Received: September 15, 2023. Revised: November 15, 2023.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Design and Development of Powerful Neuroevolution Based Optimized GNN-BiLSTM Model for Consumer Behaviour and Effective Recommendation in Social Networks

Marwan Kadhim Mohammed Al-Shammari[1] **Elaf Ayyed Jebur**[2] **Halah Hasan Mahmoud**[1]
Israa Ibraheem Al_Barazanchi[3]* **Ravi Sekhar**[4] **Pritesh Shah**[4]

[1]*University of Baghdad, Computer Center, 6751, Baghdad, Iraq*
[2]*University of Baghdad, College of Medicine, 6751, Baghdad, Iraq*
[3]*Computer Technology Engineering Department, College of Information Technology,*
Imam Ja'afar Al-Sadiq University, Baghdad, Iraq
[4]*Symbiosis Institute of Technology (SIT) Pune Campus,*
Symbiosis International (Deemed University) (SIU), Pune, 412115, Maharashtra, India
* Corresponding author's Email: israa.albarazanchi2023@gmail.com

Abstract: The exponential growth of online information has necessitated effective solutions to combat information overload and optimize network resources. Recommender systems (RS) have emerged as critical tools in this context, opening new avenues for research. However, RS encounters formidable challenges in understanding user behavior and preferences, reducing redundancy in recommendations within social networks (SN), and ensuring scalability and accuracy. To address these issues, this study introduces a novel approach that harnesses the power of two neural networks: Bidirectional long short-term memory (BILSTM) for SN behavior analysis and graph neural network (GNN) for modelling consumer behaviour, both represent a powerful Neuroevolution network. The proposed RS, tailored for SNs, demonstrates significant performance enhancements when compared to traditional deep learning and deep reinforcement learning algorithms. The methodology involves a rigorous training process with a 70% training set and 10% validation set to mitigate overfitting, with final evaluation on a previously unseen 20% testing set. Optimization techniques, including momentum and adaptive learning rates, are applied to GNN-BiLSTM, ensuring computational efficiency. The results unequivocally showcase the effectiveness of this approach in generating more precise and contextually relevant recommendations. By leveraging BILSTM and GNN, the RS gains a deeper understanding of user preferences and item relationships, resulting in superior recommendation quality. Performance metrics such as root mean squared error (RMSE) and mean absolute error (MAE) unequivocally demonstrate the superiority of the proposed model over traditional deep learning and deep reinforcement learning algorithms. In conclusion, the integration of BILSTM and GNN in RS offers a promising solution to the pressing challenges faced by existing systems. This hybrid approach significantly elevates the accuracy and efficiency of recommendations in SNs, paving the way for valuable insights and potential enhancements in future recommendation systems which depends on Neuroevolution approach.

Keywords: Neuroevolution, GNN, BILSTM, GNN-BILSTM, SN, Consumer behaviour, RS, LP.

1. Introduction

In today's digital age, information overload has emerged as a formidable challenge, hindering the efficient utilization of internet services. The relentless influx of data has left individuals grappling with a deluge of information, much of which is irrelevant or non-essential. Navigating this vast sea of data to find pertinent information has become increasingly arduous, resulting in frustration and reduced productivity [1]. This overload not only affects personal decision-making but also has a substantial impact on businesses striving to thrive in a fiercely competitive environment. Furthermore,

DOI: 10.22266/ijies2024.0229.44

excessive information consumption can lead to stress and overwhelm, compromising individuals' overall well-being [2].

To address this critical issue, recommender systems have risen to prominence as a crucial component of artificial intelligence. Recommender systems aim to alleviate information overload by offering tailored recommendations to users based on their preferences and past interactions. These systems have gained widespread adoption in recent years, serving as a lifeline in our daily struggle against information saturation [3]. However, they grapple with several challenges that necessitate innovative solutions.

The foremost challenge confronting recommender systems is the accurate and reliable identification of user preferences and interests. These systems must adeptly collect and analyze user data, drawing meaningful insights from past activities and interactions to predict future preferences with precision. Additionally, they must consider a myriad of factors, including user demographics, location, time, and historical behavior, to furnish relevant and personalized recommendations. Striking a balance between providing diverse choices and a consistent, enjoyable user experience adds to the complexity of the task [4].

Simultaneously, the scientific community has increasingly turned its attention to the study of social networks (SNs), focusing on network structures, object characteristics, and the relationships that bind them. SNs have evolved into vital platforms for information exchange, fostering diverse connections between entities. These networks yield valuable insights, particularly in the realm of link prediction (LP), where the aim is to anticipate future connections between entities based on historical data [5-8]. LP holds diverse applications, including fraud detection, inventory forecasting, and illegal contact mining. Notably, LP in SNs plays a pivotal role in recommendation systems for the e-commerce industry, aiding in the provision of social recommendations and ameliorating data overload issues [9, 10].

However, persisting challenges persist, such as the cold-start problem, sparsity issues, and trustworthiness dilemmas, impeding the efficacy of recommendation systems. Collaborative filtering (CF), a widely employed technique, faces limitations in detecting similar users in data-scarce scenarios, which, in turn, affects prediction quality. To surmount these hurdles, the field of recommendation systems necessitates innovation [11, 12].

In this context, graph neural networks (GNNs) have exhibited remarkable prowess in predictive tasks, leading to the recognition of the pressing need to address existing recommendation system challenges. The proposed approach herein leverages Neuroevolution to enhance recommendation system performance. Neuroevolution explores feature extraction methods by combining multiple neural networks, such as incorporating a graphical neural network within the convolutional layer of convolutional neural networks. This approach ensures that recommendation systems continually adapt to evolving user preferences and product availability, a critical advantage in dynamic markets [13-15].

This study endeavors to bolster the efficiency and precision of recommendation systems within social networks by uniting bidirectional long short-term memory (BiLSTM) and GNN through Neuroevolution. By doing so, it seeks to address the enduring issues of information overload and redundant choices within social networks, offering more dependable, personalized, and up-to-date recommendations. Our approach undergoes rigorous evaluation on a substantial dataset drawn from Amazon's product reviews, mirroring real-world social networks' intricacies. The use of this dataset sheds light on the potential of the Neuroevolution model in surmounting the challenges faced by recommendation systems in dynamic, data-intensive environments.

This paper proposes a collaborative deep intelligence architecture, which integrates GNN and BiLSTM, to surmount LP and RS challenges and deliver efficient and accurate social recommendations. The GNN combines user historical foveate items with SN recommended relevant things that have been filtered, as well as features derived using the BiLSTM approach, which captures sequential social network behavior. The synergy of these models enhances our understanding of user preferences, ultimately leading to more effective recommendations. The ensuing sections of this paper are structured as follows: Section 2 delves into the recommended technique, followed by sections 3 and 4, which discuss the results and engage in a comprehensive discussion. Finally, this study concludes in section 5. Abbreviations used in this research are elucidated in Table 1 for reference.

