



Deep recurrent Gaussian Nesterovs recommendation using multi-agent in social networks

Vinita Tapaskar¹ · Mallikarjun M. Math²

Received: 12 January 2022 / Accepted: 18 March 2022 / Published online: 9 April 2022
© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2022

Abstract

Due to increasing volume of big data the high volume of information in Social Network put a stop to users from acquiring serviceable information intelligently so many recommendation systems have emerged. Multi-agent Deep Learning gains rapid attraction, and the latest accomplishments address problems with real-world complexity. With big data precise recommendation has yet to be answered. In proposed work Deep Recurrent Gaussian Nesterov's Optimal Gradient (DR-GNOG) that combines deep learning with a multi-agent scenario for optimal and precise recommendation. The DR-GNOG is split into three layers, an input layer, two hidden layers and an output layer. The tweets obtained from the users are provided to the input layer by the Tweet Accumulator Agent. Then, in the first hidden layer, Tweet Classifier Agent performs optimized and relevant tweet classification by means of Gaussian Nesterov's Optimal Gradient model. In the second layer, a Deep Recurrent Predictive Recommendation model is designed to concentrate on the vanishing gradient issue arising due to updated tweets obtained from same user at different time instance. Finally, with the aid of hyperbolic activation function in the output layer, building block of the predictive recommendation is obtained. In the experimental study the proposed method is found better than existing GANCF and Bootstrapping method 13–21% in case of recommendation accuracy, 22–32% better in recommendation time and 15–22% better in recall rate.

Keywords Social network · Big data · Gaussian Nesterov's · Gradient · Deep recurrent · Predictive · Recommendation

1 Introduction

In the recent few years due to the explosion of information, obtaining useful information to the users are said to be both time consuming and laborious. Owing to this, both the academic and industrial sectors indemnify huge amount of awareness to this issue. To persuade this essential, recommender systems have made an appearance as an efficient instrument to bestow relevant suggestion for the users about what type of the products that they may be likely interested into. To address the issue of information burden, a fine

grained recommendation algorithm is highly predominant. The recommendation system via collaborative filtering here instruments services of the users' in a personalized manner via past behavior, textual information, for modeling. However, data sparsity persist crucial issues for collaborative filtering.

The neural collaborative filtering recommendation method integrated user and item secondary information called, GRU-Attention Neural Collaborative Filtering (GANCF) was proposed in Wan et al. (2021). In this method, information pertaining to user-item, user assistance and item text assistance were utilized for feature mining. Also, Stacked Denoising Auto Encoder was utilized for extracting user features whereas Gated Recurrent Unit with secondary information was utilized for extracting items' latent vectors.

Here, attention mechanism was utilized in learning key information and the latent vectors acquired through deep learning was utilized in multiple-layer non-linear networks to acquire deeper feature representations for efficiently predicting user preferences. With this deep learning-based filtering method (Angelov and Soares 2020),

✉ Vinita Tapaskar
vinita.tapaskar@gmail.com

Mallikarjun M. Math
mmath@git.edu

¹ Visvesvaraya Technological University, Research Center, Jnana Sangama, Machhe, Belagavi 590018, Karnataka, India

² Department of Computer Science and Engineering, KLS, Gogte Institute of Technology, Udyambag, Belagavi 590008, Karnataka, India

both recall and root mean square error were enhanced and hence improving the recommendation performance of the model to a greater extent.

A fine-grained social network recommender system using Bootstrapping was proposed in (Aivazoglou et al. 2020; Kemmerer 2003). First a fine grained categorization based on the like categories of interest in the users' and her friends' profiles are obtained. Followed by which similarity calculation and tweaking of score to accurately obtain the content to match the user's interests was presented. Followed by which finally top recommendations were made. As a result, accuracy of recommendation made was said to be improved with minimum runtime.

To précis, main contributions are as follows:

1. This work offers a Deep Recurrent Gaussian Nesterov's Optimal Gradient (DR-GNOG) method that recommends accurate tweets in a timely manner by means of multi-agent system for recommendation.
2. Based on DR-GNOG, this work delivers a new classification algorithm, Nesterov Accelerated Gradient Optimized Classification algorithm for obtaining relevant tweets with optimized classified results.
3. Deep Recurrent Predictive Recommendation algorithm is used to capture accurate tweets by smoothening or removing irrelevant tweets from the forget gate of internal state in DR-GNOG for recommendation.

The remaining part of this paper is structured as follows. Section 2 introduces preliminary related material into two subsections one for the multi-agent systems and another for the recommendation systems. In Sect. 3, this paper present the details of the proposed, Deep Recurrent Gaussian Nesterov's Optimal Gradient (DR-GNOG) demonstrate it on sample network, and examine its time and accuracy measure. In Sect. 4 an extensive experimentation of DR-GNOG is done and compared with other recent methods using Sentiment140 dataset. Followed by a discussion in Sect. 5 with respect to three parameters, recommendation accuracy, recommendation time and recall rate. Section 6 offers concluding remarks.

2 Related works

The acceptance of social networks has intensified over the years, like Facebook, Google +, Twitter and Micro Blog. These networks are utilized to design the human friendship by means of a graph representation wherein the members, with the assistance of nodes and edge corresponds to a relationship. In further work the use of multi agent in recommendation systems is explored.

2.1 Multi-agent systems

In (Azaouzi and Romdhane 2018), Social Action-based Influence Maximization model was designed in social networks with the objective of measuring the impact power of each individual. With the application of influence maximization minimal set of influence nodes was utilized to spread maximum information. However, memory involved in analyzing influence power of each individual was not focused. To address this issue, a multi-agent architecture was utilized in Bakliwal et al. (2018) to shed light into the memory complexity aspect. Yet another resource-constrained multi-agent system for distributed control to minimize the error involved was focused in Mastrangelo et al. (2019).

A multi-agent system necessitates multiple entities or agents who take appropriate decisions in an autonomous manner within a shared environment. Each agent with a specific purpose and objective accomplish a goal. According to the tasks of the multi-agent, they operate on an independent manner.

An overview of recent developments made in multi-agent deep reinforcement learning was investigated in Gronauer and Diepold (2021). In (Wu and Hu 2019), a bipartite consensus problem was addressed involving high-order multi-agent systems. In this work, the interaction network in correspondence with the cooperative-competitive multi-agent system was modeled by means of a signed digraph. Also on the basis of the relative neighboring agent output information, a novel distributed controller was proposed with the purpose of achieving bipartite consensus. However, optimized results were not arrived at. To address on this aspect, a multi-agent optimization with distributed information was explained in Falsone et al. (2020).

In networked systems, one of the field of work received great attention from the research community is the distributed coordination problem owing to its comprehensive relevancy in real work applications involving, social networks, drone surveillance monitoring, wireless resource management and so on (Angelov 2019; Kishore et al. 2016).

A higher-order cluster agreement issue for network involving a continuous monitoring system for any given directed graph was proposed in (Develer and Akar 2020) using multi-agent system. However, the nature of dynamics plays a major role in time-varying communication graph. In (Grammatico 2017), both dynamics and protocols were considered with proximity and convex local constraints towards minimum iterations for multi-agent in accomplishing the task (Table 1).

2.2 Recommendation systems

Recommender systems have been triumphantly merchandising with the issue of information overload. Despite the popularity received by the recommendation methods, most

Table 1 Comparison of existing multi-agent systems

Method	Contribution	Merits	Demerits
Social Action-based Influence Maximization model	Social Action-based Influence Maximization model was introduced to calculate the impact power of each individual	Spread information much faster	Memory involved in analyzing influence power of each individual was not considered
Multi-agent architecture	Multi-agent architecture was applied to reduce the memory complexity	Accuracy was improved	Time was not minimized
Resource-constrained multi-agent system	Resource-constrained multi-agent system was developed for distributed control	Error was reduced	Accuracy was not considered
Multi-agent deep reinforcement learning	An overview of recent developments made in multi-agent deep reinforcement learning was examined	Scalability was improved	System performance was not focused
High-order multi-agent systems	High-order multi-agent systems were handle the bipartite consensus issue	Achieving bipartite consensus issue	Optimized results were not arrived at
Multi-agent optimization	Multi-agent optimization with distributed information was explained	Multi-agent scenario problem was addressed	Time was higher
A higher-order cluster agreement issue for network	A higher-order cluster agreement issue for a network involving a continuous monitoring system was discussed	Accuracy was increased	The recall was not minimized
Proximal dynamics in multi-agent network games	Proximal dynamics in multi-agent network games were considered for multi-agent tasks	Global convergence was classified in multi-agent dynamics	The convergence result was poor

of them only suit to the hypothesis only involving explicit feedback or else only with the availability of prior rating. Also, certain methods only concentrate on user and item magnitudes and avoid any supplementary information like time and location.

In (Yao et al. Aug 2014), a graph-based generic recommendation method that models a Multi-Layer Context Graph (MLCG) with the aid of implicit feedback and then conducting ranking for context-aware recommendation was proposed, therefore improving accuracy. A review of text-based recommendations was investigated in depth in Kanwal et al. (2021).

In the few recent years there has seen numerous types of deep reinforcement learning techniques with the aid of cooperative multi-agent systems for handling different scenarios in social network. However, due to the absence of proposer theoretical insight, to be more specific it remains uncertain what the learning rate are employed in neural networks (Angelov and Filev 2004). In other words, enhancement of learning power was also not focused. In (Castellini et al. 2021), the learning power of numerous network models was empirically investigated.

In the last few years, context-aware recommender systems specifically designed towards the online social networks mastered discernible magnification. This has resulted

in the conduct of numerous researches in this area. These systems were in turn found to detect specific user requirements and proposed recommendations to the requested users. In (Suhaim and Berri 2021), a complete review of context-aware recommender systems for social networks was designed. A comprehensive literature review that precisely defined, the objective, its scope and finally materials and methods to conduct this research was presented in detail.

