Knowledge Graph Enhanced Contextualized Attention-Based Network for Responsible User-Specific Recommendation

EHSAN ELAHI, Ghulam Ishaq Khan Institute of Engineering Sciences and Technology, Topi, Pakistan
SAJID ANWAR, Institute of Management Sciences, Peshawar, Pakistan
BABAR SHAH, Zayed University, Abu Dhabi, United Arab Emirates
ZAHID HALIM, Ghulam Ishaq Khan Institute of Engineering Sciences and Technology, Topi, Pakistan
ABRAR ULLAH, Heriot University, Edinburgh, United Kingdom
IMAD RIDA, University of Technology of Compiègne, Compiegne, France
MUHAMMAD WAQAS, Edith Cowan University, Joondalup, Australia and University of Bahrain, Sakheer, Bahrain

With ever-increasing dataset size and data storage capacity, there is a strong need to build systems that can effectively utilize these vast datasets to extract valuable information. Large datasets often exhibit sparsity and pose cold start problems, necessitating the development of responsible recommender systems. Knowledge graphs have utility in responsibly representing information related to recommendation scenarios. However, many studies overlook explicitly encoding contextual information, which is crucial for reducing the bias of multi-layer propagation. Additionally, existing methods stack multiple layers to encode high-order neighbor information while disregarding the relational information between items and entities. This oversight hampers their ability to capture the collaborative signal latent in user-item interactions. This is particularly important in health informatics, where knowledge graphs consist of various entities connected to items through different relations. Ignoring the relational information renders them insufficient for modeling user preferences. This work presents an end-to-end recommendation framework named *KGCAN* (Knowledge Graph Enhanced Contextualized Attention-Based Network), which explicitly encodes both relational and contextual information of entities to preserve the original entity information. Furthermore, a user-specific attention mechanism is employed to capture personalized recommendations. The proposed model is validated on three benchmark datasets through extensive experiments. The experimental results demonstrate that KGCAN outperforms existing knowledge graph based recommendation models. Additionally, a case study from the healthcare domain is discussed, highlighting the importance of attention mechanisms and high-order connectivity in the responsible recommendation system for health informatics.

This work was supported by the GIK Institute graduate program research fund under the GA-F scheme.
Authors' addresses: E. Elahi and Z. Halim (Corresponding author), Ghulam Ishaq Khan Institute of Engineering Sciences and Technology, Topi, 23460, Pakistan; e-mails: ehsanelahi@giki.edu.pk, zahid.halim@giki.edu.pk; S. Anwar, Institute of Management Sciences, Peshawar, 25000, Pakistan; e-mail: sajid.anwar@imsciences.edu.pk; B. Shah, Zayed University, Abu Dhabi, 144534, United Arab Emirates; e-mail: babar.shah@zu.ac.ae; A. Ullah, Heriot-Watt University Dubai Campus, 38103, United Arab Emirates; e-mail: a.ullah@hw.ac.uk; I. Rida, University of Technology of Compiégne, Compiégne, 60203, France; e-mail: imad.rida@utc.fr; M. Waqas, School of Engineering, Edith Cowan University, Joondalup, WA, 6027, Australia, and College of Information Technology, University of Bahrain, Sakheer, 32038, Bahrain; e-mail: engr.waqas2079@gmail.com.

ACM 2157-6904/2024/7-ART83
https://doi.org/10.1145/3641288

CCS Concepts: • **Information systems** → **Extraction, transformation and loading**;

Additional Key Words and Phrases: Responsible recommendation system, graph neural network, knowledge graph, attention mechanism

ACM Reference format:
Ehsan Elahi, Sajid Anwar, Babar Shah, Zahid Halim, Abrar Ullah, Imad Rida, and Muhammad Waqas. 2024. Knowledge Graph Enhanced Contextualized Attention-Based Network for Responsible User-Specific Recommendation. *ACM Trans. Intell. Syst. Technol.* 15, 4, Article 83 (2024), 24 pages. https://doi.org/10.1145/3641288