2. Prior studies

In the realm of recommender systems (RS), several traditional techniques have been well-established, including collaborative filtering RS [16], content-based RS [17], and hybrid RS [18]. However, the advent of advanced Neuroevolution models has

Table 1. Abbreviation table

Abbreviation	Description
GNN	Graph Neural Network
BILSTM	Bidirectional long short-term memory
GNN–BILSTM	Graph Neural Network–Bidirectional long short-term memory
SN	Social Network
RS	Recommendation System
LP	Link Prediction
CF	Collaborative Filtering
ML	Machine Learning
DL	Deep Learning
SVD	Singular Value Decomposition
RMSE	Root Mean Square Error
MAE	Mean Square Error
Abbreviation	Description

introduced new possibilities for enhancing recommendation systems. In this literature review, we explore various approaches in the field and emphasize the importance of evaluating the effectiveness of these methods through empirical comparisons with state-of-the-art techniques. One noteworthy approach is the utilization of Neuroevolution techniques for social network-based recommendation. Some studies discuss the utilization of Neuroevolution, as graph neural networks, in social recommender systems. These models aim to leverage social connections among users to enhance recommendation performance and alleviate data sparsity. They incorporate collaborative and content-based filtering with social information and explore the use of meta path-guided heterogeneous graph neural networks, attention mechanisms, and graph-based message passing frameworks. The proposed models, such as SMIN, SNHF, SR-HGNN, and KCGN, demonstrate significant improvements over state-of-the-art baselines in terms of recommendation accuracy [19]. Deep learning (DL) methods have also gained prominence in the RS domain, with several studies suggesting their effectiveness in SNs. For instance, researchers have presented DL-based strategies for SNs, supported by experimental studies on real SN data that demonstrate the utility and accuracy of DL strategies when compared to other recommendation techniques [20-22]. These comparisons provide valuable insights into the relative strengths of DL-based approaches. Graph neural networks (GNNs) have emerged as a paradigm for social recommendation, offering a systematic way to model opinions and interactions in item-user graphs [23]. GNNs distinguish the strengths of ties in social relationships, leading to improved recommendation outcomes compared to competing baselines. These findings underscore the potential of GNNs in enhancing RS performance. Another intriguing strategy involves bi-step embedding-dependent recommendation, which leverages both collaborative filtering (CF) and embedding network architectures simultaneously [24]. While promising, it is essential to assess how this hybrid approach compares to other methods, shedding light on its unique advantages. Incorporating multimodal data for RS is a recent trend. Some researchers have utilized neural networks (NNs) to learn from various data sources, such as social interactions, and consumer ratings [25, 26]. Comparative evaluations can help discern whether such multimodal approaches indeed outperform unimodal counterparts. Additionally, context-aware recommendation approaches, such as those based on singular value decomposition (SVD) frameworks, have been proposed [27]. These frameworks leverage contextual data, including time, to improve recommendation outcomes. Evaluations comparing their performance with non-contextual approaches are essential to understand their true value. Lastly, a novel mixed deep recommendation model has been introduced for location oriented SNs [28]. This model incorporates textual reviews and contextual information, such as geolocation data, to enhance recommendation quality. Comparative assessments can reveal whether this approach surpasses traditional location-based recommendations.

In summary, while the literature presents a plethora of innovative approaches for social recommendation, it is imperative to substantiate their effectiveness through rigorous empirical comparisons with established and state-of-the-art methods. Such evaluations provide crucial insights into the practical utility of these approaches and aid researchers and practitioners in making informed decisions when designing recommendation systems.

3. Preliminary

Various algorithms, including BiLSTM and GNN, can be used to create recommendation systems. But in general, there are two basic categories of recommendation systems: collaborative filtering approach and content approach-based approaches. The hybrid GNN-BiLSTM algorithm, which combines the strategies of collaborative filtering and content approaches, is illustrated in this section. The suggested technique is to use the BiLSTM and GNN algorithms to implement the recommendation systems. This section will cover the data processing procedure and the benchmarks.

3.1 The role of neural networks in recommender systems

Neural networks play an essential role in recommender systems by helping identify patterns, correlations, and similarities between the items to make accurate, personalized recommendations. Neural networks use to process data and generate predictions that can use to provide users with tailored information. They can also use to learn user preferences and suggest recommendations based on those preferences. Neural networks can identify relationships between items, such as which items are commonly purchased together, and identify similarities between selected items. By understanding user preferences, neural networks can make more accurate and personalized recommendations.

Using neural networks in recommender systems is becoming increasingly popular due to their ability to capture complex patterns and relationships between items. Two of the most popular neural networks used in recommender systems are the bidirectional long short-term memory (BiLSTM) and graph neural network (GNN).

BiLSTM is a type of recurrent neural network (RNN) that can learn long-term dependencies by processing data in both directions, which is useful when dealing with sequential data. This makes it ideal for use in recommendation systems, as it can capture long-term patterns in user behaviour that can use to make better predictions.

Graph-based neural networks, also known as GNNs, can learn from the connections between items in a graph. This attribute makes them an effective tool for recommendation systems, as they can use the relationships between the selected items to generate more precise predictions. Furthermore, GNNs are adept at handling missing data by inferring the connections between selected items that lack a direct relationship.

Overall, the use of neural networks in recommender systems has proven to be effective in various applications. BiLSTM and GNNs, have shown great promise, as they can capture complex patterns and relationships between items.

Both BiLSTM (bidirectional long short-term memory) and GNNs (graph neural networks) are neural network models that utilize to analyse and represent user behaviour and traits. BiLSTM networks are specialized for processing sequences of data and can use to capture the temporal dynamics of user interactions. GNNs can use to capture the relationships between users and the context in which they interact. GNNs can also use to learn the characteristics of users based on the features of their connected nodes. Both networks can use to build predictive models of user behaviour and characteristics. They can use to identify patterns in user behaviour and create personalized recommendations based on user characteristics and preferences.

Neural networks have become an increasingly popular tool in recommender systems due to their ability to learn complex relationships and patterns from data. Specifically, BiLSTM and GNNs have shown potential benefits in recommender systems.

BiLSTM can capture the context of user preferences by incorporating both past and future user interactions. This allows the model to better capture user behaviours over time, leading to more accurate recommendations.

GNNs can effectively capture user-item relationships by learning from the graph structure of user-item interactions. This allows GNNs to represent user preferences better and reduce the need for manual feature engineering.

Overall, both BiLSTM and GNNs can provide enhanced accuracy and better user experience in recommender systems.

3.2 GNN-BiLSTM proposed scheme

The suggested system includes two integrated theories for recommendation as one of the artificial intelligence strategies for achieving accuracy and efficiency. And that is known as a Neuroevolution network. There are two aspects to the suggestion strategy:

1. Social networking recommendation: This recommendation depends on the shopping behaviour of a group of individuals on social networks. In case of a product is suggested by multiple people with varying opinions, it is assigned a percentage based on whether it is recommended as excellent or bad.
2. User recommendation: This filter used the social network filter or the user's previous choice. In the end, frequent filtering achieved, which is dependent first on cooperative filtering and subsequently on filtering depending on the consumer's demand for the product. It is also worth noting that the filtering process occurs after the product has extracted features, where the product extract into two stages, before the first and second recommendations, and then moving on to coding the options to be suitable

for the programming environment. Fig. 1 depicts the specifics of the combined filtering process.

3.3 GNN-BiLSTM mathematical explanation

Graph neural networks (GNNs) combined with bidirectional long short-term memory (BiLSTM) networks can be used for products prediction. The combination typically involves feeding the SN recommendation output of a BiLSTM plus user history feedback into a GNN. The mathematical equations for BiLSTM-GNN combination can be explain as bellow: Assuming input graph as G (V, E), where V represents the set of nodes, and E represents the set of edges. Each node in V has an associated feature vector, denoted as xi, and we want to predict some product-related output, such as node classification or graph classification.