The interaction between the users in connected social networks not only disclose the social relationships in electronics commerce systems, but also oblique the social inclinations of target user for providing numerous recommendation services. But, the prevailing research has hardly ever inspected that influence of social communication on recommendation performance. Also, the recommender systems designed suffer from poor efficiency owing to the reason that already the data in social network are found to be overload.

In (Li et al. 2019), a novel social recommendation method on the basis of the user interaction in complicated social networks was designed with the objective of providing a source for enhancing the effectiveness of the recommender systems. Yet another deep Q-learning framework to handle robust strategy with high dimensional environments was proposed in Tampuu et al. 2017. Yet another control and coordination policy was designed in Luo et al. 2016 as a means of

providing best recommendation (Table 2). A methodology for influence maximization using influence propagation was proposed in Prem Sankar et al. 2016.

Motivated by the above research standing and existing issues, in this paper the work pays attention to the accuracy in which the recommendation is made and also on the timeliness with which the recommendation is made by the recommender agent in multi-agent system. Specifically, the gradient descent and momentum of each user of social users are exploited to classify tweet and obtain relevant and optimal tweets and then the activation function is calculated to obtain the accurate recommended tweets. Based on these various analyses, this paper design a new method for social recommendation in complicated social networks via multi-agent, aiming to enhance the performance of recommendations. Extension experiments based on Sentiment 140 dataset indicate that the proposed method outperformed the up-to-date methods in terms of accuracy, time and recall.

3 Methodology

In this section, a confidence-aware tweet recommendation method called, Deep Recurrent Gaussian Nesterov's Optimal Gradient (DR-GNOG) is proposed. The DR-GNOG method includes three different layers. They are the input layer, two

hidden layer and one output layer. Figure 1 shows the block diagram of DR-GNOG method.

As shown in the above figure, in the proposed work, three different agents called, Tweet Accumulator Agent 'TAA', Tweet Classifier Agent 'TCA' and Recommender Agent 'RA' are used. In the input layer the Tweet Accumulator Agent 'TAA', initially acquires the tweets 'T', from the respective users 'U' (i.e., from Big Data). Second, in the first hidden layer (i.e., hidden layer 1), Tweet Classifier Agent 'TCA' classifies the acquired tweets 'T' for the corresponding users 'U' in an optimal manner by employing Gaussin Nesterov's Optimal Gradient model.

Next, in the second hidden layer (i.e., hidden layer 2) a relevance social media interest prediction process is performed by means of Deep Recurrent Predictive Recommendation model. Finally, the Recommender Agent 'RA' recommends the fine coarse grained tweets 'T' to the stipulated users 'U' with a deeper insight by means of Deep Recurrent Predictive Recommendation model.

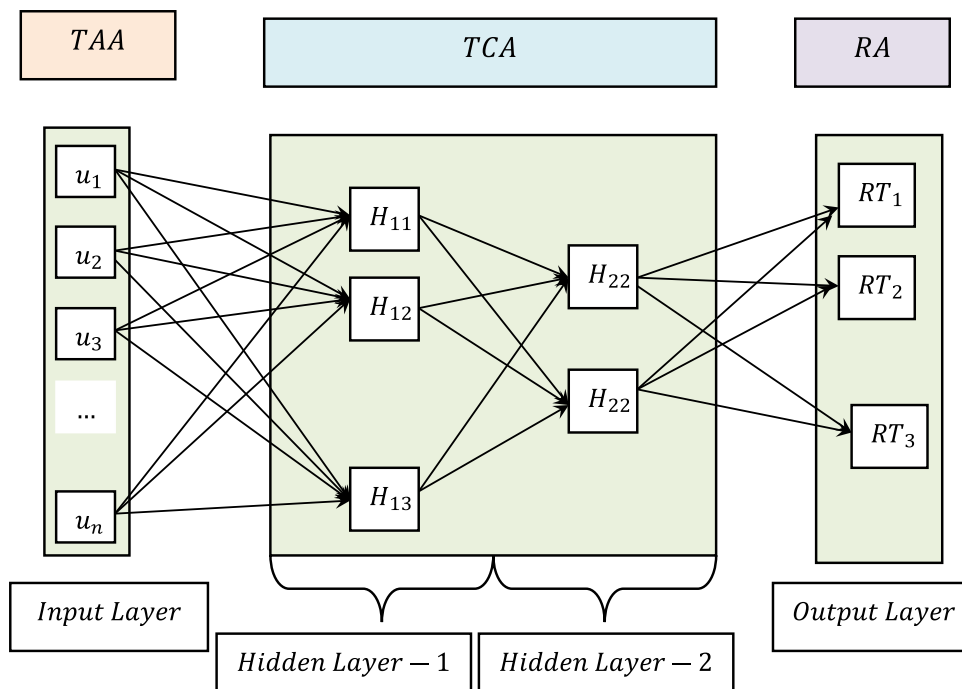
3.1 Gaussian Nesterov's optimal gradient model

To start with all the inputs, i.e., users 'U' and the corresponding tweets 'T' are obtained from the Sentiment 140 Big Data dataset (<https://www.kaggle.com/kazanova/sentiment140>).

Table 2 Comparison of existing recommendation systems

Method	Contribution	Merits	Demerits
Graph-based generic recommendation method	Graph-based generic recommendation method was designed in MLCG	Accuracy was higher	Large-scale data was not focused on ranking algorithm
Text-based recommendations	A review of text-based recommendations was discussed	The recall was enhanced	Learning power enhancement was not focused
Learning the power of numerous network models	The learning power of numerous network models was focused	Minimum communication requirements were used	Sequential multi-agent problems were not considered
Context-aware recommender systems	A review of context-aware recommender systems was designed for social networks	Research gaps, challenges were discovered	Scalability, the novelty was not determined
Novel social recommendation method	Novel social recommendation method based on user interaction was introduced to improve the effectiveness of the recommender systems	Recommendation accuracy was enhanced	Failed to focus on the dynamic learning of user behavior that changes over time
Deep Q-learning framework	Deep Q-learning framework was introduced to focus on the high dimensional environments	Performance was enhanced	Memory was not focused
Control and coordination policy	Control and coordination policy was designed to offer the best recommendation	Robustness and efficiency were obtained	Time was higher
Methodology for influence maximization	A methodology for influence maximization using influence propagation was proposed	The runtime was minimized	Failed to focus on the accuracy parameter

Fig. 1 Block diagram of DR-GNOG



$$X_t \rightarrow \sum_{i,j=1}^{m,n} U_i T_j. \tag{1}$$

From the above Eq. (1), ‘ X_t ’ forms the input acquired from the dataset and stored in the input layer at time instance ‘ t ’. Then, the user-tweet matrix is represented as given below.

$$X_t = \begin{bmatrix} u_1 t_1 & u_1 t_2 & u_1 t_3 & \dots & u_1 t_n \\ u_2 t_1 & u_2 t_2 & u_2 t_3 & \dots & u_2 t_n \\ \dots & \dots & \dots & \dots & \dots \\ u_m t_1 & u_m t_2 & u_m t_3 & \dots & u_m t_n \end{bmatrix}. \tag{2}$$

Next, with the above user-tweet matrix (2) acquired at different time instance the input is passed on to the next layer. In the first hidden layer or the hidden layer 1, optimized classification of tweets are made by the Tweet Classifier Agent ‘TCA’ by employing Gaussin Nesterov’s Optimal Gradient model. Let us assume that ‘ $U = u_1, u_2, u_3, \dots, u_n$ ’ corresponds to the user set in social network and ‘ $T = t_1, t_2, t_3, \dots, t_n$ ’ denotes the tweets made by the respective users pertaining to specific applications.

The ranks allocated by customers or users on corresponding tweets are denoted as a user-tweet ranking matrix ‘ $R \in R^{i*j}$ ’, where ‘ $R_{i,j}$ ’ represents the ranking of tweet ‘ j ’ allocated by user ‘ i ’ respectively. In proposed work, the ranking are frequently denoted as integers ‘1 – 10’ (i.e., the proximate value range differs according to the tweets involved in simulation, with 50,000 tweets considered for simulation the proximate value ranges between 1 and 10).

Also in a ranking social network of users and tweets made by them, each user ‘ i ’ has a set of proximate ‘ P_i ’ and ‘ $C_{i,j}$ ’ denotes the social confidence value of user ‘ i ’ allocates on user ‘ j ’ in the range ‘1 – 5’ respectively. If the resultant range value is Zero or ‘0’, it refers no confidence exists between the user ‘ i ’ and ‘ j ’, and on the contrary, it refers full confidence. Figure 2 given below shows the structure of Confidence-aware Tweet Rank Social Network model.

Figure 2 illustrates the structure of confidence-aware tweet rank social network model. By using this model, the social network includes the user, and tweet ranking matrix. The user denoted as ‘ u_1, u_2, \dots, u_5 ’, and ‘ t_1, t_2, \dots, t_5 ’ represents the tweet ranking matrix. As shown in the above figure, from the ranking matrix that exclusively a fragment of the user-tweet ranking matrix is utilized for recommendation and the other rankings are not studied. Hence, the confidence-aware recommendation task in proposed work is described as given a user ‘ i ’ and tweet ‘ j ’, the objective remains in predicting the ranking on tweet ‘ j ’ from user ‘ i ’ by utilizing the confidence-aware tweet rank social network model via user-tweet ranking matrix ‘ R ’ and the confidence association matrix ‘ C ’ respectively. Figure 3 given below shows the sample user-tweet ranking matrix.