1 INTRODUCTION

The usage of recommendation systems is increasing with time, as almost every online platform that deals with a large customer base is equipped with one. Examples include e-commerce websites, social media platforms, and news portals, to name a few. The purpose of deploying a recommendation system is to display content that aligns with the users' interests. In this way, e-commerce websites expand their businesses by engaging users with personalized recommendations, as the users do not know what they actually need from the large list of products the system is displaying. Additionally, there are new users who may be unaware of the top trending items in the market, leveraging the recommendation system to allow users to choose from top-selling items. That is why recommendation systems are being deployed in e-commerce websites to capture the changing needs of their customers. Different users may have different needs, and therefore it will not be appropriate to display the same set of items to each user. Furthermore, the engagement of users with the items may not be attained, which is the key concern in any e-commerce website. Consequently, it will lead to poor user experience. Over time, the recommendation system requires modification so that changing needs of the users and businesses are fulfilled. In recent years, researchers have been constantly trying to design and model the recommendation system to fulfill this necessity.

Many recommendation strategies have been proposed in the literature, and among them, **Collaborative Filtering (CF)** [1] is one such approach. CF assumes that users with similar histories have similar interests in items. **Matrix Factorization (MF)** [2], which is based on CF, assumes that there are latent relationships between user-item interactions. Although widely used in the past, MF suffers from the data sparsity problem since the only source of information is user-item interaction [3]. Thus, the lack of side information and other responsibility aspects like diversity, novelty, and serendipity make MF less applicable in real-world scenarios.

Although traditional recommendation systems such as MF have been utilized and remain successful, they have their own limitations to the cold start problem. In the cold start problem, the system finds it difficult to recommend some item (a book, some music, news, etc.) for the new user in the system. Additionally, they suffer from a sparsity issue in which user-item interaction is inadequate to provide some quality recommendation. In this digital era when the users have changing demands and to meet this challenge, there is a need to have some sophisticated, innovative strategy that leverages the rich semantic side information as well.

In recent years, deep learning based approaches have been proposed in the recommendation domain, which transform traditional CF approaches into a neural network format. **Neural Collaborative Filtering (NCF)** [4] is one such approach that has two key components. First, the embedding component transforms the user and item into vector representations. Second, the interaction modeling component uses these vectorized representations to reconstruct the user-item interaction. For interaction modeling, a translation-based method [5] has also been proposed, which replaces the inner product with Euclidean distance. However, these approaches do not yield satisfactory embeddings of users and items. The possible reason for this is the utilization of descriptive attributes only, such as ID, for the embedding component. This causes the model to overfit, as the interaction function encodes the complex relationship through a deep neural network. Thus, it exacerbates the data sparsity problem. To alleviate this issue, side information has been proposed to be incorporated into the recommendation scenario. For this purpose, **Knowledge Graphs (KGs)** have been utilized and gained much attention from researchers [6, 7]. The KG provides the structured and semantic enrich representation of real-world knowledge. It consists of different entities which are connected among each other with some meaningful relations that give more power to them. These semantic relations in the KG provide the ability of reasoning and explaining of various recommendation-related tasks. In the recommendation scenario, KGs are widely being used in recent literature owing to their ability to provide rich information and to overcome data sparsity as well as the cold start problem. The KG is an integral part of our model, as it enables us to express organized, semantic relationships between items and entities. It also allows for the integration of domain-specific knowledge such as item-item relationships and user-item interactions, which improves recommendation quality. Existing KG-based methods fall into three categories: path-based, embedding-based, and propagation-based methods.

In path-based methods, predefined meta-paths are used to capture the semantic relatedness among users and items. These methods heavily rely on hand-crafted paths and thus fail to identify unseen connectivity paths [8, 9]. Embedding-based methods apply KG embedding techniques to transfer the KG into vectorized representation while preserving the KG structure [10]. These methods capture the KG relation implicitly but are not suitable for explicitly encoding the semantic relation of the KG into the recommendation. Propagation-based methods capture higher-order information by stacking multiple layers. The problem with propagation-based methods is that they do not consider the semantic relational information along links in the KG, thus allowing the introduction of noisy entities in the aggregation. Moreover, the uniqueness of each node may disappear due to the oversmoothing problem introduced in the propagation [11].

In general, an entity in the KG is connected to items and users, forming a heterogeneous graph with diverse relations and node types. Each entity has first-order neighbors directly attached to it, as well as higher-order neighbors attached to the neighbors of neighbors. In recent literature, **Graph Neural Network (GNN)**-based recommendation models [12, 13] have been proposed to encode the semantic information of both first-order and higher-order neighbors. Nevertheless, they may have some shortcomings that prevent them from effectively addressing the given challenges.

