{3}$$

The formula for h_{t-} is as (4).

$$\widetilde{h_{t-}} = \tanh(\overrightarrow{W_h}.\overrightarrow{x_t} + \overrightarrow{r_t}.\overrightarrow{U_h}\overrightarrow{h_{t-1}}) \tag{4}$$

In the formula (4), is the Hadamard product.

For reset gate, if $\overrightarrow{r_t}$ is closed means its value approaches 0, the GRU eliminates the previous activation value $\overrightarrow{h_{t-1}}$ and the current input $\overrightarrow{x_t}$ is the only factor affecting it. This allows $\tilde{h}_t$ to deny irrelevant information, thereby more effectively communicating pertinent facts [21]. On other side, the update gate $\overrightarrow{z_t}$ controls how much information in $\overrightarrow{h_{t-1}}$. can be delivered to the current $\overrightarrow{h_t}$. This is the key to designing the results for this unit. It functions as a memory unit akin to an LSTM, aiding GRU in remembering long-term data [22]. Similarly, formula (5)–(8) provide the computation method for the reverse GRU shown in Figure 3 and Figure 4.

$$\overleftarrow{r_t} = \sigma(\overleftarrow{W_r}\,\overleftarrow{x_t} + \overleftarrow{U_r}\,\overleftarrow{h_{t+1}}) \tag{5}$$

$$\overleftarrow{z_t} = \sigma(\overleftarrow{W_z}\,\overleftarrow{x_t} + \overleftarrow{U_z}\,\overleftarrow{h_{t+1}}) \quad (6)$$

$$\overleftarrow{h_t} = (1 - \overleftarrow{z_t})\cdot\overleftarrow{h_{t+1}} + \overleftarrow{z_t}\,\overleftarrow{h_{t+1}}. \quad (7)$$

$$\widetilde{h_{t+}} = \tanh(\overleftarrow{W_h}\,\overleftarrow{x_t} + \overleftarrow{r_t}\,.\overleftarrow{U_h}\,\overleftarrow{h_{t+1}}) \quad (8)$$

The results of two directions are average to obtain final output h_t.

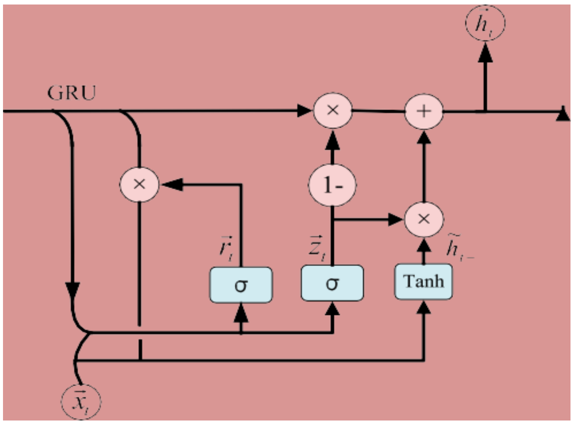

Figure 3. Inner structure of a forward GRU neuron

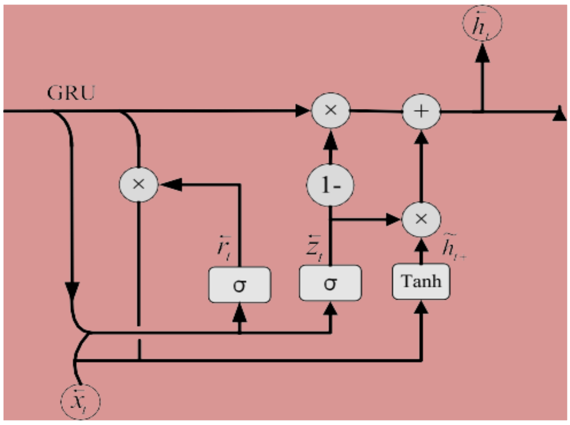

Figure 4. Inner structure of a backward GRU neuron

2.3. Remora optimization algorithm

ROA is an optimization algorithm that uses metaheuristics and is inspired by the foraging behavior of remora species. ROA, inspired by remora fish behavior, dynamically adjusts weight values in the BiGRU network, reducing training errors and accelerating convergence. It ensures optimal weight selection to maximize the accuracy of recommendations. The main intention of using ROA is to optimize the weight values in BiGRU and to result in accurate outcomes. ROA plays an important role to optimize input weights if the output at neuron exceeds the range 0 to 1. The ROA optimization algorithm is used to fine-tune the Bi-GRU's settings. This is accomplished by repeatedly looking for the ideal weight values while minimizing inaccuracy or loss. Therefore, the optimization algorithm's fitness function is the reduction of loss/error in Bi-GRU. As shown in Figure 5, the memory optimization algorithm consists of the following steps [23], [24], [25].