As shown in the above figure, certain values are missing in the user-tweet ranking matrix. As the user tweet relationships are governed by hardly any essential features, and the execution of each feature will have a greater impact Matrix Factorization is first implemented to evaluate the missing or lost values in the user-tweet ranking matrix, therefore laying a deeper insight even in the presence of heavy voluminous of

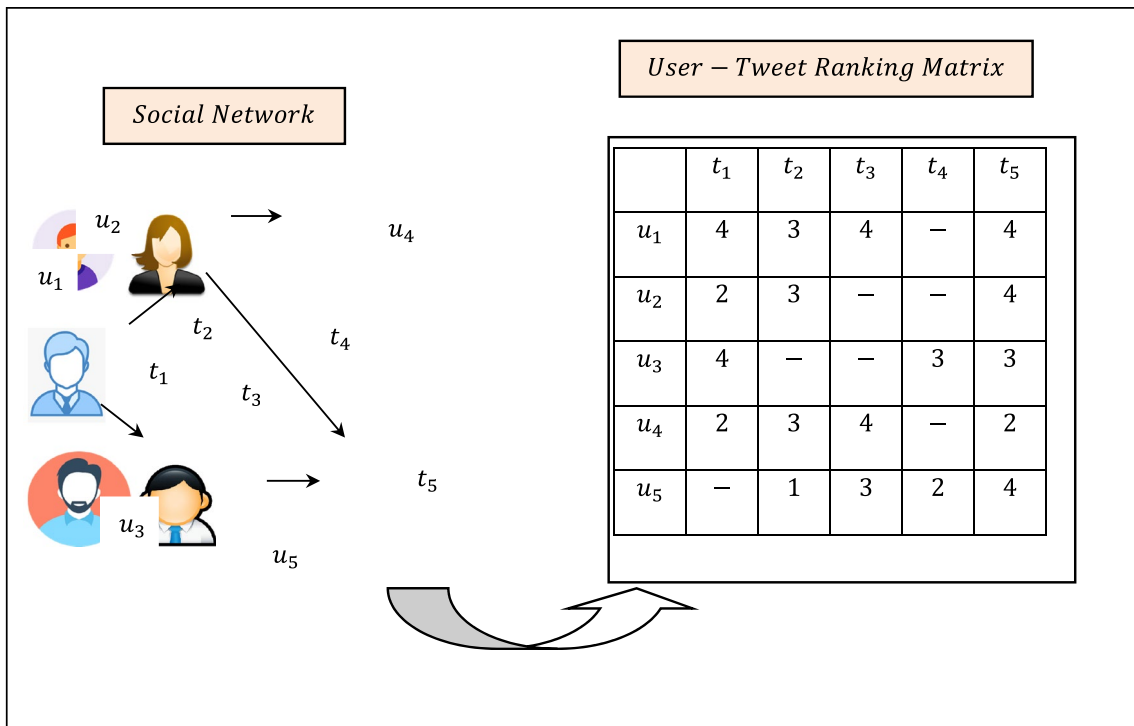
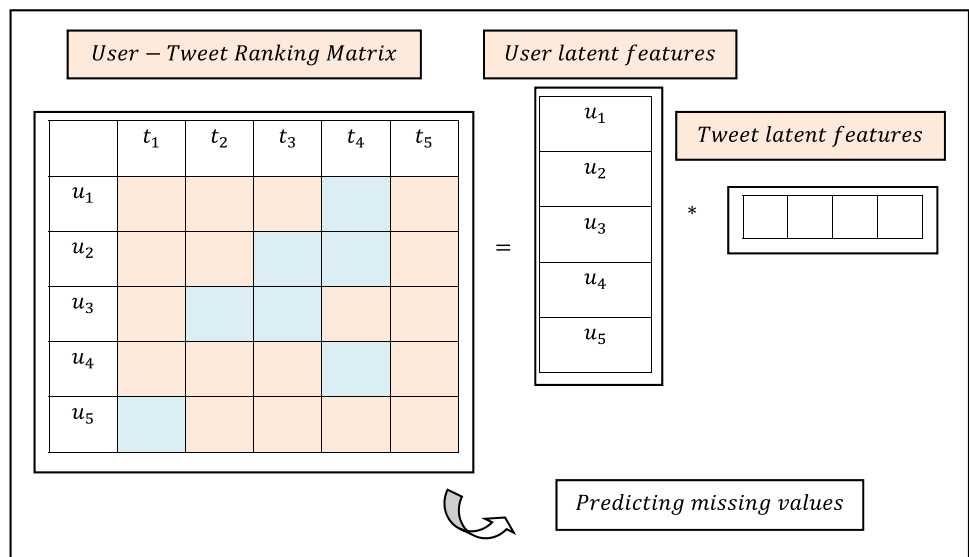


Fig. 2 Structure of Confidence-aware Tweet Rank Social Network model

Fig. 3 Sample user-tweet ranking matrix



data (i.e., improving the recall rate). In the proposed work, the contingency collinear model with Gaussian function is employed. The contingent classification over the perceived ranking is mathematically formulated as given below.

$$Res = Prob(R|X, Y, \sigma^2) = \prod_{i=1}^m \prod_{j=1}^n [\mathcal{N}(R_{ij}|X_i^T Y_j, \sigma^2) C_{ij}]. \tag{3}$$

From the above Eq. (3), ' $\mathcal{N}(a|\mu, \sigma^2)$ ' represents the Gaussian function and the criterion function ' C_{ij} ' is equal to '1' if user ' i ' has ranked ' j ' whereas the criterion function ' C_{ij} ' is equal to '0' if user ' i ' has not ranked with any value. Then perform the Nesterov Accelerated Gradient in ' X ' and ' Y ' to minimize the objective function given in (1). This Nesterov Accelerated Gradient is utilized in

proposed work for training neural networks to optimize the contingent classification function. The objective in work remains in optimizing the two different parameters ‘ θ ’ (i.e., users’ tweets in work while identifying missing values) by means of different iterations ‘ l ’ and moving a certain distance in the direction of the negative of the gradient (i.e., proximate to the user), the distance being related to the learning rate ‘ $\varepsilon > 0$ ’ (Sutskever et al. 2013) with momentum coefficient ‘ $\mu \in [0, 1]$ ’ (<https://www.kaggle.com/kazanova/sentiment140>). Then the update formulation for two different users ‘ X ’ and ‘ Y ’ with its corresponding gradient descent is given as in Eqs. (4) and (5).

$$X[\theta^l] = X_i^{(l+1)}[Res^l] - \varepsilon^l \nabla f(Res^l), \tag{4}$$

$$Y[\theta^l] = Y_i^{(l+1)}[Res^l] - \varepsilon^l \nabla f(Res^l). \tag{5}$$

Followed by which, the momentum of each user ‘ X ’ and ‘ Y ’ is mathematically formulated as in Eqs. (6) and (7).

$$Res^{l+1} = X[\theta^l] + \mu^l (X[\theta^l] - X[\theta^{l-1}]), \tag{6}$$

$$Res^{l+1} = Y[\theta^l] + \mu^l (Y[\theta^l] - Y[\theta^{l-1}]). \tag{7}$$

With the above momentum values relevant and optimized classification of tweets even in case of missing values are obtained, therefore contributing to better recall rate. The pseudo code representation of Nesterov Accelerated Gradient Optimized Classification is given below.

Input: Users ‘ $U = u_1, u_2, u_3, \dots, u_n$ ’
Output: Relevant and optimized tweet classification
Step 1: Initialize tweets ‘ $T = t_1, t_2, t_3, \dots, t_n$ ’, iterations ‘ l ’
Step 2: Initialize learning rate ‘ ε ’, momentum coefficient ‘ μ ’
Step 3: Initialize Tweet Accumulator Agent ‘ TAA ’, Tweet Classifier Agent ‘ TCA ’, Recommender Agent ‘ RA ’
Step 4: Begin
Step 5: For each Users ‘ U ’ with tweets ‘ T ’
//Input layer
Step 6: Obtain the inputs and store it in user-tweet matrix as in equation (2)
//Hidden layer 1
Step 7: Evaluate contingency collinear model with Gaussian function as in equation (3)
Step 8: Estimate gradient descent for two different users ‘ X ’ and ‘ Y ’ as in equation (4) and equation (5)
Step 9: Estimate momentum for two different users ‘ X ’ and ‘ Y ’ as in equation (6) and equation (7)
Step 10: Return optimal classified tweets (‘ TP, TN, TNe ’)
Step 11: End for
Step 12: End

As given in the above Nesterov Accelerated Gradient Optimized Classification algorithm, the objective remains in attaining relevant and optimized tweet classification with good recall rate. In the input layer for each user with tweets is obtained as input. The user-tweet ranking matrix is obtained according to the confidence value. Then, the input is transmitted into the first hidden layer. In that layer, a contingency collinear model with a Gaussian function is computed. Gradient descent for two dissimilar users is measured. Next, momentum for two different users is measured. Next, to address the missing values, the Nesterov Accelerated Gradient momentum function is applied that with the aid of gradient descent to classify the relevant tweets with high recall. Finally, higher the ranking more probability of the tweets being positive ‘*TP*’ and lower the ranking the tweets are considered as negative ‘*TN*’. Finally, the tweet centroid remains the neutral ‘*TNe*’. In this manner, relevant tweets with optimized classified results are said to be obtained.

3.2 Deep recurrent predictive recommendation model

In this section, with the obtained optimized classified relevant tweets, a Deep Recurrent Predictive Recommendation model is proposed to improve both the recommendation time and accuracy to a greater extent. Here, Deep Recurrent Predictive Recommendation model is proposed owing to the reason that evolving with time, the likes and dislikes of the users’ changes and also their tweets. Hence, a Deep Recurrent Neural Network based on Long Short Term Memory (LSTM) that holds the previous likes and dislikes of the users’ in the memory cell state. The cell state remembers likes/dislikes over arbitrary time instances and the three gates involved in LSTM (i.e., input gate, output gate and forget gate) tunes the flow of tweets into and out of the cell, therefore contributing to the recommendation accuracy.

The key to Deep Recurrent Neural Network based on LSTM is the cell state. The cell state in proposed work can be regarded as a social medial conveyor belt. It runs through the entire or different types of sentiments chain, involving both positive emoticons and negative emoticons, with only minor linear interactions. Each memory cell either provides a positive or negative tweet could post likes or dislikes to the cell state via cautiously outlined compositions called

gates. The gate comprises of a sigmoid layer and a point-wise (i.e., tweets changes made according to the location or place) multiplication operation.