Challenge 1. User-specific preferences are ignored in most GNN-based recommendation methods [14, 15]. In other words, the first-order neighbors of an entity are aggregated without considering the user-specific preferences. For example, in Figure 1, both u_1 and u_2 have interacted with item i_2, but the reasons for their interactions with i_2 may vary. It is possible that u_1 interacted with i_2 due to entity e_1, whereas u_2 interacted with i_2 due to the attribute entity e_3. These user-specific preferences are disregarded in most recommendation methods, which makes them insufficient for encoding the personalized preferences of a given user.

Challenge 2. For the given entity, its higher-order neighbor information is useful to enrich its representation as well as to reason about the possible preference for recommendation. Most existing methods try to encode this high-order information by selecting relevant paths, which is laborious, and domain knowledge is required to select the relevant paths [8, 9]. Another approach is propagation-based methods, which stack multiple layers to encode such information, but the problem with them is that they do not take relation into consideration, as a KG has entities which are connected to each other with different types of relations. So, ignoring relational information makes them insufficient to model the preferences of user. Relational information is necessary to

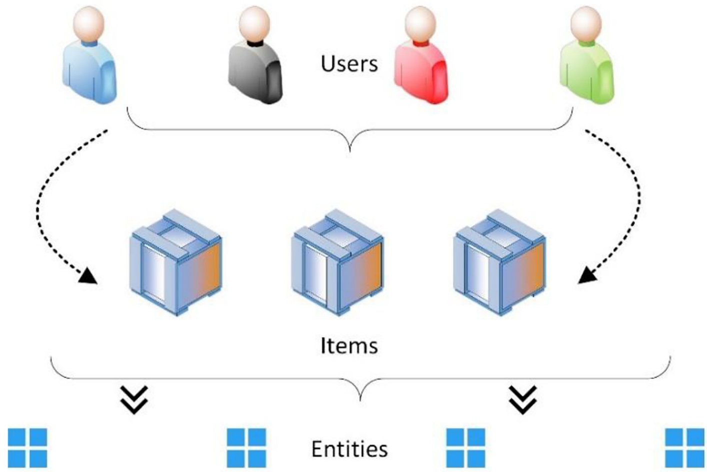

Fig. 1. A toy example of user-item interaction and a knowledge graph.

capture the semantics and context of entities in a KG. The model may struggle to understand the complex details of relationships between items and entities if this information is not given. As such, recommendations may be inadequate and irrelevant. Furthermore, user-specific preferences and behaviors are often captured by relationships between items and entities. Ignoring these relationships can result in generalized recommendations that do not take individual user interests into account. As a result, personalization loses its effectiveness. The lack of considering relational information may result in homogeneous recommendations, limiting user engagement and satisfaction. The model may not appropriately recommend novel or less-explored items if relational information is not considered. Users might get recommendations that have a significant bias toward popular choices, inhibiting their exposure to novel stuff.

To address these aforementioned challenges, we propose the recommendation model KGCAN, which leverages both the first-order and higher-order graph context while paying attention to user-specific preferences for a given entity. Moreover, KGCAN utilizes context-aware attentive knowledge propagation, which not only assigns more weight to relevant entities but also ensures that bias is not introduced due to the propagational layers, thereby preserving the original representation of an entity. The main contributions of this work can be summarized as follows:

- A responsible recommendation model KGCAN is proposed that explicitly exploits interaction information as well as a KG for side information in an end-to-end fashion.
- We emphasize the significance of contextualized representations for users and items to preserve the original information and mitigate bias caused by multilayer propagation.
- We propose a relational user-specific attention mechanism that generates personalized recommendations for each user and assigns varying importance to their neighbors based on the connection relation.
- Extensive experiments are conducted to validate the proposed model on three benchmark datasets. The experimental results demonstrate that our model, KGCAN, outperforms state-of-the-art baselines.

Our proposed approach in this article is a responsible recommender system that focuses on diversity, novelty, and serendipity. In the realm of recommendation systems, it is essential to consider user-specific preferences and the contextualized representations of entities in a KG.