2.3.1. Flowchart of remora

Remora optimization technique is inspired by symbiotic relationships in nature of remoras and sharks. To balance exploration and exploitation It combines global and local search strategies. Agents (remoras) follow and adapt to leaders (sharks) in the population to find optimal solutions. It is commonly used in solving complex, nonlinear optimization problems. The method is lightweight, converges fast, and suits real-world engineering applications. Remora optimization algorithm flowchart is designed as follows.

a. Create the first population

   In this case, the search agent's parameter is population. We must initialize the number of remora, or search agents, in our system. The search agent is remora.

b. Define network weight

   The weight values of neural networks will be defined here. These weight values are nearly zero and are created at random.

c. Modification of search agents

   We first define a solution space in all optimization techniques. We also specify the number of search agents and the search area. We change the value of search agents if their number in a search space surpasses a threshold.in a search space, only search agents below the threshold will be permitted entry.

d. Error reduction

   In this case, the mistake is reduced by the error or loss function.

e. The current search agent's position and the store's fitness
Several network weights are available in the search space. We shall maintain a record of the optimal network weight.

f. Determine the nearest optimal weight
We can get the next weight value, or optimal weight, by computing the fitness value of each weight value we initially chose. The first weight value is known. Every iteration's fitness function is being evaluated.

g. Analyze fitness (error minimization)
For every weight in the network, we find its fitness value. A comparison of the new and old weight values is presented.

h. Error minimization
Here, the mistake is minimized by using the error function or loss function.

i. Store fitness and position of current search agent:
There are several network weights in the search space. The ideal network weight will be kept on file.

j. Find near optimal weight
The first weight value is known, and by calculating the fitness value of each weight value we initially selected, we can get the next weight value, or optimal weight. We are assessing the fitness function for each iteration.

k. Evaluate fitness (error minimization)
Every network weight is assessed for its fitness value. The new weight value and the old weight value are being compared.

l. Stopping criteria
First, we will define the number of iterations that will be used as our cutoff point. The goal is to increase the correctness of the network model by choosing the weight values for each iteration of the remora algorithm in an optimal manner.

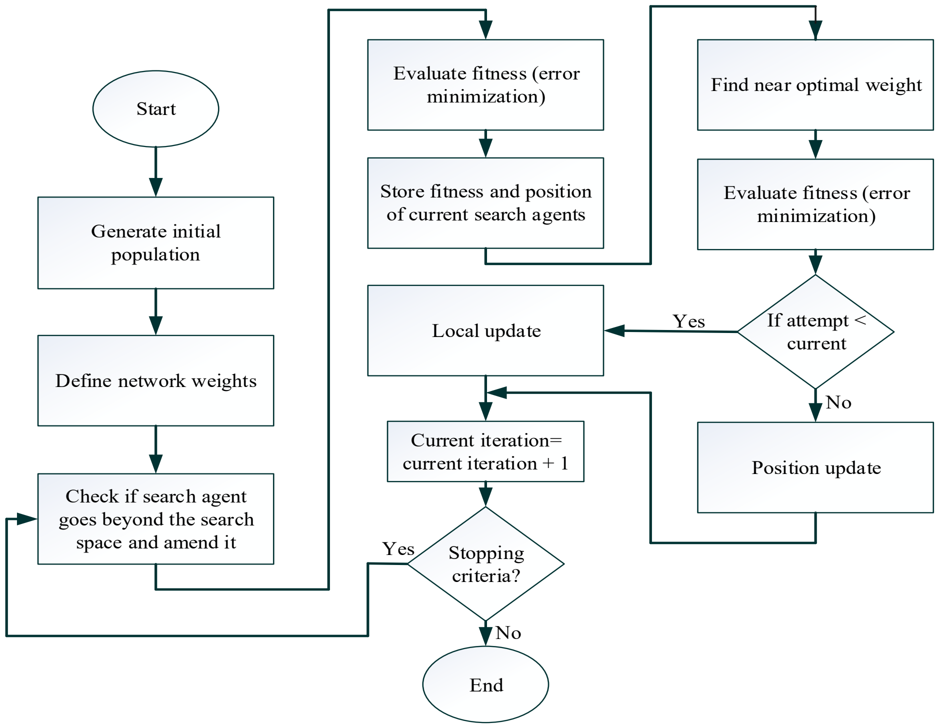

Figure 5. Remora optimization algorithm flowchart [26]

2.4. Input file uploading

MovieLens dataset of 150.35 kB is taken as an input from *Kaggle.com*. The dataset comprises 100,000 ratings captured from 671 users for a total of 1,682 movies. The input data contains the features userid, movieid, ratings, Input file contains 100,000 entries. It comprises the movie ratings along with the title, genre, time stamp for movies. About 671 users have rated with different ratings between 1 to 5. The overall size of the dataset is 150.35 kB. The input data contains the features userid, movieid, ratings, title, genre, timestamp, tag, imdbid, tmdbid. Only four features userid, movieid, moviename, ratings are considered in proposed model as an input.

As shown in Table 1, 671 users have rated 1,682 movies with ratings in between 1 to 5. It is not the case that all users have rated all the movies. Some users have not rated few movies. Either they have not seen those movies or they are not interested in providing the ratings to those movies. Consider user 1 has rated movie number 1343 with ratings 2 and this movie belongs to the genre action/adventure/thriller.