The forget gate includes a forgetting coefficient by obtaining the input (i.e., optimal classified tweets) from the input layer ‘ X_t ’ and the previous hidden layer ‘ H_{t-1} ’ for the corresponding cell state ‘ C_{t-1} ’ respectively. The forget gate as given below assists the cell to smoothen (i.e., irrelevant) the tweet contents of the internal state.

$$F_t = \sigma(W_F * [H_{t-1}, X_t] + B_F). \quad (8)$$

Followed by which activation from the input layer ‘ X_t ’ is obtained and previous hidden layer ‘ H_{t-1} ’ using the tanh function to model the aggregated weighted input. This is formulated as given below.

$$G_t = \tanh(W_G * [H_t, X_t] + B_G). \quad (9)$$

The input gate on the other hand determines the tweets to be updated in corresponding cell state ‘ C_{t-1} ’ and its output accumulates the resultant input node value to get a new cell state towards the corresponding cell state ‘ C_t ’. The input gate as given below controls how much tweets is to enter the cell.

$$I_t = \sigma(W_I * [H_{t-1}, X_t] + B_I). \quad (10)$$

Next, the memory or the internal cell state ‘ C_t ’ is mathematically formulated as given below.

$$C_t = F_t * C_{t-1} + I_t * G_t. \quad (11)$$

Finally, the output cell state ‘ O_t ’ (i.e., the output layer) that forms the building block of the predictive recommendation is given below.

$$O_t = \sigma(W_O * [H_{t-1}, X_t] + B_O). \quad (12)$$

$$H_t(RT) = O_t * \tanh(C_t). \quad (13)$$

From the above Eqs. (12) and (13), the results of the hidden layer ‘ H_t ’ are obtained via internal state ‘ C_t ’ and resultant tweets from the output gate ‘ O_t ’ respectively. The pseudo code representation of Deep Recurrent Predictive Recommendation is given below.

Input: Users ' $U = u_1, u_2, u_3, \dots, u_n$ '
Output: Accurate recommended tweets ' RT '
Step 1: Initialize tweets ' $T = t_1, t_2, t_3, \dots, t_n$ ', iterations ' l '
Step 2: Initialize learning rate ' ε ', momentum coefficient ' μ '
Step 3: Initialize Tweet Accumulator Agent ' TAA ', Tweet Classifier Agent ' TCA ', Recommender Agent ' RA '
Step 4: Begin
Step 5: For each Users ' U ' with tweets ' T '
//Hidden layer 2
Step 6: Formulate the forget gate as in equation (8) [smoothen (i.e., irrelevant) tweet of internal state]
Step 7: Formulate aggregated weighted input as in equation (9)
Step 8: Obtain updated input gate as in equation (10) [controls irrelevant tweets entering the cell]
Step 9: Estimate cell state as in equation (11)
//Output layer
Step 10: Formulate output cell state as in equations (12) and (13) [building block of predictive recommendation]
Step 11: Return (recommended tweets ' RT ')
Step 12: End for
Step 13: End

As given in the above Deep Recurrent Predictive Recommendation algorithm, the objective remains in obtaining accurate recommended tweets with maximum accuracy and minimum time. With this objective a deep recurrent neural network forming the input gate, forget gate, memory and cell state are utilized. The forget gate is formulated to smoothen (i.e., irrelevant) tweets of the internal state. Aggregated weighted input is formulated. The input gate is updated to controls the irrelevant tweets entering the cell. Followed by, cell state is measured. Output cell state is formulated to the building block of predictive recommendation. Also with the previous hidden layer output obtained from Nesterov Accelerated Gradient Optimized Classification, the issue of vanishing gradient is addressed where a new layer

is created for each time instance of an input tweets processed by the network. With this, the accuracy and time with which the predictive recommendation is performed is said to be improved.

4 Experimental setup

In the arena of confidence-aware recommendation area, the openly available Sentiment140 dataset and coronavirus tweets NLP text classification dataset (Angelov and Filev 2004) are used. It includes 1,600,000 tweets which are derived through the twitter api. The tweets have been ranked as '0 = negative', '4 = positive' and these ranks can be used

Table 3 Average recommendation accuracy [experimented with increased number of tweets]

Number of tweets	Recommendation accuracy (%)		
	DR-GNOG	GANCF	Bootstrapping
5000	97.1	95.3	85.3
10,000	94.35	83.15	82
15,000	92.15	80.25	79.15
20,000	90	78.15	76
25,000	88.15	76.25	73.15
30,000	86.25	75	70
35,000	84	73.15	69.35
40,000	82.15	72	66
45,000	80.35	71	63
50,000	80	70	61

Table 4 Average recommendation accuracy [experimented with increased number of tweets] using coronavirus tweets NLP text classification dataset

Number of tweets	Recommendation accuracy (%)		
	DR-GNOG	GANCF	Bootstrapping
4000	98	96.24	88.5
8000	95.55	86.73	84
12,000	94.23	83.25	80.45
16,000	92	80.78	78
20,000	90	79.44	74.55
24,000	87	77	73
28,000	86.44	76.59	72.65
32,000	85.45	75	70
36,000	83	74	68
40,000	82	72	67

to noticesentimentality of the users. It Sentiment data set includes the below given six fields.

- target: This field shows the polarity of the tweet (0 =negative, 2 =neutral, 4 =positive),
- ids: The unique number given to the tweet ($200,087$),
- date: the date when the tweet has been done (*Sat May 16 23:58:44 UTC 2008*),
- flag: It is flag to induct query tweet or not(NO_QUERY)
- user: the username who had made the tweet (*john*) and
- text: the text which user has typed in the tweet (*This is cool*).

The proposed work use three metrics, the recommendation accuracy, recommendation time and recall rate, precision to quantify the performance of proposed projected method, Deep Recurrent Gaussian Nesterov's Optimal Gradient (DR-GNOG) comparing with other up-to-date

recommendation methods, GRU-Attention Neural Collaborative Filtering (GANCF) and Bootstrapping. The methods have been implemented in JAVA program language using the Cloud infrastructure.

5 Discussion

In this section comparison is done on proposed method DR-GNOG with existing methodology GANCF and Bootstrapping. The result have been compared with 3 optimization parameters those are recommendation accuracy, recommendation time, and recall rate.

5.1 Performance analysis of recommendation accuracy

Recommendation accuracy evaluates the accuracy rate at which the recommendation method has been made. In other words, recommendation accuracy ' R_{acc} ' refers to the percentage ratio of tweets correctly recommended by the respective recommender agent ' T_{CR} ' to the tweets utilized for simulation purpose ' T_i '. It is measured in terms of percentage (%). The recommendation accuracy is mathematically formulated as given below.

$$R_{acc} = \sum_{i=1}^n \frac{T_{CR}}{T_i} * 100. \quad (14)$$

With the resultant values obtained from (14), the average performance of recommendation accuracy for three different methods, DR-GNOG, GANCF and Bootstrapping is presented in Tables 3 and 4.

Figures 4 and 5 depicts the results of the recommendation accuracy of three different methods, DR-GNOG, GANCF and Bootstrapping using two different data set. Overall, the projected method has upper scores at both 5000 and 50,000 numbers of tweets, and 4000 to 40,000 number of tweets which shows that projected method has more better performance in recommendation accuracy than the other methods. It suggests that the proposed method by means of recommender agent with the assistance of tweet accumulator and tweet classifier agent could identify and bestow more relevant tweets for users. Furthermore, it can be seen that there is a difference in recommendation performance between 5000 and 10,000 numbers of tweets using sentiment140 dataset. More specifically, the performance of the proposed method at 5000 tweets reaches 97.1%, 95.3% and 85.3%, respectively. In proposed method at 10,000 tweets shows a better performance, reaching 94.35%, 83.15% and 82%. This difference indicates that preferences in users' profiles and the confidence measure were continuously changing and, as a result, had different weights at 5000 and 10,000 numbers

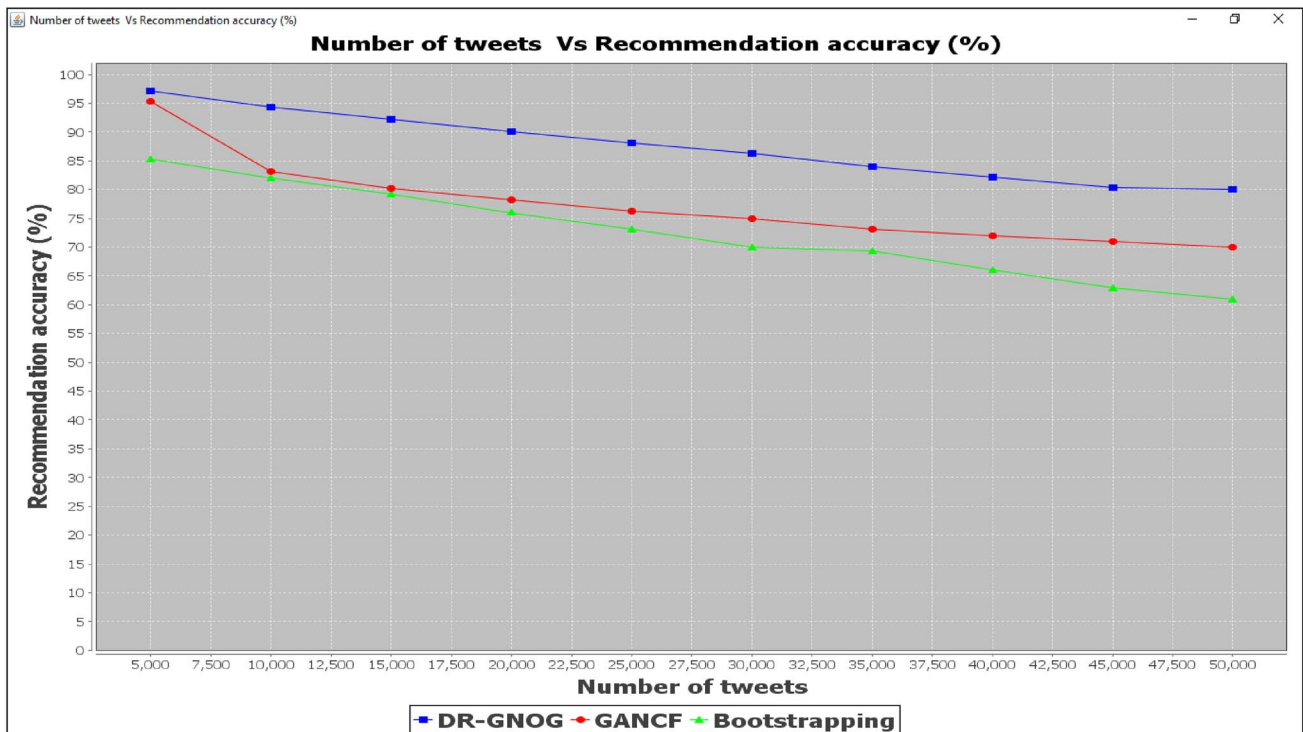


Fig. 4 Recommendation accuracy of three recommendation method [changes made] using sentiment 140 dataset