Existing GNN-based recommendation models have limitations in addressing these challenges. First, they often overlook user-specific preferences by aggregating the first-order neighbors of an entity without taking into account individual user variations. This results in insufficient encoding of personalized preferences. Second, whereas higher-order neighbor information can enrich entity representations and provide valuable insights for recommendations, existing methods struggle with selecting relevant paths or fail to consider the relations within the KG. To overcome these challenges, we introduce the recommendation model KGCAN. KGCAN leverages both first-order and higher-order graph context, paying attention to user-specific preferences. It employs context-aware attentive knowledge propagation to assign appropriate weights to relevant entities while preserving the original entity representation. Our contributions include proposing KGCAN as a responsible recommendation model that integrates interaction information and KG side information. We emphasize the significance of contextualized representations to reduce bias caused by multilayer propagation. The contextualized representation is obtained by concatenating the learned representation of the user/item with their original representations. By doing so, the oversmoothing problem is reduced and the uniqueness of each entity is preserved. Moreover, we introduce a relational user-specific attention mechanism that generates personalized recommendations by considering the connection relations. An attention mechanism enables the model to dynamically weigh the significance of various entities based on user behavior. Moreover, it has the capability to encode personalized recommendations by assigning different importance to different entities in the recommendation scenario. For each user, a user-specific attention mechanism dynamically adjusts the focus on various parts of the given input data in the neural network. This assigning of weights depends on the users' preferences and historical interactions of the user. By integrating the user-specific attention mechanism, the system is allowed to give personalized recommendations by adjusting to the user's preferences for each user. Given a scenario of a movie recommendation system, a neural network is used in this system to predict whether a user would like a certain movie based on user and movie features. When identifying what aspects to focus on for each user, the user-specific attention mechanism comes into play.

Without User-Specific Attention. A traditional model would have the neural network consider all user and movie features equally. This implies that, regardless of individual preferences, the neural network will look at the same aspects of movies for all users.

With User-Specific Attention. The neural network's emphasis can be dynamically adjusted due to the user-specific attention mechanism. For instance, if user A prefers action movies, the attention mechanism may prioritize features associated with action genres when user A interacts with the system. However, if user B likes romantic comedies, the attention mechanism may focus on characteristics related to that genre. Extensive experiments on three benchmark datasets demonstrate that KGCAN outperforms state-of-the-art baselines, validating its effectiveness.

The rest of the article is organized as follows. Section 2 provides a description of recent and relevant works that utilize the KG for recommendation purposes. Section 3 presents the task formulation of the research problem to be addressed, whereas Section 4 illustrates the framework and methodology of the proposed recommendation model. In Section 5, the experimental results are presented, along with their discussion. Finally, we conclude this work and provide insights into future directions for further study.

2 RELATED WORK

This section reviews recent work done in the recommendations system domain using the GNN and the KG.

2.1 GNN-Based Recommendation

Recently, the **Convolutional Neural Network (CNN)** has been extended to the GNN [16], to model the graph structure data. Since graph data is of irregular structure having no fixed number of nodes in each graph, it may not be appropriate to apply a CNN on the graph data, having no fixed size matrices. The operation of a **Graph Convolutional Network (GCN)** [17] is to aggregate the neighboring nodes' information, so as to enrich the representation of each target node. Each node in the graph has neighbors from which it iteratively aggregate information, thus allowing the capture of local neighborhood information (one hop away) as well as higher-order neighbor information (more than one hop away). GCN methods are broadly classified into two types: the spectral methods and the spatial methods.

In spectral methods, first the graph's features are transferred into the Fourier domain, then a convolution operation is applied in the Fourier domain. In other words, a convolution operation is not applied directly on the graph. For example, Bruna et al. [18] have utilized the spectral GCN where the eigen decomposition is carried out in Fourier domain. Chebyshev polynomials are also utilized to approximate the convolution operation on the eigenvalue's matrix [19], in an attempt to reduce computational complexity. This study falls in the category of spatial methods, so more attention is given to spatial methods.

In spatial methods, the convolution operation is applied directly on the graph structure, which essentially means that information is propagated to directly attached nodes of a graph. The first and basic spatial method is proposed [20], where the information of neighboring nodes is summed up to enrich the target node's representation. Then, to preserve the information from the previous layer, a residual connection is utilized at each layer. The number of neighboring nodes may vary for each node, so to tackle this, a sampling approach is proposed in the literature. Sampling approaches sample the fixed number of neighboring nodes and then an aggregator is applied to aggregate the information from these neighbors [21]. One shortcoming of these methods is that they are designed for homogeneous graphs where only the user-item information is utilized and encoded.