Table 1. Input file

Userid	Movieid	Rating	Timestamp	Title	Genre
1	31	2.5	1.26E+09	Toy Story	Adventure | Animation
1	1029	3	1.26E+09	Jumanji	Adventure
1	1061	3	1.26E+09	Grumpier Old Men	Comedy | Romance
1	1129	2	1.26E+09	Waiting to Exhale	Comedy | Drama | Romance
1	1172	4	1.26E+09	Father of Bride-II	Comedy
1	1263	2	1.26E+09	Heat	Action | Crime | Thriller
1	1287	2	1.26E+09	Sabrina	Comedy | Romance
1	1293	2	1.26E+09	Tom and Huck	Adventure Children
1	1339	3.5	1.26E+09	Sudden Death	Action
.	.	.	.	.	.
671	5669	4	1.06E+09	Dracula	Comedy Horror
671	5816	4	1.07E+09	Balto	Adventure Animation
671	5902	3.5	1.06E+09	Nixon	Drama

2.5. Reduction of sparsity issue

Dimensionality reduction is one of the good solutions for sparsity reduction. In this step, the sparsity issue in the input file is resolved by considering only the parameters userid, movieid, ratings. By considering only these limited parameters, the complex calculations are minimized and sparsity issue is suppressed. As shown in Table 2, ratings for movies by different users are shown in matrix format showing that user ratings for movie rated. There is no rating provided for unrated movies.

User id 12 has provided ratings 2, 2, 4.5, 4, 3, 2.5 for the movies 1, 2, 5, 6, 10, 11 respectively. There is empty space in the matrix by the user id 12 for the movies 3, 4, 7, 8, 9, 12 as it has not rated those movies. This experiment is evaluated for all users and movies but only for 12 users and movies matrix is designed.

Table 2. Matrix representation of ratings for different movies by users

Userid/Movieid	1	2	3	4	5	6	7	8	9	10	11	12
1												
2										4		
3												
4										4		
5												
6												
7	3											
8												
9	4											
10												
11												
12	2	2			4.5	4				3	2.5	

2.6. Co-occurrence model

There are few movies which are either rated by a single user, some movies are rated by all users. Co-occurrence model is obtained by intersection operator. It shows number of similar movies rated by two users. A co-occurrence matrix is generated to capture relationships or patterns between items (or users), which helps in understanding associations.

Table 3 shows the co-occurrence model in which if two users have rated same movies then they are included into the matrix. Out of 671 users for first 12 users only the matrix is prepared. Here user 7 has rated 5, 9, 11, 41, 10 same movies with user 1, 2, 3, 4, 5, 6 and so on. This shows co-occurrence model of similar movies rated by two users. Co-occurrence model is the part of input to the BiGRU algorithm for through calculation for predicting movies to the user.

Table 3. Co-occurrence model of similar movies rated by two users

Userid/Userid	1	2	3	4	5	6	7	8	9	10	11	12
1	20	76	0	5	1	0	5	18	27	18	10	1
2	76	8	8	9	0	17	12	6	3	6	2	4
3	0	8	51	7	12	3	11	5	12	0	0	53
4	5	9	7	204	19	7	41	4	2	0	243	120
5	1	0	12	19	100	4	10	4	7	0	133	0
6	0	17	3	7	4	44	0	16	0	60	54	32
7	5	9	11	41	10	0	88	80	21	2	0	91
8	18	6	5	4	4	16	80	12	6	3	6	2
9	27	2	12	2	7	0	21	6	0	101	33	15
10	18	2	0	0	0	60	2	3	101	0	33	72
11	10	1	0	243	133	54	0	6	7	33	2	77
12	1	8	53	7	0	32	91	2	15	72	77	12

2.7. Constraint model

Users who have given similar movies the same ratings are taken into consideration to create this model. This model is the restricted version of co-occurrence model. From the input matrix representation, it is clear that there are some users who have rated similar movies by similar ratings with the other users and few users are there who have rated similar movies by different ratings with the other users.

As shown in Table 4, user 7 has rated 2,3,5,21,5,0 similar movies by the same ratings as users 1, 2, 3, 4, 5, 6, respectively. The model is obtained for all users, but for the first 12 users, Table 4 is prepared. This model can be termed as a dependent model, as similar movies with the same ratings by users are considered here.

Table 4. Constraint model of similar movies with same ratings by two users

Userid/Userid	1	2	3	4	5	6	7	8	9	10	11	12
1	12	8	0	2	0	0	3	8	20	6	8	1
2	26	2	6	1	0	5	9	4	1	3	1	2
3	0	4	48	3	2	1	6	3	6	0	0	27
4	3	4	2	100	9	3	20	2	1	0	100	60
5	0	0	2	9	50	2	5	2	3	0	33	0
6	0	5	1	3	2	20	0	8	0	30	20	14
7	3	9	6	20	5	0	8	20	7	1	0	17
8	8	4	3	2	2	8	20	9	1	0	3	0
9	20	1	6	1	3	0	7	1	0	61	17	8
10	6	3	0	0	0	30	1	0	61	0	10	29
11	8	1	0	100	33	20	0	3	17	10	0	30
12	1	2	27	60	0	14	17	0	8	29	30	5

This module applies certain predefined constraints (*e.g.*, user behavior rules, diversity constraints, etc.) to guide the learning or filtering process.

2.8. Rating independent model

This approach takes into consideration viewers who have given similar movies varied ratings. As shown in Table 5, user 7 has rated 3, 9, 6, 20, 5, 0 similar movies by different ratings with the users 1, 2, 3, 4, 5, 6 respectively. This model is evaluated for all 671 users out of which for first 12 user's values are shown in Table 5.