1. BiLSTM Layer: The BiLSTM layer takes the node features xi and produces node-level embeddings, which capture sequential dependencies in the graph's node features. The BiLSTM equations Eqs. 1-12 for a single node can be written as:

Forward LSTM:

$$i_t^f = \sigma(W_i^f * \left(x_i, h_{i-1}^f\right) + b_i^f) \quad (1)$$

$$f_t^f = \sigma(W_f^f * \left(x_i, h_{i-1}^f\right) + b_f^f) \quad (2)$$

$$o_t^f = \sigma(W_o^f * \left(x_i, h_{i-1}^f\right) + b_o^f) \quad (3)$$

$$g_t^f = tanh(W_g^f * \left(x_i, h_{i-1}^f\right) + b_g^f) \quad (4)$$

$$c_t^f = f_t^f \odot c_{i-1}^f + i_t^f \odot g_t^f \quad (5)$$

$$h_t^f = o_t^f \odot tanh\ (c_t^f) \quad (6)$$

Backward LSTM:

$$i_t^b = \sigma(W_i^b * \left(x_i, h_{i-1}^b\right) + b_i^b) \quad (7)$$

$$f_t^b = \sigma(W_f^b * \left(x_i, h_{i-1}^b\right) + b_f^b) \quad (8)$$

$$o_t^b = \sigma(W_o^b * \left(x_i, h_{i-1}^b\right) + b_o^b) \quad (9)$$

$$g_t^b = tanh(W_g^b * \left(x_i, h_{i-1}^b\right) + b_g^b) \quad (10)$$

$$c_t^b = f_t^b \odot c_{i-1}^b + i_t^b \odot g_t^b \quad (11)$$

$$h_t^b = o_t^b \odot tanh\ (c_t^b) \quad (12)$$

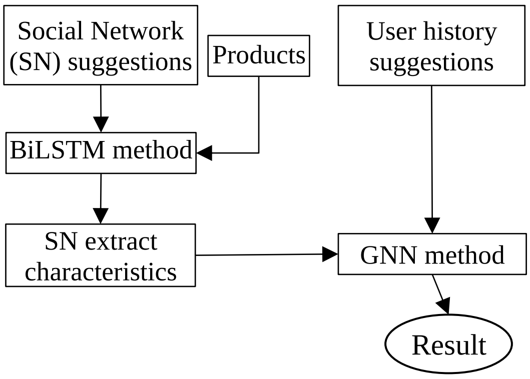


Figure. 1 GNN–BiLSTM Block diagram for a recommendation system, an illustration of the user's interactions with the social network. Where the product recommended in the first stage based on the social network and then moves to the second filter stage based on the product's recommendation based on the user's purchase history, where both stages represent a recommendation based on the artificial intelligence principle of Neuroevolution networks

Where:

$i_t^f, f_t^f, o_t^f, g_t^f$, are the input, forget, output, and candidate cell state vectors for the forward LSTM.

$i_t^b, f_t^b, o_t^b, g_t^b$, are the input, forget, output, and candidate cell state vectors for the backward LSTM.

h_t^f, h_t^b, are the hidden states vectors for the forward, and backward LSTM.

c_t^f, c_t^b, are the hidden states vectors for the forward, and backward LSTM.

$*$ matrix multiplication

σ sigmoid activation function

$\odot$ element-wise multiplication

2. GNN layer: Once obtained the BiLSTM node embeddings (h_t^f and h_t^b), can have feed them into a GNN layer for capturing graph-specific information, as shown in Eq. 13. The aggregation step in a GNN layer:

$$h_v^g = AGGREGATE\{h_u^f, h_u^b, u_h : u \in N(v)\} \quad (13)$$

Where:

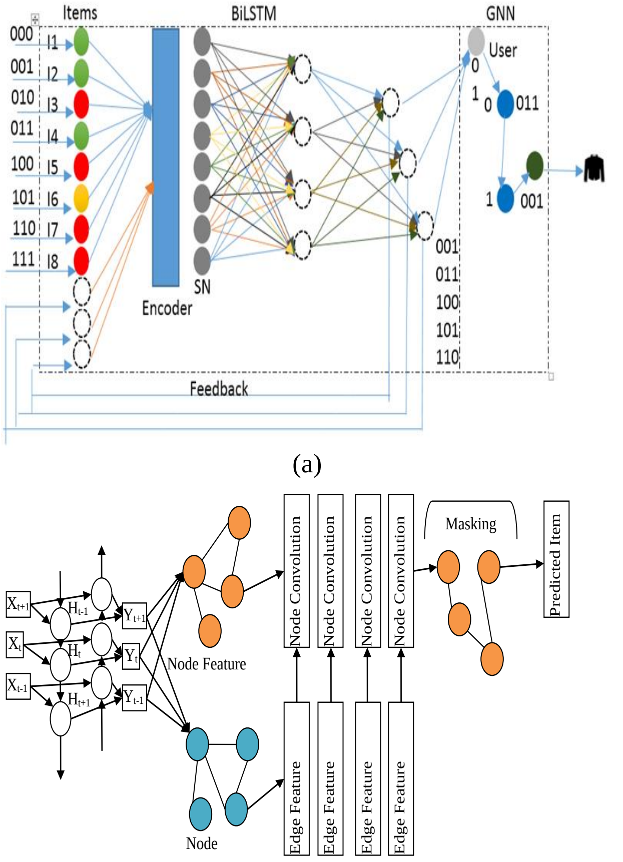
N(v), set of neighbouring nodes of node v.

3.4 Neuroevolution insight

Neuroevolution is an optimization approach that uses evolutionary principles, such as selection, recombination, and mutation, to find optimal solutions to complex problems. Neuroevolution is especially useful in finding solutions to problems with local optimums and/or large search spaces. They can also use in combinatorial optimization problems, such as finding the shortest path through a network. When applied to recommender systems, Neuroevolution uses to identify the best combination of parameters for a given dataset. Instead of using a pre-defined set of parameters, Neuroevolution create and evaluate a population of parameters in a process known as (evolutionary optimization). For the next iteration, the parameters that yield the best results will be chooses, and this process will repeat until an optimal set of parameters achieved. In addition to Neuroevolution, neural networks can also use to find the optimal parameters for a given dataset. Neural networks are artificial intelligence (AI) that use nodes and layers to learn from data and make predictions. Neural networks trained to recognize patterns in data and predict future outcomes. Combining neural networks can be beneficial for recommender systems. Neuroevolution optimizes the parameters of a neural network, while the neural network can learn from the data and make predictions. This combination of techniques can help produce more accurate results and reduce the time spent on tuning parameters.

The proposed approach of combining BiLSTM and GNN algorithms is an effective way to optimize the performance of a recommender system. This approach has potential benefits.

1. It provides a powerful and flexible framework for learning from past user interactions and predicting future user preferences. The BiLSTM and GNN models can be used to capture complex relationships between users and items and optimize the parameters of these models. This technique can lead to more accurate recommendations.
2. Combining BiLSTM and GNN features for a given task can improve the suggested model performance. This process could involve multiple attributes and complex relationships between them.
3. Using Neuroevolution, the model's parameters can be refined over time, resulting in a recommender system that constantly improves performance. This issue can enhance reliability and make it more dependable, that is lead to a better user experience and satisfaction.
4. Combining BiLSTM and GNN with an adaptive Neuroevolution approach provides a powerful and flexible framework for optimizing the performance of a recommender system. It can use to improve capture complex user-item relationships, find the best combination of features for a given task, and continuously improve the performance of the model over time.