Veličković et al. [22] introduce the concept of the attention mechanism GAT, arguing that instead of treating each neighboring node equally during aggregation, it is reasonable to weight each node. The benefit of the attention mechanism is that it can deal with the variable input size of nodes. Hence, more focus is given to the relevant neighboring nodes, thus increasing the performance of the model. Sequential-based recommenders such as RetaGNN [13] are also proposed in the literature, which, for each user, recommend the next item based on the user's previous interacted items. LightGCN [23] is an approach in which an attempt is made to make the GCN a light model. The authors argue that feature transformation as well as nonlinearity (activation function) make the model complex and hence should be avoided, as in some cases performance may suffer from using this. IMP-GCN [24] is proposed, which tries to reduce the oversmoothing problem that is inherent in the higher-order information aggregation of neighboring nodes. The problem with these models is that they do not consider the importance of the attention mechanism and thus treat every neighbor equally, which may not be rational in a real-world scenario.

2.2 KG-Based Recommendation

In the literature, the KG is being widely adopted in the recommendation scenario. KG-based recommendation work is divided into three main categories: embedding-based methods, path-based methods, propagation-based methods. Details for each are presented next.

Embedding-Based Methods. In existing embedding-based methods, entities in the KG are embedded to vectorized form, whereas the structural information of KG is preserved. This embedded form is used to regularize learning of latent vector representation of the user and the item. The user-item interactions are leveraged to optimize the loss function of the recommender system. In

CKE [10] and DKN [25], semantic embedding of the nodes of the KG is generated, through KG embedding (KGE) approaches. Afterward, these generated embeddings are fed into the recommender system to regularize the user and item representation learning. The authors of RippleNet [6] and KGCN [26] utilize the GNN to embed the item in a KG. These item embeddings are used to encode the item relationships with neighboring entities on the KG, thus capturing the collaborative signal for recommendation. KGNN-LS [27] extends KGCN by emphasizing the label smoothness to ensure regularization over the edges of neighboring entities.

Path-Based Methods. In path-based methods, paths and meta-paths are designed to infer preferences of the user. More formally, the representation of each entity in a KG is enriched with higher-order connectivity information, by designing paths from the start entity to the L-hops away entity. Since there can be many paths from a given entity to other entities, two approaches are noted. The first approach is the selective approach [8], in which most significant paths are selected and thus used to enrich the entity representation. The second approach is based on meta-path patterns to provide a limit of paths [9]. Yu et al. [28] proposed an attribute-rich model known as the heterogeneous information network (HIN) for improving recommendation quality. Dong et al. [29] employed a meta-path-based random walk component, which encodes the heterogeneous neighborhood information for a node. CGAT [30] introduced the concept of biased random walk, which employs the gated recurrent unit to encode the higher-order connectivity information of a given node. For an entity, multiple paths of fixed length are explored by repeating the biased random walk strategy, thus ensuring the wider search. One major drawback of using path-based methods is that they are labor intensive and much domain knowledge is required to extract the significant paths for a given entity. In real-world scenarios, where the KG size may reach up to a million entities, this makes the situation impossible to efficiently design paths.

Propagation-Based Methods. As the name suggests, these methods operate on a KG where information is propagated iteratively to provide auxiliary information for recommendation. These methods have attained much attention in the research community. For example, the RippleNet [6] model enriches the potential preferences of the user by propagating along KG links. However, the RippleNet model has not considered the importance or relevance of KG links in which information propagation is being done. KGAT [15] is another propagation-based method that uses a **Collaborative Knowledge Graph (CKG)** to integrate the user-item interaction and the KG. In KGAT, the user and entities are treated in the same manner, which may not be rational because users have different meanings in a user-item bipartite graph, whereas entities in a KG represent different meaning. Moreover, the KGAT model needs to retrain itself for the new upcoming user so that recommendation can be provided to her. This makes KGAT highly computationally expensive in the real world where there are a million entities. CKAN [14] presents the heterogeneous propagation in which collaboration propagation and KG propagation are integrated. This end-to-end framework leverages attentive embedding to learn varying weights of neighboring nodes, for each node. As the KGs have diverse relations, which is essential to incorporate to have personalized recommendation, the attention mechanism is tailored to incorporate this diverse information. However, it does not encode the contextual information that is essential to reduce the bias of high-order propagation as well as for better-quality recommendations.