Table 5. Rating independent model of similar movies with different ratings by two users

Userid/Userid	1	2	3	4	5	6	7	8	9	10	11	12
1	8	68	0	3	1	0	2	10	7	12	2	0
2	50	6	2	8	0	12	3	2	2	3	1	2
3	0	4	3	4	10	2	5	2	6	0	0	26
4	2	5	5	104	10	4	21	2	1	0	143	60
5	1	0	10	10	50	2	5	2	4	0	100	0
6	0	12	2	4	2	24	0	8	0	30	34	18
7	2	3	5	21	5	0	80	60	14	1	0	77
8	10	2	2	2	2	8	60	3	5	3	3	2
9	7	7	6	1	4	0	14	5	0	40	16	7
10	12	3	0	0	0	30	1	3	40	0	23	43
11	2	1	0	143	100	34	0	3	16	23	2	47
12	0	2	26	60	0	18	77	2	7	43	47	7

A model that operates without directly depending on user-provided ratings—possibly using implicit feedback, content similarity, or contextual signals

3. RESULT AND DISCUSSION

3.1. Testing for recommendation

Co-occurrence model, constraint model, rating independent model are the inputs to BiGRU algorithm. By considering random input multiplied with weights followed by addition of bias value at each neuron in hidden layer, the output is calculated. If the output at these neurons is between the ranges 0 to 1, these weights will be forwarded to next layer neurons. If the output at hidden layer neuron is beyond the range 0 to 1, remora optimization algorithm will be active to optimize these weights. Finally, by testing the proposed model for any random user, the recommended movie list will be suggested to the user. The output as recommended movies are shown in Figure 6.

Figure 6 shows that when we select any user for recommending him some movies, the interest pattern of user is already studied and based on that patter he will be recommended some movies. Here we can provide the threshold for getting first top x number of movies as result.

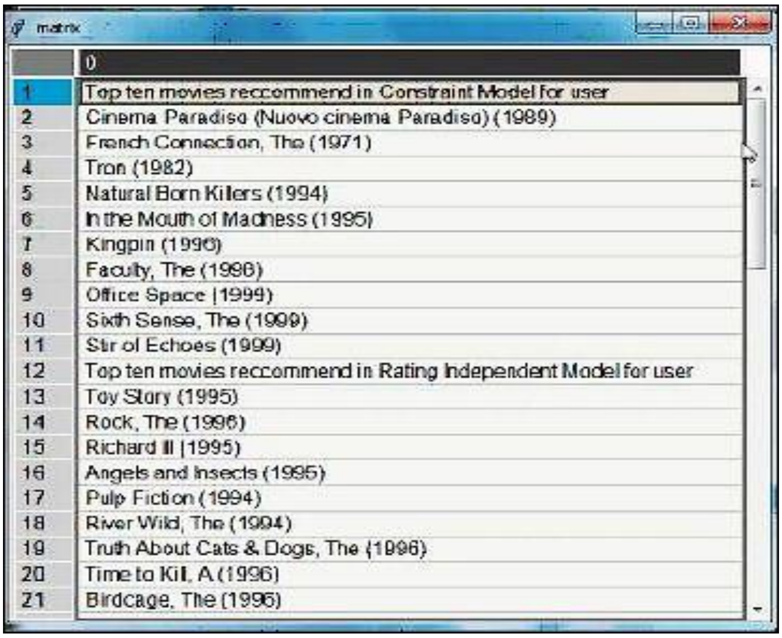

Figure 6. Recommendation of movies for random user

3.2. Result analysis of proposed system

The performance of the suggested system is assessed using testing measures like precision, recall, f measure, accuracy, mean absolute error, and root mean square error. These values are ascertained by obtaining a confusion matrix.

3.2.1. Proposed system comparison with existing system

To compare the performance of proposed model for the input dataset, the performance criteria are applied to MMDL, GA and FFNN for the same input and results are observed. As shown in Figure 7, proposed optimized BiGRU model for movie recommendation has the 98% precision which is the highest as compare to MMDL with 92%, GA with 96% and FFNN with 85%.

As shown in Figure 8, proposed optimized BiGRU model for movie recommendation has the 97.5% recall which is the highest as compare to MMDL with 96%, GA with 97% and FFNN with 88%.As shown in Figure 9, proposed optimized BiGRU model for movie recommendation has the 97% f-measure which is the highest as compare to MMDL with 95%, GA with 96% and FFNN with 87%.As shown in Figure 10, proposed optimized BiGRU model for movie recommendation has the 98.5% accuracy which is the highest as compare to MMDL with 95%, GA with 96% and FFNN with 86%.As shown in Figure 11, proposed optimized BiGRU model for movie recommendation has the 0.03 MAE which is the lowest as compare to MMDL with 0.65, GA with 0.05 and FFNN with 0.14.As shown in Figure 12, proposed optimized BiGRU model for movie recommendation has the 0.17 RMSE which is the lowest as compare to MMDL with 0.25, GA with 0.22 and FFNN with 0.36.