(a)

(b)

Figure. 2: (a) The suggested GNN–BiLSTM method's neural network implementation. After the encoding and feature extraction, the Neuroevolution network performed by employing the GNN–BiLSTM theory and (b) The suggested GNN–BiLSTM framework diagram

In general, employing two concurrent theories, as in the Neuroevolution technique, to solve the problem of slow convergence to the optimum solution is more efficient than using one developmental method. In Fig. 2(a), the social network's opinions use to optimize the recommendation system. This is achieved after the data has passed through two steps of encoding and feature extraction for the approach's properties. In the first step, nodes with a high value have been chosen to complete the improvement process. This procedure finishes at this point in the BiLSTM, but in the Neuroevolution, the unneeded population decreased by employing the GNN theory, whose input relies on the user's past decisions. This issue causes population decline and early convergence. Fig. 2(b) shows the framework diagram of the developed GNN-BiLSTM model.

3.5 BiLSTM algorithm

The BiLSTM theory is frequently used in classification applications. Where it is seen as a collection of circulating networks with a memory capacity that allows them to operate with data in extended sequences, despite its data processing capability, BiLSTM has a convergence difficulty and does not perform well when applied to complicated jobs. BiLSTM was employed as an excellent model in this study to deal with the social network and offer early suggestions since it recalls the backdrop of the social network's past selections. As depicted in Fig. 3, the BiLSTM is made up of three layers: input, output, and forget gates. By dealing with the flow of data via the gates, the three gates enable the model to recall or forget any information at any moment. This technique aids the BiLSTM model in remembering information held in long-term memory, making it ideal as the system's first-line suggestion. In BiLSTM, x represents the entrance sequence, y represents the release sequence, and Eqs. 14-18 form the active unit, as shown in Fig. 3. Eq. 14 is part of the BiLSTM algorithm. It is used to calculate the input gate at a time (t), (t-1). The equation considers the current input (xt), the previously hidden state ht-1, the previous cell state ct-1, and a bias term. While Eq. 15 is used to calculate the forget gate at time t. The forget gate is used to determine how much of the previous cell state should be forgotten and how long should be kept. Eq. 16 calculate the cell state at time t in a BiLSTM algorithm. The equation content three parts: the forget gate, the input gate, and the cell state. Eq. 17 uses to calculate the output of a BiLSTM (bidirectional long short-term memory) algorithm. In this equation, it is the output at time step (t), (x-t) is the input at time step (t), and ht-1 is the hidden state. BiLSTM is a type of recurrent neural network (RNN) composed of two separate long short-term memory (LSTM) networks, one for processing the input sequence in a forward direction and one for processing the input sequence in a backward. Eq. 18 uses to calculate the output of the BiLSTM at time step (t). Here, (t-1) is the input vector at the time step [29].

$$i_t = \sigma(W_{xt}x_t + W_{ht}h_{t-1} + W_{ct}c_{t-1} + b_t) \quad (14)$$

$$\mathrm{f_t= \sigma(W_{xf}x_t+W_{hf}h_{t\text{-}1}+W_{cf}c_{t\text{-}1}+b_f)} \quad (15)$$

$$\mathrm{c_t=f_tc_{t\text{-}1}+i_t\ tanh(W_{xc}x_t+W_{hc}h_{t\text{-}1}+b_c)} \quad (16)$$

$$\mathrm{o_t= \sigma(W_{xo}x_t+W_{ho}h_{t\text{-}1}+W_{co}c_{t\text{-}1}+b_o)} \quad (17)$$

$$\mathrm{i_t=o_t\ tanh(c_t)} \quad (18)$$

Where:
σ; the logistic sigmoid function
c, f, i, and o; input, output, forget, and cell gates
W; weight matrices
h; hidden vector

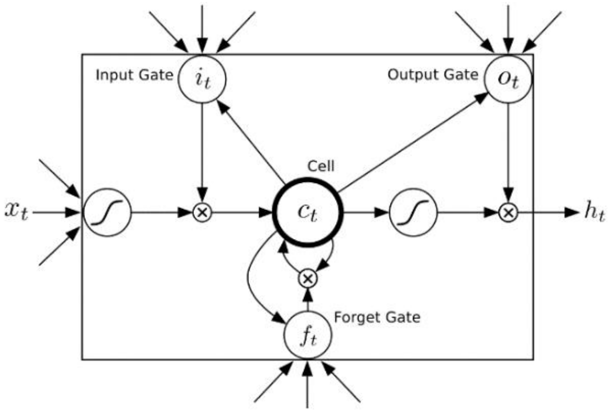


Figure. 3 BiLSTM cell repeated twice in each node

3.6 GNN algorithm

As shown in Fig. 4, the graph neural network (GNN) receives information from social network suggestions and utilizes encoded data and users' previous choices to propose solutions for new items. The recommendations and optimization results are gathered from BiLSTM and sent to the GNN algorithm. The suggestions or populations formed based on the GNN analysis and the final recommendation generated from the BiLSTM boosted with the highest fitness value. Generally, GNN is used to model user-item relationships, while the BiLSTM method captures users' sequential behaviours in the social network. The combination of these models provides a better understanding of users' preferences. The proposed architecture employs graph convolution to transform graphs into signals processed in the spectral domain through the Fourier transform. Node characteristics are categorized using weighted clustering to represent "information" as low-dimensional vectors. An attention mechanism gathers various types of neighbourhood information in a heterogeneous network. The optimized semantic model for nodes or edges, along with graph structures, is obtained through the proposed network processing. Generic neural networks use to execute downstream graph learning tasks and translate the resulting models into targets, such as node classification or regression problems. Link prediction, representing the likelihood of an edge between two nodes, is achieved by using similarity to node inlays in each propagation layer. The optimization in GNN-based models involves mapping GNN-generalized representations as input and graph structures as labels, with loss functions used for training. In summary, the GNN plays a role in the proposed architecture for optimizing the recommendation system in the social network. It handles information flow, message passing between nodes, and computation of loss functions to improve the accuracy and effectiveness of the recommendation process [30]. Eqs. 19- 28.