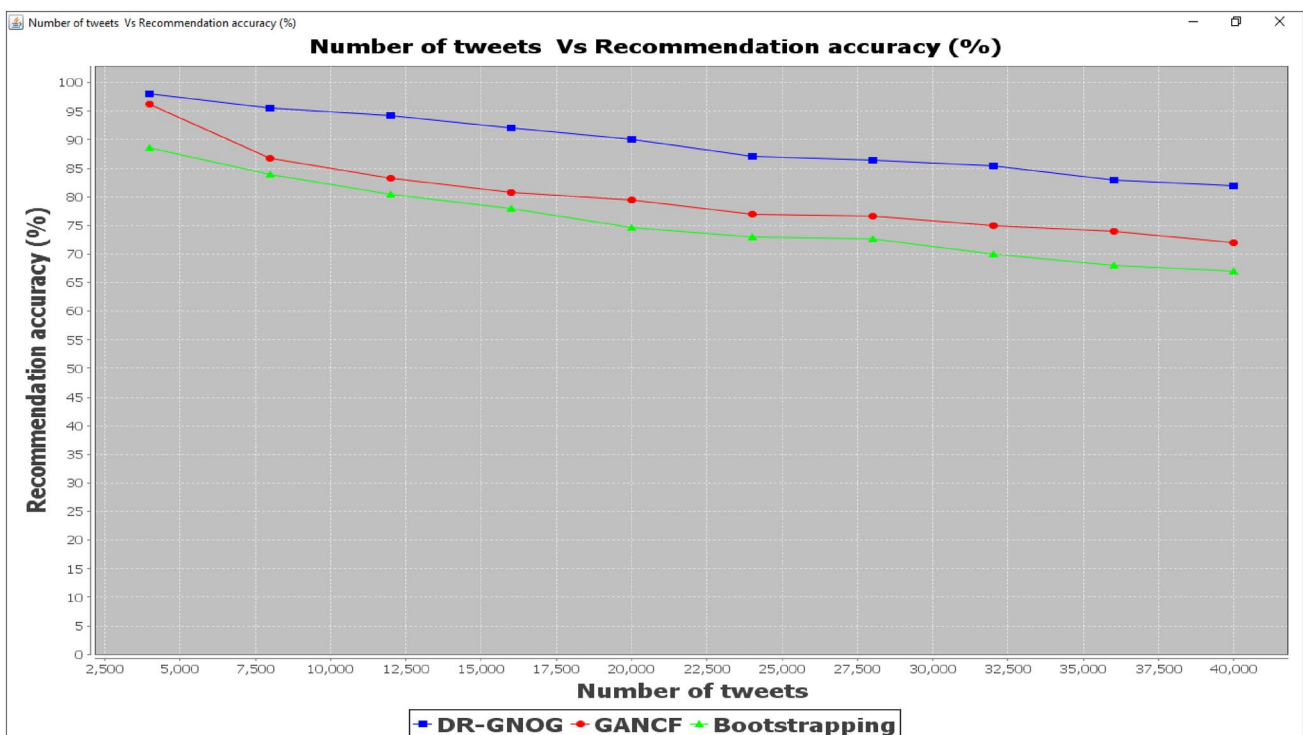


Fig. 5 Recommendation accuracy of three recommendation method [changes made] using coronavirus tweets NLP text classification dataset

Table 5 Average recommendation time [experimented with increased number of tweets] using sentiment 140 dataset

Number of tweets	Recommendation time (ms)		
	DR-GNOG	GANCF	Bootstrapping
5000	4175	5775	6625
10,000	6325	8135	9425
15,000	7155	9035	10,555
20,000	8235	10,435	11,325
25,000	9545	11,035	12,455
30,000	10,255	13,455	15,315
35,000	11,355	14,255	17,145
40,000	12,485	15,315	19,325
45,000	13,525	17,855	20,135
50,000	14,155	19,325	21,455

Table 6 Average recommendation time [experimented with increased number of tweets] using coronavirus tweets NLP text classification dataset

Number of tweets	Recommendation time (ms)		
	DR-GNOG	GANCF	Bootstrapping
4000	4585	5925	6955
8000	6555	8475	9745
12,000	7495	9265	12,765
16,000	8525	12,345	13,555
20,000	9975	13,435	15,785
24,000	12,455	15,655	17,865
28,000	14,865	17,865	19,325
32,000	16,455	18,245	21,785
36,000	18,275	19,355	22,455
40,000	20,345	21,655	23,565

of tweets respectively. These changes in time points have been captured by proposed method and in a similar manner for 50,000 tweets. The simulation result showed improvement using DR-GNOG upon comparison with GANCF and Bootstrapping. Owing to the reason that Deep Recurrent Predictive Recommendation algorithm has been applied for obtaining accurate recommended tweets where gate utilized in proposed work employed sigmoid layer and a point-wise (i.e., tweets changes made according to the location or place) multiplication operation. With this changes in the user preferences moderately affect the recommendation performance, in accuracy, was found to be comparatively better using DR-GNOG than 13% compared to GANCF and 21% compared to Bootstrapping respectively. For each method, ten accuracy results are observed with different inputs. By applying the sentiment140 dataset, the average of ten comparison results indicates that the proposed DR-GNOG improves the accuracy by 13% compared to GANCF and 21% compared

to Bootstrapping respectively. In coronavirus tweets NLP text classification dataset, accuracy is enhanced by 12% as compared to GANCF and 18% compared to Bootstrapping respectively.

5.2 Performance analysis of recommendation time

Recommendation time refers to the time consumed in making accurate recommendations by the recommender agent as requested by the users. In other words, recommendation time ' R_{time} ' refers to the time involved in classifying optimal and relevant tweets ' $Time[RecT_i]$ ' for the corresponding number of tweets involved in simulation ' T_i '. The recommendation time is mathematically formulated as given below.

$$R_{time} = T_i * Time[RecT_i]. \quad (15)$$

With the resultant values obtained from (15), the average performance of recommendation time for three different methods, DR-GNOG, GANCF and Bootstrapping is presented in Tables 5 and 6.

Additionally, the average performance of each measurement approach in terms of the recommendation time metric can be viewed in Figs. 6 and 7. The results depict that the scores for the projected method are lower than other methods in terms of recommendation time, indicating that the proposed method has a better average performance. The results hence approve that better performance in terms of recommendation time can improve the overall performance. In addition, the average performance between 5000 numbers of tweets and 50,000 numbers of tweets for sentiment140 dataset displays the differences in terms of recommendation time. This implies that the recommendation execution is influenced by time differently, that in turn records the significance of users' interest changes. In sentiment140 dataset. The recommendation time hence was found to be comparatively better using DR-GNOG by 22% compared to GANCF and 32% compared to Bootstrapping respectively. In coronavirus tweets NLP text classification dataset, the recommendation time is reduced using DR-GNOG by 18% compared to GANCF and 29% compared to Bootstrapping respectively. These uncovering recommend that encapsulating users' interest alternates does not only make precision improvements but also improve the memento and the recommendation range. At this juncture, the projected method is useful for effectively providing recommendations to the users who whichever comes to hand constantly change their preferences or maintain steady preferences about their tweets.