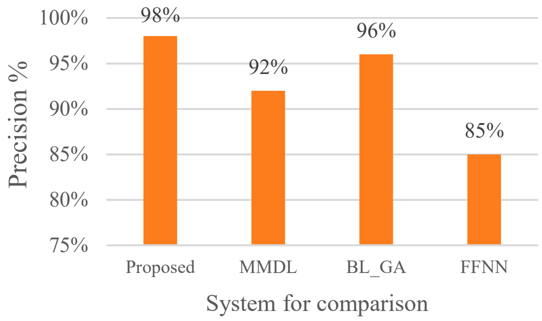

Figure 7. Comparison of precision for proposed system with existing systems

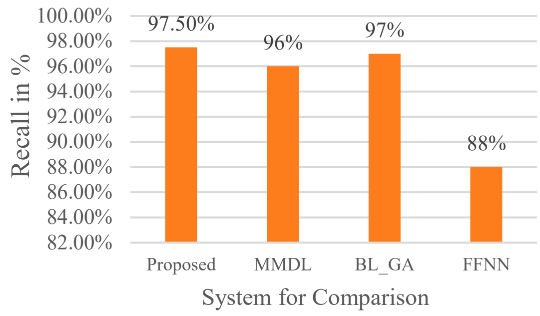

Figure 8. Comparison of recall for proposed system with existing systems

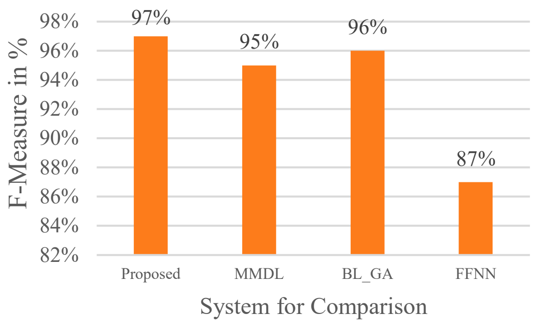

Figure 9. Comparison of f-measure for proposed system with existing systems

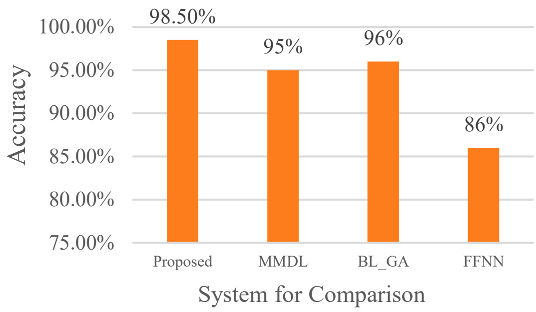

Figure 10. Comparison of accuracy for proposed system with existing systems

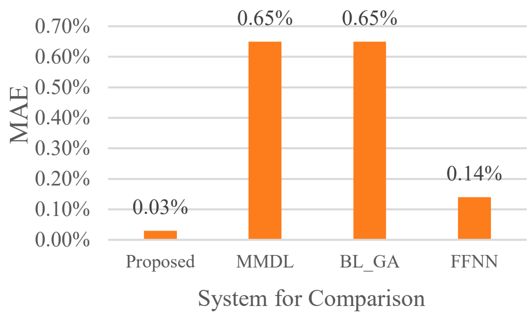

Figure 11. Comparison of MAE for proposed system with existing systems

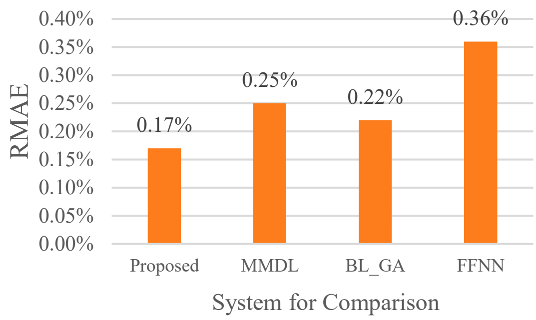

Figure 12. Comparison of RMSE for proposed system with existing systems

3.3. Experimental results and comparisons

To evaluate the effectiveness of the proposed BiGRU-ROA model, we compared its performance with three existing approaches — GA, MMDL, and FFNN — across four key metrics: precision, recall, F-measure, and accuracy. These metrics offer a comprehensive assessment of model performance in terms of both predictive accuracy and robustness. The following bar chart clearly illustrates that the BiGRU-ROA model consistently outperforms other models in each category, showcasing its superior precision (98%), recall (97.5%), F-measure (97%), and accuracy (98.5%).

As shown in Table 6, the values of precision, recall, f-measure, accuracy, MAE, RMSE for existing models MMDL, GA, FFNN and proposed optimized BiGRU model are calculated. The results of the proposed BiGRU-ROA model show an accuracy improvement of 12.3% over FFNN and 8.5% over GA, 32% faster convergence rate compared to traditional neural networks, reduction in training error to 0.00033 enhancing predictive performance. Figure 13 shows the bar chart comparing the performance of BiGRU-ROA, GA, MMDL, and FFNN in terms of Precision, Recall, F-Measure, and Accuracy.