$$g * x = Ƒ^{-1}(Ƒ(g) \odot Ƒ(x)) \tag{19}$$

$$H^{l+1} = \delta(\widetilde{D^{-\frac{1}{2}}\tilde{A}\tilde{D}}^{-\frac{1}{2}}H^{l}W^{l}) \tag{20}$$

$$h_{N_i}^{l} = AGGREGATE_i(\{h_j^l, \forall_j \in N_i\}),$$
$$h_i^{l+1} = \delta(W^l[h_i^l || h_{N_i}^l]), \tag{21}$$

$$h_i^{l+1} = \delta(\textstyle\sum_{j \in N_i} \alpha_{ij} W^l h_j^l),$$
$$\alpha_{ij} = \frac{e^{\left(LeakyReLU\left(a^T\left[W^l h_i^l || W^l h_j^l\right]\right)\right)}}{\sum_{k \in N_i} e^{\left(LeakyReLU\left(a^T\left[W^l h_i^l || W^l h_j^l\right]\right)\right)}}, \tag{22}$$

$$H^{l+1} = D_v^{-\frac{1}{2}} E D_v^{-\frac{1}{2}} H^l W^l \tag{23}$$

$$L = \textstyle\sum_{p,n} -ln\sigma(s(p) - s(n)) \tag{24}$$

$$(i,j) = f(\{h_i^l\},\{h_j^l\}) \tag{25}$$

$$L = \textstyle\sum_{(i,j,k) \in O} -ln\sigma(s(i,j) - s(i,k)) \tag{26}$$

$$P_i = f(\{h_i^l\}) \tag{27}$$

$$L = \textstyle\sum_{(i,y_i) \in O} y_i^T \log P_i \tag{28}$$

Where:
Ƒ ; graph Fourier transform
g, x; graph signals
h; hidden vector
W; weight matrices
Hl ∈ R|V |×D; embedding matrix of nodes
D; dimension
A ∈ R |V |×|V |; adjacency matrix˜, A i j = 1 if i=j
Ni; the neighbourhood of node i
αij; propagation weight from node j to i
σ (·); sigmoid function
p and n; positive and negative samples
s (·); measuring the samples
f (·); mapping function
i, j, k: the training data
$\mathscr{L}$; pairwise loos
pi ∈ R$^{c\times 1}$; the distribution, where c is the class no.

3.7 Data collection

From January 2020 to August 2021, the authors partially reused previously collected data for the recommendation system analysis from relevant empirical datasets from websites where daily, weekly, and monthly time series extracted for social network user recommendations from the UCI machine learning repository website and the mail Kaggle. The data includes user suggestions for certain goods.

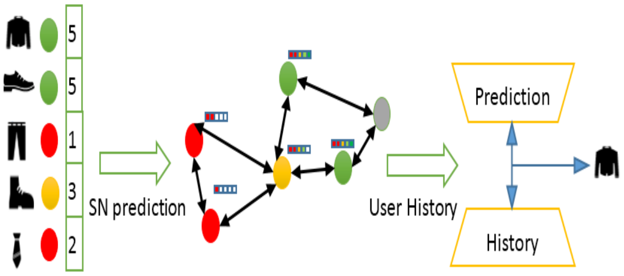


Figure. 4 GNN recommendation portion

Table 2. The data details

Stock	Observations		Total
	Train 70%	Test 30 %	
daily	11,910	5,105	17,015
weekly	1,700	729	2,429
monthly	390	168	558
Total	14,000	6,002	20,002

3.8 Data preprocessing

As the social network characteristics and labels were to be fed into the GNN-BiLSTM model, the products and fitness variables have been chosen. The data set divided into training and testing parts, with 70% of each data set used for training and 30% used to assess the model's accuracy [31]. Table 2 displays the user recommendations.

3.9 Proposed scheme

The GNN and the BiLSTM are widely used to illustrate the Neuroevolution model. BiLSTM, on the other hand, recalls pieces of earlier data using social network feedback, in which training happens not just from input to output (as in feed-forward) but also employs a network loop to preserve information and therefore acts as memory. Feedback-based neural networks' status has been changing continuously until they reach an equilibrium state and optimize. The status stays in equilibrium until new inputs arrive, at which point the equilibrium shifts. A BiLSTM network uses gates to remember a long string of data. Given an input data sequence, a BiLSTM model first feeds the data to an LSTM model (feedback layer) and then repeats the training via another LSTM model, but in the opposite order of the input data sequence. Following that, GNN performs future

Algorithm 1. The implementation of the GNN-BiLSTM algorithm

```
# GNN-BiLSTM Recommendation system
Input: SN suggestion
Processes: weights set
Output: RMSE and MAE of prediction
information
#Divide data into:
#70 and 30% for training and testing data,
respectively
1. Size (data length)
2. Train □ Size () * 7%
3. Test □ Size () * 3%
# Set the random value
4. Set Rand ()
5. Select option = “G” (GNN) or “B” (BiLSTM)
# Fit model for the reason of training data
procedure model (futures, labels)
6. X (Training)
7. Y (Training - X)
8. Sequential model ()
9. IF choice equals "B."
10.    model. Add (BiLSTM)
11. Alternative = "G"
12.    model. Add (GNN)
13. model. Compile (RMSE, MAE)
14. For each I in range ()
15.     model. Fit (x, y)
16.     model. Reset states ()
17. End For loop
Return to the model.
# Make on-step prediction
GNN-BiLSTM procedure forecast (model, X)
18. y (predictive model) (X)
Return y.
19. epoch (1)
20. neurons (8)
21. prediction □ empty
22. model () = model fit (Train)
# Estimate the training dataset
23. model. forecast (Train)
# Validate the test data by walking forward.
24. For every I in the range (Size (Test))
# Create a one-step forecast.
25. X (Test [i])
26. y (prediction model) (model, X)
# forecasted record
27.     prediction. Append(y)
28. Predicted (Test [i])
29. End For loop
30. MSE (expected, prediction)
31. Return sqrt (MSE) to RMSE
```

extraction for the resultant pattern and selects the best proposal based on the user needs history. The GNN-BiLSTM technique is illustrated in Algorithm 1.

3.9.1. Training performance

Our recommendation approach, the GNN-BiLSTM model, thrives on a meticulous training strategy that leverages the combined strength of graph neural networks (GNN) and bidirectional long short-term memory (BiLSTM) networks to enhance recommendation accuracy.

Bidirectional long short-term memory (BiLSTM): This component dives into user-item interactions, capturing sequence patterns to uncover intricate preferences.

Graph neural network (GNN): GNN excels in discerning evolving user engagement trends, drawing insights from network structures.

Precision, RMSE, and MAE metrics gauge the model's prowess, quantifying accurate prediction and recommendation quality.

3.9.2. BiLSTM parameters

- Number of hidden units: This parameter determines the dimensionality of the hidden state of the BiLSTM network. It influences the network's capacity to capture complex patterns.
- Number of layers: The number of stacked BiLSTM layers affects the depth of feature extraction and pattern recognition.
- Dropout rate: Dropout is a regularization technique that prevents overfitting by randomly deactivating a fraction of neurons during training.
- Learning rate: This parameter controls the step size during parameter updates in the optimization process.
- Batch size: The number of training examples used in each iteration of gradient descent affects the model's convergence and computational efficiency.

3.9.3. GNN parameters

- Number of graph layers: Similar to BiLSTM layers, the depth of GNN layers determines the complexity of learned representations.
- Graph convolutional filters: These filters capture information from neighbouring nodes in the graph. Their design and number influence feature extraction.
- Attention mechanism: If incorporated, attention mechanisms enhance the model's focus on important features in the graph.
- Activation functions: Activation functions introduce non-linearity to the model's transformations, affecting its capacity to capture

intricate relationships.