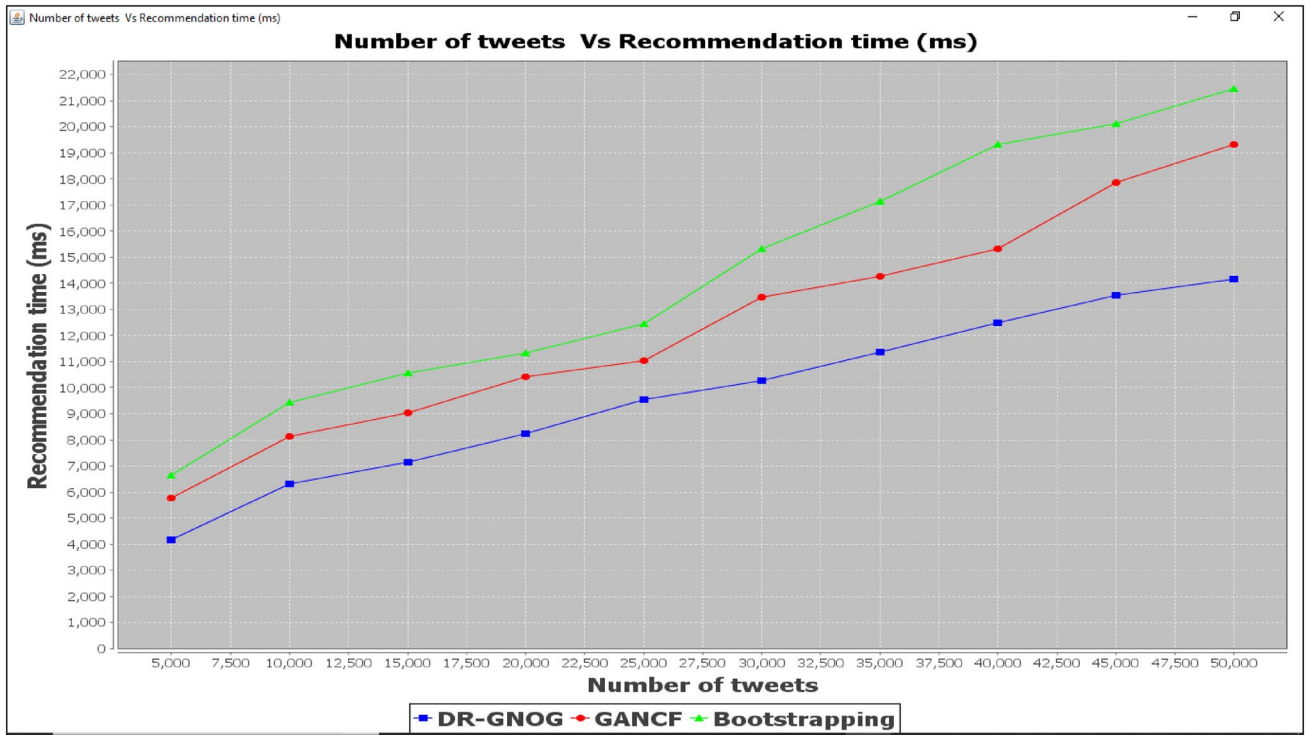


Fig. 6 Recommendation time of three recommendation method [changes made] using sentiment 140 dataset

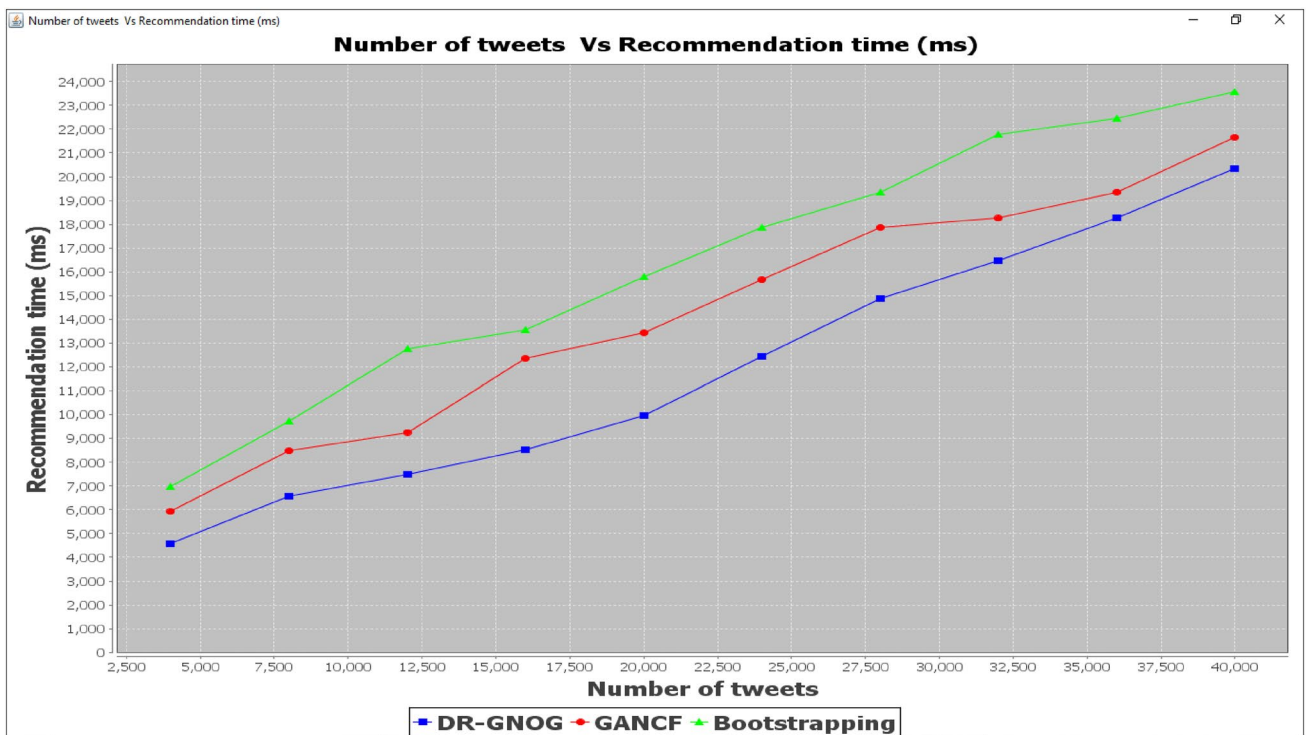


Fig. 7 Recommendation time of three recommendation method [changes made] using coronavirus tweets NLP text classification dataset

Table 7 Average recall rate [experimented with increased number of tweets] using sentiment140 dataset

Number of tweets	Recall (%)		
	DR-GNOG	GANCF	Bootstrapping
5000	0.85	0.83	0.82
10,000	0.84	0.82	0.81
15,000	0.83	0.82	0.8
20,000	0.82	0.81	0.79
25,000	0.82	0.8	0.79
30,000	0.81	0.8	0.77
35,000	0.81	0.79	0.75
40,000	0.81	0.79	0.75
45,000	0.8	0.79	0.74
50,000	0.8	0.77	0.73

Table 8 Average recall rate [experimented with increased number of tweets] using coronavirus tweets NLP text classification dataset

Number of tweets	Recall (%)		
	DR-GNOG	GANCF	Bootstrapping
4000	0.91	0.87	0.84
8000	0.9	0.86	0.83
12,000	0.89	0.85	0.81
16,000	0.88	0.84	0.8
20,000	0.88	0.84	0.8
24,000	0.87	0.83	0.79
28,000	0.87	0.83	0.79
32,000	0.84	0.81	0.77
36,000	0.83	0.8	0.76
40,000	0.82	0.79	0.75

5.3 Performance analysis of recall rate

Finally, the recall rate is estimated in proposed work. Higher recall will help correctly identify more true positives (i.e., correct recommendations made by the recommender agent) and fewer false negatives (i.e., false recommendations made by the recommender agent) from the data. This is mathematically formulated as given below.

$$R = \frac{TP}{TP + FN}. \quad (16)$$

From the above equation recall rate ' R ' is estimated based on the true positive rate ' TP ', and the false negative rate ' FN ' respectively. With the resultant values obtained from (16), the average performance of recall rate for three different methods, DR-GNOG, GANCF and Bootstrapping is presented in Tables 7 and 8.

Additionally, the results of the recommendation performance regarding the recall are exposed in Figs. 8 and 9. As presented in the figure, among the measure methods, the performance of this proposed method at the time points with sentiment140 dataset for 10,000 different tweets obtained at different time interval is located on the top, showing improved recommendation performance. More specifically, the proposed method has a higher proportion of relevant items for 5000 and 10,000 numbers of tweets using sentiment140 dataset. However, it is important to note the performance at 10,000 numbers of tweets is very close to the performance at 20,000 numbers of tweets. A probable verification is that the percentage of relevant tweets or true positive tweets obtained by the recommender agent via tweet accumulator and tweet classifier agent remains anchored at distinct time periods. This may have happened owing to the reason that the feasibility in the user preferences poses small influence on recall performance. Moreover, in sentiment140 dataset the results recommend that the performance at 5000 number of tweets is better than the performance at 10,000 numbers of tweets. Results also show that the performance in terms of the recall decreases moderately at both 10,000 and 15,000 tweets. This in turn infers that a optimistic association prevails between the ratio of relevant or true positive tweets and number of recommendations. The reasons behind the improvement using DR-GNOG was owing to the reason that the forget gate used in the work smoothen or removes the irrelevant tweet contents present in the internal state. With aid of the sentiment 140 dataset, the recall rate using DR-GNOG is improved by 2% compared to GANCF and 6% compared to Bootstrapping respectively. With aid of coronavirus tweets NLP text classification dataset, the DR-GNOG of recall rate is enhanced by 4% compared to GANCF and 9% compared to Bootstrapping respectively.

5.4 Performance analysis of precision

The fourth parameter of precision is estimated in this proposed work. The precision is measured as given below.

$$Precision = \frac{TP}{TP + FP}. \quad (17)$$

From the above Eq. (9), precision ' Pre ' is estimated based on the true positive rate (i.e., correct analysis made using tweets) ' TP ' and the false positive rate (i.e., falsely analysis the results using tweets) ' FP '. With the resultant r values obtained from (17), the average performance of precision for three different methods, DR-GNOG, GANCF, and Bootstrapping is presented in Tables 9, and 10.