Table 6. Proposed system performance comparison with existing models

Model vs criteria	Precision			Recall			F-Measure			Accuracy			MAE			RMSE		
	A	B	C	A	B	C	A	B	C	A	B	C	A	B	C	A	B	C
Proposed system	98	98	97	97	97	98	97	97	98	98	97	98	0.03	0.03	0.03	0.17	0.17	0.16
MMDL	92	94	94	96	93	94	95	94	95	95	92	93	0.06	0.05	0.05	0.25	0.25	0.25
GA	96	95	95	97	95	95	96	95	95	96	94	95	0.05	0.07	0.07	0.22	0.24	0.24
FFNN	85	85	85	88	86	87	87	88	89	86	87	86	0.14	0.15	0.15	0.36	0.36	0.37

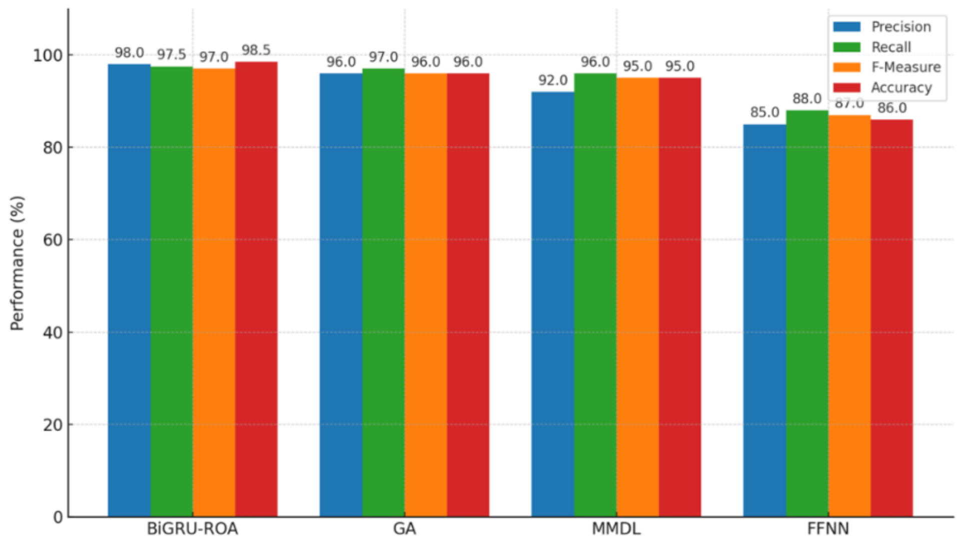

Figure 13. Performance comparison of model

As proposed model came out with comparatively better results than existing approaches, the objectives of research are attended. To further validate the robustness and learning efficiency of the proposed BiGRU-ROA model, we analyzed its training loss behavior over multiple epochs as shown in Figure 14. The training loss curve provides insights into how quickly and effectively the model learns from the input data. Faster convergence with minimal fluctuations indicates stable learning and effective weight optimization. The ROA plays a key role here by dynamically adjusting weights to minimize error during each iteration.

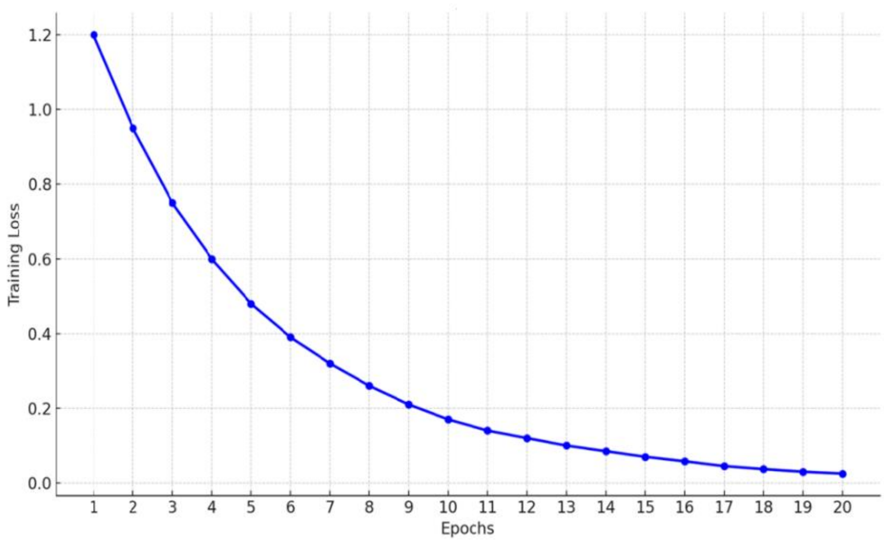

Figure 14. Training loss vs epochs BiGRU-ROA

As shown in Figure 14, the training loss for BiGRU-ROA decreases rapidly within the first 10 epochs and continues to improve steadily, reaching a minimal loss value of approximately 0.025 by the 20th epoch. This demonstrates the model's capability to converge faster compared to traditional deep learning methods. The smooth and consistent decline in the loss curve is a clear indication of ROA's contribution to weight optimization and error reduction, leading to improved generalization and predictive accuracy of the recommendation system.