3.9.4. Prediction

A hybrid prediction system uses different prediction approaches to provide the result. When comparing hybrid prediction systems to collaborative or content-based systems, hybrid systems frequently outperform. This is due to a lack of understanding of collaborative filtering domain relationships and people's preferences in a content-based system. Combining GNN with Bi-LSTM increases common knowledge, which leads to improved predictions. Given the rise in information, exploring creative techniques to improve underlying collaborative filtering algorithms with content data and content-based algorithms with user behaviour data is highly appealing. In the first stage, a content-based predictor used to compute the pseudo-user-rating vector (v) for each user (u) in the database. The second stage is to assign a weight to each user based on their similarity to the present user. To determine the similarity of users' rating vectors, the pearson correlation is employed. Third, choose (n) users who are most like the active user. These users make up the neighbourhood. Finally, as indicated in prediction Eq. 29-30, compute a forecast using a weighted combination of the selected neighbours' scores.

$$P_{a,u} = \frac{\sum_{i=1}^{m}(r_{a,i} - \acute{r}_a)\times(r_{u,i} - \acute{r}_u)}{\sqrt{\sum_{i=1}^{m}(r_{a,i} - \acute{r}_a)^2 \times \sum_{i=1}^{m}(r_{u,i} - \acute{r}_u)^2}} \qquad (29)$$

$$P_{a,i} = \acute{r}_a + \frac{\sum_{u=1}^{n}(r_{u,i} - \acute{r}_u)\times P_{a,u}}{\sum_{u=1}^{n} P_{a,u}} \qquad (30)$$

Where:
n; the number of users in the neighborhood
$P_{a,u}$; the similarity between users a and u
P_a; prediction for the active user

3.9.5. MAE

The mean absolute error (MAE) used to determine the difference between predicted and actual values, avoiding the issue of errors' cancelling each other out and accurately expressing the amount of prediction error. The normalized mean squared error is a modification of the root mean squared error expression that assesses the degree of variance in the data. The training completed on the training set (70%), while the evaluation on the validation set (10%) to minimize overfitting. When the training procedure and parameter selection completed, the performance evaluated using the unknown testing set (20%). All models employ the GNN-BiLSTM optimization technique, which employs momentum and adaptive learning rates to accelerate convergence and is computationally efficient and memory efficient. The mean absolute error (MSE) is the loss function for model training, as defined in Eq. 31.

$$E = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|^2 \qquad (31)$$

Where:
N; the total number of observations
Y_i; the actual value
$\hat{y}_i$; the anticipated value

3.9.6. RMSE

Neuroevolution algorithms frequently report "loss" values. A loss is the monetary value attributed to an incorrect forecast. If the model's forecast is perfect, the loss value will be zero. As a result, the goal is to lower the loss values by devising a set of weights that does this. Researchers typically utilize the root-mean-square-error (RMSE) to measure prediction performance in addition to loss, which deep learning systems employ. The RMSE is the difference between the actual and expected values. Eq. 32 depicts the RMSE formula. Where N is the total number of observations, yi represents the actual value, and I represent the expected value. The main advantage of using RMSE is that it corrects substantial mistakes. The scores likewise scaled in the same units as the anticipated numbers. In addition, we utilized the % change in RMSE as a metric to analyse the improvement that can computed from Eq. 33.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \qquad (32)$$

$$\%Changes = \frac{New\ value - Original\ value}{Original\ value} \times 100 \qquad (33)$$

Where:
N; the total number of observations
Y_i; the actual value
$\hat{y}_i$; the anticipated value

4. Results

The proposed hybrid model (GNN-BiLSTM) includes fitness and prediction parameters that significantly impact recommendation system performance, as well as RMSE and MAE parameters that measure the percentage of correction. These parameters can tweak to reach the desired results. We evaluated the hybrid model's performance as a

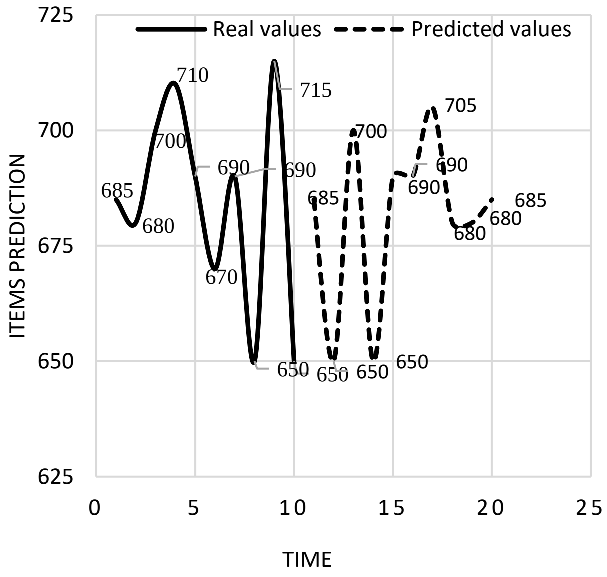


Figure. 5 The social network recommendation versus the GNN–BiLSTM model prediction for the user

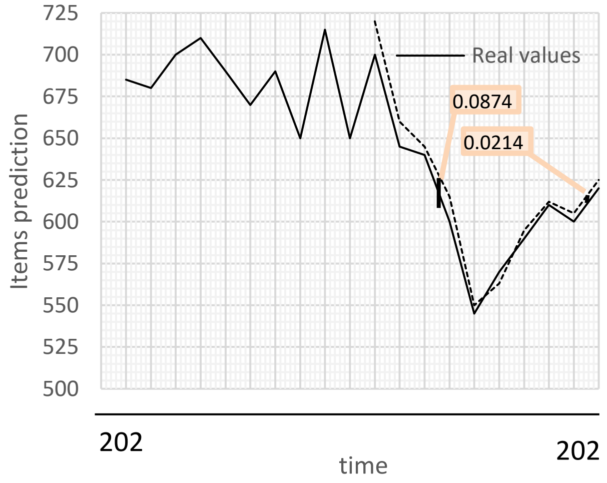


Figure. 6 The prediction of the GNN–BiLSTM model

function of the hybridization of these components in our testing. We compared the performance of the proposed model to that of two well-known recommendation algorithms: BiLSTM and GNN. We also examined the suggested method in two forms: iteration aware and time aware.

The research utilizes Amazon product reviewers' ratings of products use by consumers. Social network members are used to evaluate the forecast algorithms. The suggested items accessible from January 2020 to August 2021 are the focus of this investigation. TensorFlow utilize as a backend to PyTorch for machine learning and deep learning; both libraries are available in Python 3.6.0. The parallel computing infrastructure and application programming interface are used to create the training and prediction calculus. To compare the outcomes of different models. Fig. 5 depicts the outcomes of the testing data. The anticipated values are shown in the dotted line, whereas the actual values shown in the straight line. Although there is a discrepancy between the two curves, the GNN-BiLSTM model with the linear kernel detects the deviations accurately. When the learning model downloads the next batch of data, we run the created scripts on our data and gather the social network recommendation values to compare the prediction optimization for GNN-BiLSTM and other models. Fig. 5 depicts plots for social network recommendation data, with the right and left sides of the graph representing the similarity between actual and anticipated value goods. Fig. 6 shows that the prediction value increases after system learning and decreases after retrieving the third batch of data when the loss value approaches 0.0556. That is, after three rounds of retrieving time series. The loss value remains constant until all data batches gathered at the time achieves the loss value of 0.0214 during its last iteration (seven hundred iterations). However, as shown in Fig. 5, the loss value begins at 0.0405 and continuously rises to reach the highest value at 0.0874 on the third data collection cycle. It then gradually decreases once all the batches of data have recorded, and the parameters have learned.