Figures 10 and 11 shows the performance analysis of precision for three methods with respect to the number of tweets taken in the range from 4000 to 40,000. By comparing the proposed DR-GNOG technique with the existing methods

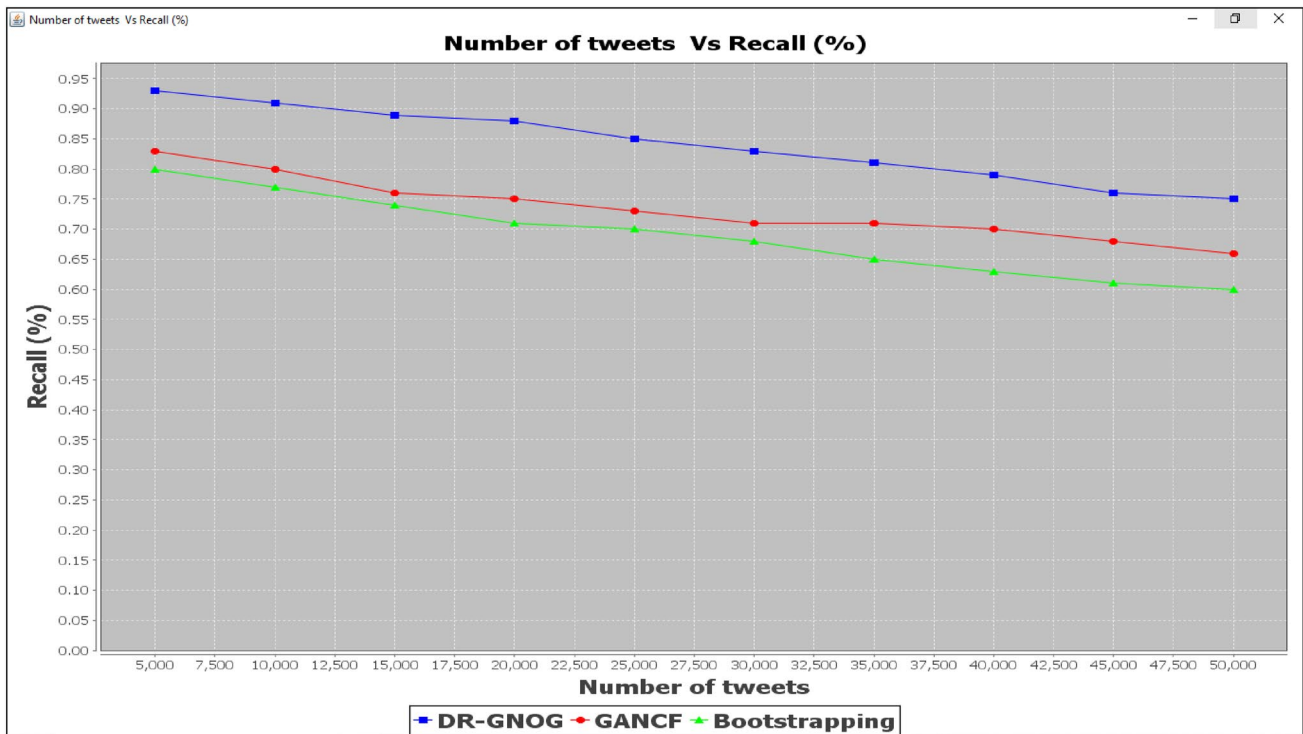


Fig. 8 Recall rate of three recommendation method [changes made] using sentiment140 dataset

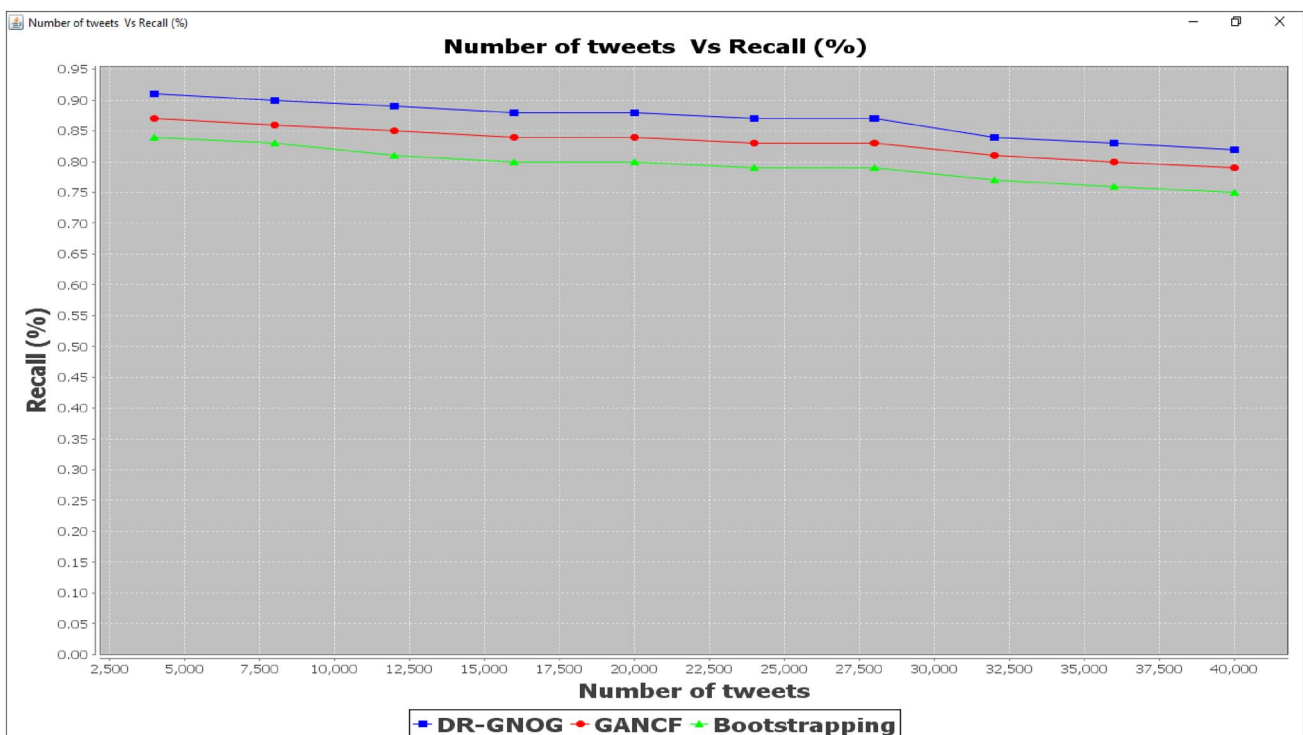


Fig. 9 Recall rate of three recommendation method [changes made] using coronavirus tweets NLP text classification dataset

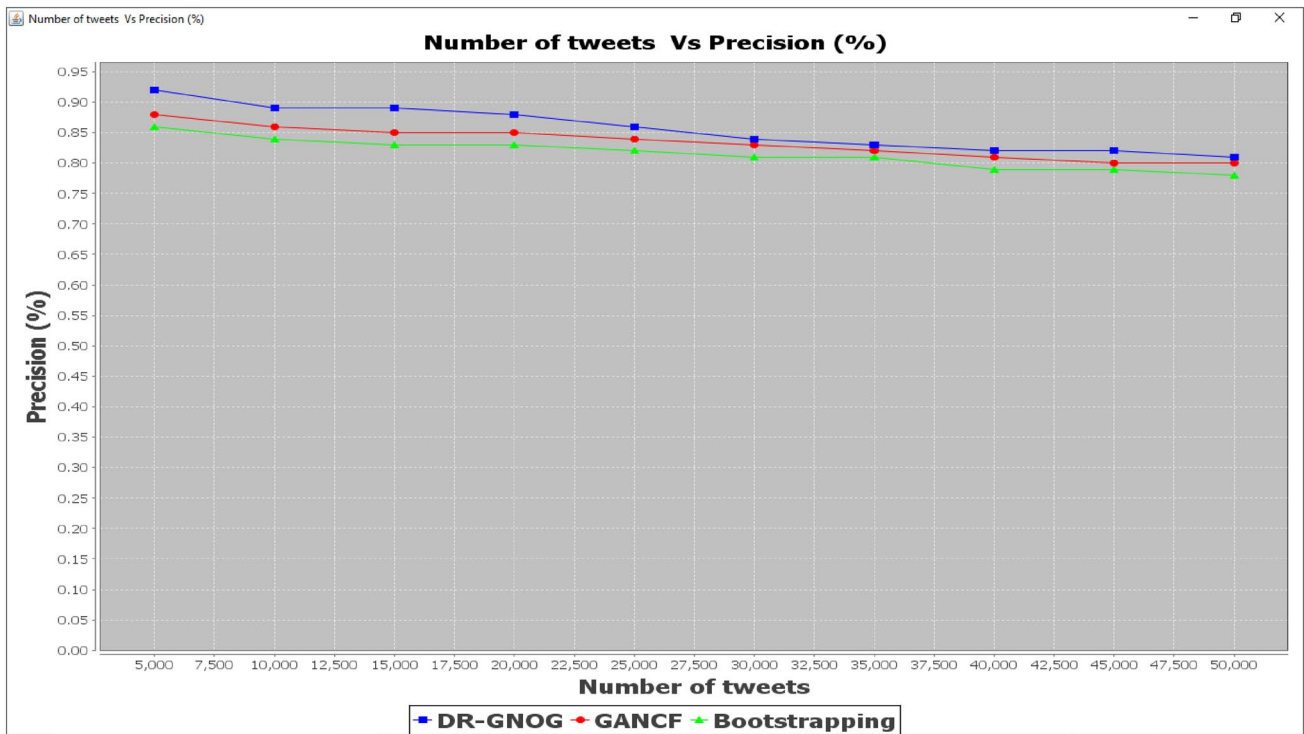


Fig. 10 Precision of three recommendation method [changes made] using sentiment 140 dataset

Table 9 Average precision [experimented with increased number of tweets] using sentiment 140 dataset

Number of tweets	Precision (%)		
	DR-GNOG	GANCF	Bootstrapping
5000	0.92	0.88	0.86
10,000	0.89	0.86	0.84
15,000	0.89	0.85	0.83
20,000	0.88	0.85	0.83
25,000	0.86	0.84	0.82
30,000	0.84	0.83	0.81
35,000	0.83	0.82	0.81
40,000	0.82	0.81	0.79
45,000	0.82	0.80	0.79
50,000	0.81	0.80	0.78

Table 10 Average precision [experimented with increased number of tweets] using coronavirus tweets NLP text classification dataset