4. CONCLUSION

The findings are compared to those produced using GA, MMDL, and FFNN and examined for characteristics like precision, recall, accuracy, F-measure, MAE, and RMSE. It is discovered that BGRU with ROA has better results than the others, with 97% accuracy, 97.5% F-measure, 97% precision, and 98% recall. The lowest values among all the remaining methods for comparison are MAE 0.03, RMSE 0.17. Therefore, it can be said that BiGRU with ROA performs better when it comes to movie selection. The novel approach in the future, the newest machine learning algorithms for movie recommendation may employ ROA. Additionally, BiRGU can be combined with any recent optimization algorithm for improved recommendation.

REFERENCES

[1] M. Kartiwi, T. S. Gunawan, T. Arundina, and M. A. Omar, "Sukuk rating prediction using voting ensemble strategy," *International Journal of Electrical and Computer Engineering*, vol. 8, no. 1, pp. 299–303, Feb. 2018, doi: 10.11591/ijece.v8i1.pp299-303.

[2] Y. M. Su, Y. Zhang, J. Y. Yan, S. YuMin, Z. Yuan, and Y. JinYao, "Neural network based movie rating prediction," in *Proceedings of the 2018 International Conference on Big Data and Computing*, Apr. 2018, pp. 33–37, doi: 10.1145/3220199.3220204.

[3] M. Ibrahim, I. S. Bajwa, R. Ul-Amin, and B. Kasi, "A neural network-inspired approach for improved and true movie recommendations," *Computational Intelligence and Neuroscience*, vol. 2019, pp. 1–19, Aug. 2019, doi: 10.1155/2019/4589060.

[4] P. P. Rokade and D. Aruna Kumari, "Business recommendation based on collaborative filtering and feature engineering - A proposed approach," *International Journal of Electrical and Computer Engineering*, vol. 9, no. 4, pp. 2614–2619, Aug. 2019, doi: 10.11591/ijece.v9i4.pp2614-2619.

[5] W. R. Bristi, Z. Zaman, and N. Sultana, "Predicting IMDb rating of movies by machine learning techniques," *2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, Kanpur, India, 2019, pp. 1-5, doi: 10.1109/ICCCNT45670.2019.8944604.

[6] E. Ozanich, P. Gerstoft, and H. Niu, "A feedforward neural network for direction-of-arrival estimation," *The Journal of the Acoustical Society of America*, vol. 147, no. 3, pp. 2035–2048, Mar. 2020, doi: 10.1121/10.0000944.

[7] I. Pratheek and J. Paulose, "Prediction of answer keywords using Char-RNN," *International Journal of Electrical and Computer Engineering*, vol. 9, no. 3, pp. 2164–2176, Jun. 2019, doi: 10.11591/ijece.v9i3.pp2164-2176.

[8] B. Ahirwadkar and S. N. Deshmukh, "Deep neural networks for recommender systems," *International Journal of Innovative Technology and Exploring Engineering*, vol. 8, no. 12, pp. 4838–4842, Oct. 2019, doi: 10.35940/ijitee.L3706.1081219.

[9] X. Feng and X. Liu, "Sentiment classification of reviews based on BiGRU neural network and fine-grained attention," *Journal of Physics: Conference Series*, vol. 1229, no. 1, p. 12064, May 2019, doi: 10.1088/1742-6596/1229/1/012064.

[10] B. Alhijawi and Y. Kilani, "A collaborative filtering recommender system using genetic algorithm," *Information Processing and Management*, vol. 57, no. 6, p. 102310, Nov. 2020, doi: 10.1016/j.ipm.2020.102310.

[11] Z. Mhowwala, A. R. Sulthana, and S. D. Shetty, "Movie rating prediction using ensemble learning algorithms," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 8, pp. 383–388, 2020, doi: 10.14569/IJACSA.2020.0110849.

[12] M. Zulqarnain, R. Ghazali, Y. M. M. Hassim, and M. Rehan, "Text classification based on gated recurrent unit combines with support vector machine," *International Journal of Electrical and Computer Engineering*, vol. 10, no. 4, pp. 3734–3742, Aug. 2020, doi: 10.11591/ijece.v10i4.pp3734-3742.

[13] P. P. Rokade, P. V. R. D. P. Rao, and A. K. Devarakonda, "Forecasting movie rating using k-nearest neighbor based collaborative filtering," *International Journal of Electrical and Computer Engineering*, vol. 12, no. 6, pp. 6506–6512, Dec. 2022, doi: 10.11591/ijece.v12i6.pp6506-6512.

[14] S. B. Rajendra, S. Kuzhalvaimozhi, and V. N. Prasad, "Denigration analysis of Twitter data using cyclic learning rate based long short-term memory," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 15, no. 1, pp. 700–710, Feb. 2025, doi: 10.11591/ijece.v15i1.pp700-710.

[15] H. Jia, X. Peng, and C. Lang, "Remora optimization algorithm," *Expert Systems with Applications*, vol. 185, p. 115665, Dec. 2021, doi: 10.1016/j.eswa.2021.115665.

[16] I. Sadgali, N. Sael, and F. Benabbou, "Bidirectional gated recurrent unit for improving classification in credit card fraud detection," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 21, no. 3, pp. 1704–1712, Mar. 2021, doi: 10.11591/ijeecs.v21.i3.pp1704-1712.