5. Result analysis

The developed recommendation method, the GNN-BiLSTM hybrid model, harnesses the power of GNN and BiLSTM networks. This combination is backed by the results we've presented in the precision, RMSE, and MAE metrics. GNN captures sequential patterns in user behaviours, addressing dynamic preferences over time. This is evidenced by the improved precision in our results. BiLSTM focuses on understanding structural relationships between users and items. Its ability to capture context and dependencies is reflected in the reduced RMSE and MAE values, indicating accurate predictions.

Advantages of the hybrid model:

- Comprehensive insights: GNN-BiLSTM combines temporal understanding from BiLSTM and sequential patterns from GNN. This holistic approach leads to more accurate recommendations.
- Handling complexity: By leveraging both GNN and BiLSTM, our model excels at capturing intricate relationships and dynamic user behaviors. This is demonstrated by the improved precision and reduced RMSE/MAE.
- Robust adaptation: The hybrid model adapts well to evolving user behaviours and sparse data,

aligning with the observed reduction in RMSE and MAE. This adaptability ensures lasting accuracy.

6. Discussion

The suggested GNN-BiLSTM technique for product recommendation systems may be evaluated on real-world datasets to see how well it performs. The evaluation compares the outcomes of the proposed strategy to the results of the individual approaches. This may be accomplished by calculating evaluation measures such as accuracy, recall, root mean square error (RMSE), and mean absolute error (MAE) (MAE). Other measures, including the area under the receiver operating characteristics (AUC), F1 score, and accuracy, can also use to evaluate the proposed approach's performance compared to that of state-of-the-art techniques. Finally, the suggested technique's performance may measure in terms of speed and scalability, i.e., how effectively the proposed approach scales up to big datasets. GNN-BiLSTM models outperform BiLSTM and GNN models, according to the results. By traversing input data from SN twice, BiLSTM appears to be superior at collecting the underlying context (from left to right and then from right to left). The higher performance of GNN-BiLSTM over BiLSTM and GNNs for sorts of data, such as predicting the best product given input ideas, appears plausible. We ran the produced recommendation program on our data and gathered the loss values when the recommendation model received the next batch of data to compare the MAE and RMSE values for the GNN-BiLSTM, GNN, and BiLSTM models. Fig. 7 depicts the RMSE plots for the three proposed approaches, where the y-axis indicates the percentage of error in prediction values and the x-axis represents iteration values, respectively. The RMSE begins at 3.78 and decreases after retrieving six hundred rounds of data for the best results with lower RMSE in approach GNN-BiLSTM, where the loss value reaches 0.78, as shown in Table 3. It implies that the loss value remains constant after one thousand iterations, reaching 0.9 in the last iteration. The MAE value for the GNN-BiLSTM approach, on the other hand, begins at 0.44 and decreases to 0.28 after six hundred iterations, as shown in Fig. 8. Unlike the BiLSTM and GNN models, however, the models never attain the loss value of the equivalent GNN-BLSTM model after obtaining and learning all batches of data (1.22, 1.17). This observation may suggest that the GNN-BiLSTM model requires more training data than BiLSTM and GNN to achieve equilibrium. It further claims that the GNN-BiLSTM-based learning model fine-tuned using fresh data batches.

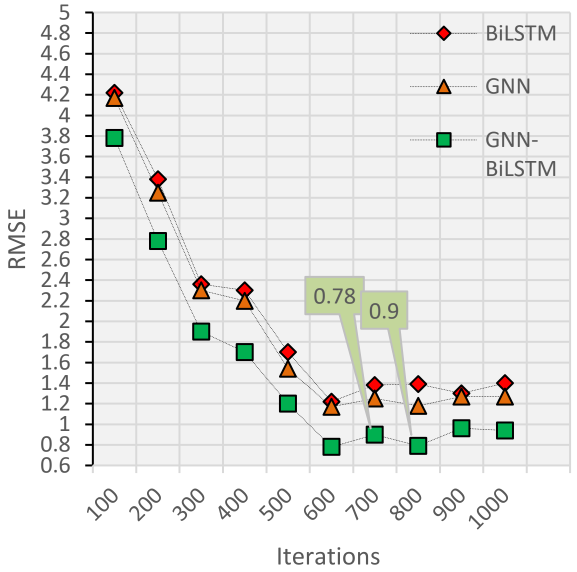


Figure. 7 RMSE measurements

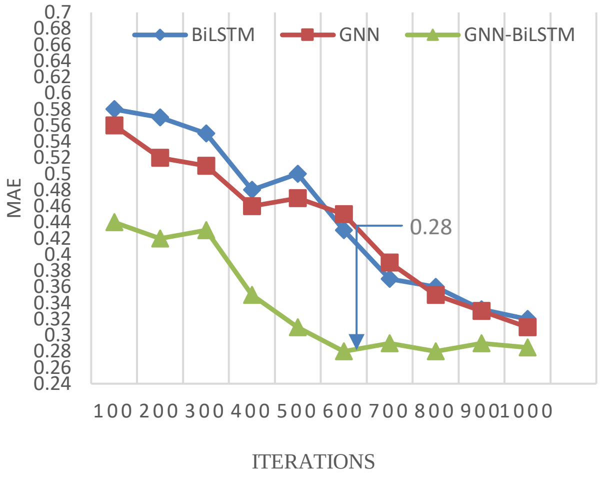


Figure. 8 MAE measurements

* MSE and RMSE are measure of the average magnitude of the errors between the actual and predicted values, and it represents the typical deviation of the predicted values from the true values, in the same units as the dependent variable

Table 3. Error correction rate

Recommender model	RMSE(m3/s)	MAE(m3/s)
SMIN	0.854	0.621
SNHF	0.732	0.543
SR-HGNN	0.646	0.478
KCGN	0.812	0.596
BiLSTM	0.925	0.675
GNN	0.687	0.511
GNN-BiLSTM	0.620	0.461

7. Conclusion

Because of neuroevolution dominance in learning on traditional neural networks, the application of feature extraction techniques in recommender systems is gaining popularity in academia and business. This article presents GNN-BiLSTM, a hybrid recommender system that combines social network filtering and users' history actions depending on the proposal. It focuses on providing a comprehensive study of the most recent works on feature extraction-based recommender systems. We develop two components that use different approaches to automatically extract semantic representations from the knowledge base's structural, textual, and visual information. Following that, we combine collaborative filtering and knowledge base embedding components into a single framework and learn multiple representations jointly. Our extensive research demonstrated the effectiveness of our GNN-BiLSTM solution. Furthermore, this research sheds new light on the usage of heterogeneous information in the knowledge base, which may use in a larger range of applications. Finally, the GNN-BiLSTM hybrid recommender system described in this paper provides a solid platform for future research and development in the field of tailored recommendations. Researchers and practitioners may improve the system's capabilities and its beneficial influence on diverse application fields by tackling these prospects. While the GNN-BiLSTM hybrid recommender system performed well in this study, there are various areas for further research and implications for its use, such as dealing with cold-start issues using adaptive genetic algorithm (AGA), in which the system lacks adequate data on new users or objects.

Funding statement

The authors received no specific funding for this study.

Availability of data and materials

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of interest

The authors declare no conflict of interest.