Number of tweets	Precision (%)		
	DR-GNOG	GANCF	Bootstrapping
4000	0.93	0.91	0.88
8000	0.91	0.89	0.87
12,000	0.90	0.87	0.85
16,000	0.89	0.86	0.84
20,000	0.89	0.86	0.84
24,000	0.88	0.85	0.82
28,000	0.86	0.84	0.82
32,000	0.85	0.83	0.81
36,000	0.84	0.82	0.80
40,000	0.84	0.82	0.79

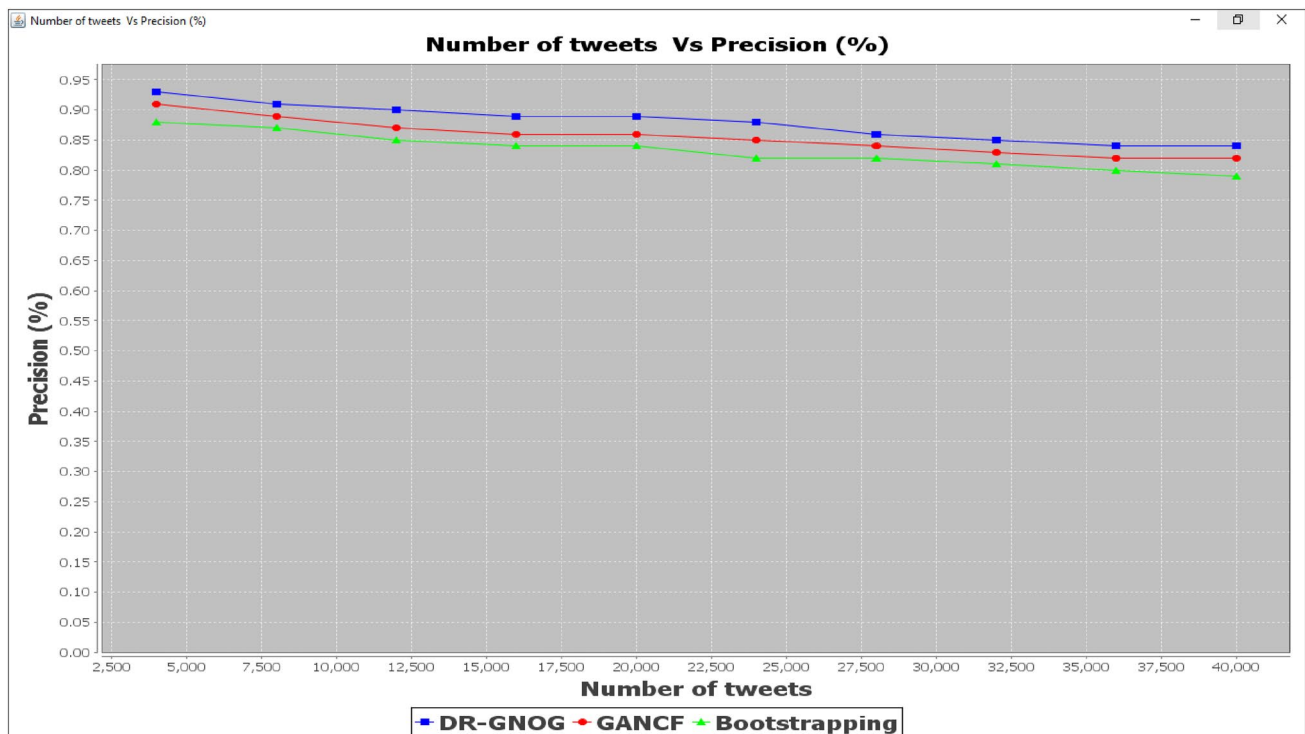


Fig. 11 Precision of three recommendation method [changes made] using coronavirus tweets NLP text classification dataset

such as GANCF and Bootstrapping, the experiments are conducted. From this result, the precision rate was found to be comparatively higher using DR-GNOG for two datasets upon comparison with GANCF and Bootstrapping. The reason behind the higher precision rate was due to the application of Deep Recurrent Neural Network based on Long Short Term Memory (LSTM). By applying this function, where the forget gate is applied to eliminate the irrelevant tweet contents. With this, the precision using DR-GNOG for sentiment 140 dataset was found to be improved by 3% and 5% compared to GANCF and Bootstrapping. In a similar manner, the precision using DR-GNOG for corona virus tweets dataset was said to be improved by 3% and 6% compared to GANCF and Bootstrapping respectively.

6 Conclusion

The recommendation systems service is a refining service to confront against information excess that are progressively increasing the thoughtfulness of the researchers and the academics. On previous research work which specify that there is a requirement to design an effective recommendation method to provide accurate positioning of information targets and smooth resource deployment. In specific, there is a requirement to address time awareness and accuracy

involving huge volumes of data that with user preference transposes to validate the enhancement of recommendation performance. Hence, this work address user preference changes via a multi-agent model to develop a Deep Recurrent Gaussian Nesterov's Optimal Gradient (DR-GNOG) method that provides recommendations in an accurate and timely manner with a high recall rate. First, a Confidence-aware Tweet Rank Social Network model was designed that in turn classified the relevant tweets in an optimized manner. Next, the Deep Recurrent Predictive Recommendation model was designed to obtain accurate recommended tweets. The experimental results show that the DR-GNOG method can get better results in terms of recommendation accuracy by 34%, time by 27%, and recall 4% on Sentiment140 dataset than others, which fully shows that applying it to deep learning can enhance recommendation performance of the method. Also, the DR-GNOG method gets the better performance of recommendation accuracy, recall, precision is enhanced by 15%, 7%, 4%, and the time is minimized by 24% with aid of coronavirus tweets dataset.

References

- Aivazoglou M, Roussos AO, Margaris D, Vassilakis C, Ioannidis S, Polakis J, Spiliotopoulos D (2020) A fine-grained social network

- recommender system, Social network analysis and mining. Springer, New York [Bootstrapping]
- Angelov PP, Filev DP (2004) Flexible models with evolving structure. *Int J Intell Syst* 19(4):327–340
- Angelov PP, Gu X (2009) Empirical approach to machine learning, vol 800. Springer Nature, New York (ISBN: 978-3-030-02383-6)
- Angelov P, Soares E (2020) Towards explainable deep neural networks (xDNN). *Neural Netw* 130:185–194
- Azaouzi M, Romdhane LB (2018) An efficient two-phase model for computing influential nodes in social networks Using Social Actions. *J Comput Sci Technol*, Springer
- Bakliwal K, Dhada MH, Palau AS, Parlikad AK, Lad BK (2018) A multi agent system architecture to implement collaborative learning for social industrial assets, international federation of automatic control. Elsevier, Amsterdam
- Castellini J, Oliehoek FA, Savani R, Whiteson S (2021) Analyzing factorizations of action-value networks for cooperative multi-agent reinforcement learning, autonomous agents and multi-agent systems. Springer, New York
- Develer U, Akar M (2020) Higher-order cluster consensus of a multi-agent network with continuous-time dynamics, *Transactions of the Institute of Measurement and Control*
- Kemmerer RA (2003) Cybersecurity. In: 25th International Conference on Software Engineering, 2003. Proceedings, pp 705–715. <https://doi.org/10.1109/ICSE.2003.1201257>.
- Falsone A, Margellos K, Prandini M, Garatti S (2020) A scenario-based approach to multi-agent optimization with distributed information, International federation of automatic control. Elsevier, New York
- Grammatico S (2017) Proximal dynamics in Multi-Agent Network Games, *IEEE Transactions on Control of Network Systems*
- Gronauer S, Diepold K (2021) Multi-agent deep reinforcement learning: a survey, *Artificial intelligence review*. Springer, New York
- Mastrangelo JM, Baumann D, Trimpe S (2019) Predictive triggering for distributed control of resource constrained multi-agent systems, international federation of automatic control. Elsevier, Amsterdam
- Kanwal S, Nawaz S, Malik MK, Nawaz Z (2021) A Review of Text-Based Recommendation Systems, *IEEE Access*
- Kishore M, Kulkarni SB (2016) Approaches and challenges in classification for hyperspectral data: a review. In: 2016 international conference on electrical, electronics, and optimization techniques (ICEEOT). IEEE, pp 3418–3421
- Li Y, Liu J, Ren J (2019) Social recommendation model based on user interaction in complex social networks. *PLoS ONE*. <https://doi.org/10.1371/journal.pone.0218957>
- Luo Y, Iyengar G, Venkatasubramanian V (2016) Soft regulation with crowd recommendation: coordinating self-interested agents in sociotechnical systems under imperfect information. *PLoS ONE*. <https://doi.org/10.1371/journal.pone.0150343>
- PremSankar C, Aharaf S, Satheesh Kumar K (2016) Learning from bees: an approach for influence maximization on viral campaigns. *PLoS ONE*. <https://doi.org/10.1371/journal.pone.0168125>
- Suhaim AB, Berri J (2021) Context-Aware Recommender Systems for Social Networks: Review, Challenges and Opportunities, *IEEE Access*
- Sutskever I, Martens J, Dahl G, Hinton G (2013) On the importance of initialization and momentum in deep learning. In: International conference on machine learning. PMLR, pp 1139–1147
- Tampuu A, Matiisen T, Kodelja D, Kuzovkin I, Korjus K, Aru J, Aru J, Vicente R (2017) Multiagent cooperation and competition with deep reinforcement learning. *PLoS ONE*. <https://doi.org/10.1371/journal.pone.0172395>
- Wan L, Xia F, Kong X, Hsu C-H, Huang R, Ma J (2021) Deep Matrix Factorization for Trust-Aware Recommendation in Social Networks. nN: *IEEE Transactions on Network Science and Engineering*, Vol. 8, No. 1, Jan 2021 [GRU-Attention Neural Collaborative Filtering (GANCF)]
- Yanzhi Wu, Jiangping Hu (2019) Bipartite Consensus of high-order multi-agent systems, International federation of automatic control. Elsevier, New York
- Yao W, He J, Huang G, Cao J, Zhang Y (2014) A Graph-based model for context-aware recommendation using implicit feedback data, *World Wide Web*. Springer, New York