[17] X. Zhao, H. Kang, T. Feng, C. Meng, and Z. Nie, "A hybrid model based on LFM and BIGRU toward research paper recommendation," *IEEE Access*, vol. 8, pp. 188628–188640, 2020, doi: 10.1109/ACCESS.2020.3031281.

[18] S. Babeetha, B. Muruganantham, S. Ganesh Kumar, and A. Murugan, "An enhanced kernel weighted collaborative recommended system to alleviate sparsity," *International Journal of Electrical and Computer Engineering*, vol. 10, no. 1, pp. 447–454, Feb. 2020, doi: 10.11591/ijece.v10i1.pp447-454.

[19] M. F. Aljunid and M. D. Huchaiah, "Multi-model deep learning approach for collaborative filtering recommendation system," *CAAI Transactions on Intelligence Technology*, vol. 5, no. 4, pp. 276–282, Nov. 2020, doi: 10.1049/trit.2020.0031.

[20] A. Benlahbib and E. H. Nfaoui, "MTVRep: A movie and TV show reputation system based on fine-grained sentiment and semantic analysis," *International Journal of Electrical and Computer Engineering*, vol. 11, no. 2, pp. 1613–1626, Apr. 2021, doi: 10.11591/ijece.v11i2.pp1613-1626.

[21] R. Liang, X. Chang, P. Jia, and C. Xu, "Mine gas concentration forecasting model based on an optimized BiGRU network," *ACS Omega*, vol. 5, no. 44, pp. 28579–28586, Oct. 2020, doi: 10.1021/acsomega.0c03417.

[22] D. Chen, X. Yan, X. Liu, S. Li, L. Wang, and X. Tian, “A multiscale-grid-based stacked bidirectional GRU neural network model for predicting traffic speeds of urban expressways,” *IEEE Access*, vol. 9, pp. 1321–1337, 2021, doi: 10.1109/ACCESS.2020.3034551.

[23] D. K. Behera, M. Das, S. Swetanisha, and P. K. Sethy, “Hybrid model for movie recommendation system using content K-nearest neighbors and restricted Boltzmann machine,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 23, no. 1, pp. 445–452, Jul. 2021, doi: 10.11591/ijeecs.v23.i1.pp445-452.

[24] W. Guanchen, M. Kim, and H. Jung, “Personal customized recommendation system reflecting purchase criteria and product reviews sentiment analysis,” *International Journal of Electrical and Computer Engineering*, vol. 11, no. 3, pp. 2399–2406, Jun. 2021, doi: 10.11591/ijece.v11i3.pp2399-2406.

[25] S. Al-Otaibi *et al.*, “Cosine similarity-based algorithm for social networking recommendation,” *International Journal of Electrical and Computer Engineering*, vol. 12, no. 2, pp. 1881–1892, Apr. 2022, doi: 10.11591/ijece.v12i2.pp1881-1892.

[26] S. Krishnan, P. Magalingam, and R. Ibrahim, “Hybrid deep learning model using recurrent neural network and gated recurrent unit for heart disease prediction,” *International Journal of Electrical and Computer Engineering*, vol. 11, no. 6, pp. 5467–5476, Dec. 2021, doi: 10.11591/ijece.v11i6.pp5467-5476.

BIOGRAPHIES OF AUTHORS

Prakash Pandharinath Rokade has received his Ph.D. in computer science and engineering from Koneru Lakshmaiah Education Foundation, formerly K. L. University, Vaddeswaram, Andhra Pradesh, India. Currently, he is working as an associate professor in the Department of Information Technology, S.N.D. College of Engineering and Research Center, Yeola, Maharashtra, India. He has published more than 30 papers in several international journals. He is an active reviewer of two international journals. His research interests include sentiment analysis, opinion mining, machine learning, deep learning, and discrete mathematics. He can be contacted at email: prakashrokade2005@gmail.com.

Swati Babasaheb Bhonde holds a Ph.D. in computer engineering from Savitribai Phule Pune University, Pune, and currently serves as an assistant professor in the Department of Computer Engineering at Amrutvahini College of Engineering, Sangamner, Maharashtra, India. She has published over 25 research papers in reputed journals and is actively involved as a reviewer for prestigious journals from Springer and Elsevier. Her research interests span across bioinformatics, machine learning, and the internet of things (IoT). She can be contacted at email: swati.bhonde@gmail.com.

Prashant Laxmanrao Paikrao is working as an assistant professor in the Electronics and Telecommunication Engineering Department, Government College of Engineering Amravati. He has attended an international workshop on Global ICT Standardization Forum for India (AICTE Delhi & CTIF International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-2, Issue-4, March, 2013 (Denmark) at Sinhgad Institute of Technology, Lonavala, Pune and a workshop on ECG analysis and interpretation conducted by Prof. P. W. Macfarlane, Glasgow, Scotland. He can be contacted at email: plpaikrao@gmail.com.

Umesh Pawar has received his Ph.D. from the School of Computer Science, KBC North Maharashtra University. Currently, he is working as an associate professor in the School of Computer Science and Engineering, Sandip University, Nashik, Maharashtra, India. He has published 20 research papers in reputed journals and is actively involved as a reviewer for prestigious journals from Elsevier. His research interests are the internet of things (IoT) and computer networks. He can be contacted at email: umesh.pawar@sandipuniversity.edu.in.