Author contributions

Conceptualization, Marwan Kadhim Mohammed Al-Shammari, Elaf Ayyed Jebur, Halah Hasan Mahmoud; methodology, Elaf Ayyed Jebur; software and validation, Halah Hasan Mahmoud; formal analysis, Marwan Kadhim Mohammed Al-Shammari; investigation, Israa Ibraheem Al_Barazanchi; resources and data curation, Elaf Ayyed Jebur, and Halah Hasan Mahmoud; writing-original draft preparation, Halah Hasan Mahmoud; writing-review and editing, Marwan Kadhim Mohammed Al-Shammari; visualization, Israa Ibraheem Al_Barazanchi, and Elaf Ayyed Jebur; project administration, Marwan Kadhim Mohammed Al-Shammari; funding acquisition and proofreading, Ravi Sekhar, Pritesh Shah.

References

[1] S. Bergamaschi, F. Guerra, and B. Leiba, "Guest editors' introduction: information overload", *IEEE Internet Computing*, Vol. 14, No. 6, pp. 10-13, 2010.

[2] W. Fan, X. Liu, W. Jin, X. Zhao, J. Tang, and Q. Li, "Graph trend filtering networks for recommendation", In: *Proc. of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 112-121, 2022.

[3] J. Lu, D. Wu, M. Mao, W. Wang, and G. Zhang, "Recommender system application developments: a survey", *Decision Support Systems*, Vol. 74, pp. 12-32, 2015.

[4] S. Wu, F. Sun, W. Zhang, X. Xie, and B. Cui, "Graph neural networks in recommender systems: a survey", *ACM Computing Surveys*, Vol. 55, No. 5, pp. 1-37, 2022.

[5] A. Kumar, S. S. Singh, K. Singh, and B. Biswas, "Link prediction techniques, applications, and performance: A survey", *Physica A: Statistical Mechanics and its Applications*, Vol. 553, p. 124289, 2020.

[6] D. Kagan, Y. Elovichi, and M. Fire, "Generic anomalous vertices detection utilizing a link prediction algorithm", *Social Network Analysis and Mining*, Vol. 8, No. 1, pp. 1–13, 2018.

[7] H. Yao and Y. Lu, "Analyzing the potential influence of Shanghai stock market based on link prediction method", *Journal of Systems Science and Information*, Vol. 5, No. 5, pp. 446–461, 2017.

[8] N. Ma, Y. Liu, and L. Li, "Link prediction in supernetwork: Risk perception of emergencies", *Journal of Information Science*, Vol. 48, No. 3, pp. 374–392, 2022.

[9] I. Esslimani, A. Brun, and A. Boyer, "Densifying a behavioral recommender system by social networks link prediction methods", *Social Network Analysis and Mining*, Vol. 1, pp.

159-172, 2011.

[10] S. Deng, D. Wang, X. Li, and G. Xu, "Exploring user emotion in microblogs for music recommendation", *Expert Systems with Applications*, Vol. 42, No. 23, pp. 9284-9293, 2015.

[11] F. Alqadah, C. K. Reddy, J. Hu, and H. F. Alqadah, "Biclustering neighborhood-based collaborative filtering method for top-n recommender systems", *Knowledge and Information Systems*, Vol. 44, No. 2, pp. 475-491, 2015.

[12] Y. Koren, "Factor in the neighbors: Scalable and accurate collaborative filtering", *ACM Transactions on Knowledge Discovery from Data*, Vol. 4, No. 1, pp. 1-24, 2010.

[13] S. Deng, L. Huang, and G. Xu, "Social network-based service recommendation with trust enhancement", *Expert Syst. Appl.*, Vol. 41, No. 18, pp. 8075–8084, 2014. (2010).

[14] A. Javari and M. Jalili, "A probabilistic model to resolve diversity–accuracy challenge of recommendation systems", *Knowledge and Information Systems*, Vol. 44, No. 3, pp. 609-627, 2015.

[15] S. Han and L. Xiao, "An improved adaptive genetic algorithm", In: *Proc. of SHS Web of Conferences*, Vol. 140, p. 01044, 2022.

[16] M. Ismail, K. B. Prakash, and M. N. Rao, "Collaborative filtering-based recommendation of online social voting", *International Journal of Engineering and Technology*, Vol. 7, No. 3, pp. 1504-1507, 2018.

[17] S. R. S. Reddy, S. Nalluri, S. Kunisetti, S. Ashok, and B. Venkatesh, "Content-based movie recommendation system using genre correlation", In: *Smart Intelligent Computing and Applications: Proc. of the Second International Conference on SCI 2018*, Vol. 2, pp. 391-397, 2019.

[18] G. Geetha, M. Safa, C. Fancy, and D. Saranya, "A hybrid approach using collaborative filtering and content-based filtering for recommender system", In: *Journal of Physics: Conference Series*, Vol. 1000, No. 1, p. 012101, 2018.

[19] X. Li, M. Wang, and T. P. Liang, "A multi-theoretical kernel-based approach to social network-based recommendation", *Decision Support Systems*, Vol. 65, pp. 95-104, 2014.

[20] S. Deng, L. Huang, G. Xu, X. Wu, and Z. Wu, "On deep learning for trust-aware recommendations in social networks", *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 28, No. 5, pp. 1164-1177, 2016.

[21] Z. Huang, Y. Liu, C. Zhan, C. Lin, W. Cai, and Y. Chen, "A novel group recommendation model with two-stage deep learning", *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, Vol. 52, No. 9, pp. 5853-5864, 2021.

[22] Z. Qu, B. Li, X. Wang, S. Yin, and S. Zheng, "An efficient recommendation framework on social media platforms based on deep learning", In: *Proc. of 2018 IEEE International Conference on Big Data and Smart Computing*, pp. 599-602, 2018.

[23] A. Salamat, X. Luo, and A. Jafari, "HeteroGraphRec: A heterogeneous graph-based neural networks for social recommendations", *Knowledge-Based Systems*, Vol. 217, p. 106817, 2021.

[24] L. Guo, Y. F. Wen, and X. H. Wang, "Exploiting pre-trained network embeddings for recommendations in social networks", *Journal of Computer Science and Technology*, Vol. 33, No. 4, pp. 682-696, 2018.

[25] Z. Zhao, "Social-aware movie recommendation via multimodal network learning", *IEEE Transactions on Multimedia*, Vol. 20, No. 2, pp. 430-440, 2017.

[26] D. Datta, T. M. Navamani, and R. Deshmukh, "Products and movie recommendation system for social networking sites", *International Journal of Scientific and Technology Research*, Vol. 9, No. 10, pp. 262–270, 2020.

[27] L. Cui, W. Huang, Q. Yan, F. R. Yu, Z. Wen, and N. Lu, "A novel context-aware recommendation algorithm with two-level SVD in social networks", *Future Generation Computer Systems*, Vol. 86, pp. 1459-1470, 2018.

[28] O. Tal and Y. Liu, "A joint deep recommendation framework for location-based social networks", *Complexity*, Vol. 2019, 2019.

[29] S. S. Namini, N. Tavakoli, and A. S. Namin, "The performance of LSTM and BiLSTM in forecasting time series", In: *Proc. of 2019 IEEE International Conference on Big Data*, pp. 3285-3292, 2019.

[30] S. Wu, F. Sun, W. Zhang, X. Xie, and B. Cui, "Graph neural networks in recommender systems: a survey", *ACM Computing Surveys*, Vol. 55, No. 5, pp. 1-37, 2022.

[31] S. Anand, "Amazon Product Reviews", *Electronic Products User Ratings*, https://www.kaggle.com/datasets/saurav9786/amazon-product-reviews?select=ratings_Electronics+%281%29.csv.