Like CKAN, MKGAT [39] treats the UII and KG information as a unified graph in such a way that users and items are the same type of entities, whereas the relation between the two is treated as an interaction. In recent years, more focus is being given to the construction of a subgraph—that is, only those entities and relations are considered in the subgraph which are relevant to the user-item pair. Sha et al. [40] work on the assumption that shorter distance between the two nodes represents a strong and reliable pair. Therefore, it utilizes TransR [41] to train the entities' embedding and then compute the Euclidean distance among linked entities while keeping the K number of shortest paths from the target user-item node. Considering that these methods rely on subgraph construction to have a relevant user and item pair, one main drawback of these methods is that they are time consuming. Therefore, further scientific research is needed to explore the efficient techniques of subgraph construction.

3 TASK FORMULATION

Before we introduce the proposed framework, we first demonstrate the KG-based recommendation system. There are two types of domain information being used in a KG-based recommendation system: user-item interaction and item KG information.

3.1 User-Item Interaction Information

In a typical recommendation system, there is a set of users U as well as a set of items I. The user-item interaction can be represented as a bipartite graph where the link denotes that the user has interacted with the item (e.g., user u has purchased an item i or viewed an item i). Thus, the user-item interaction matrix $Y \in \mathbb{R}^{M \times N}$ is constructed (where M and N are the number of users and items, respectively) in which each entry y_{ui} is either 0 or 1, where

$$y_{ui} = \begin{cases} 1 & \text{if interaction } (u, i) \text{ is observed} \\ 0 & \text{otherwise} \end{cases}$$

Here, $y_{ui} = 0$ does not mean that user disliked the item; rather, it may mean that user u is unaware of that item or may be ignored by the user due to many items being displayed.

3.2 Item KG Information

Item KG is a directed graph with relation information to have auxiliary information, and thus it is utilized to alleviate the sparsity problem of the user-item interaction matrix. It is in the form of triples (*h, r, t*), which represents that there exists relation r from head entity h to tail entity t. For example, in the triple (Christopher, film.film.actor, Batsman Begins), the relation is actor and Christopher is the head entity, whereas Batsman Begins is the tail entity. This triple is described as Christopher being the film actor of the movie *Batsman Begins*. One thing to highlight here is that the movie is an item for which semantic rich information is obtained by utilizing a KG into the context.

Once we have user-item interaction matrix Y as well as item KG G, the recommendation task is to predict the probability that given user u would be interested in item i with which he/she has not interacted (unobserved interaction). The probability function is $\hat{y}_{ui} = \mathcal{F}(u, iY,\ G,\ \theta)$, where θ denotes all of the model's parameters of probability function $\mathcal{F}$. Table 1 shows all notations used in this article.

4 PROPOSED FRAMEWORK

In this work, a KG-based recommendation framework (i.e., KGCAN) is presented that assists in the selection of relevant information to have user-specific recommendations. The overall framework of the proposed recommendation model is shown in Figure 2. As depicted in the figure, there are three main components of the framework: heterogeneous knowledge propagation, contextualized attention-aware embedding, and the prediction layer. The details of these components are elaborated in the following sections.

4.1 Heterogeneous Knowledge Propagation

Table 1. Notations and Their Descriptions

Notation	Description
U	Users' set
I	Items' set
$\mathcal{E}$	Entity set
R	Relations' set
G	Knowledge graph
Z	Alignment set
(h, r, t)	(head, relation, tail) A triplet of knowledge graph
Y	Interaction matrix of users and items
$\hat{y}_{ui}$	Probability score
p_u	User's aggregated representation
q_i	Item's aggregated representation
Θ	Parameters of the model
$\mathcal{E}_u^{l+1}$	User entity set at $l + 1$ propagation layer
$\mathcal{E}_i^{l+1}$	Item entity set at $l + 1$ propagation layer
T_u^{l+1}	User triple set after $l + 1$ propagation layer
T_i^{l+1}	Item triple set after $l + 1$ propagation layer
$\mathcal{L}_{\mathrm{CE}}$	Loss function

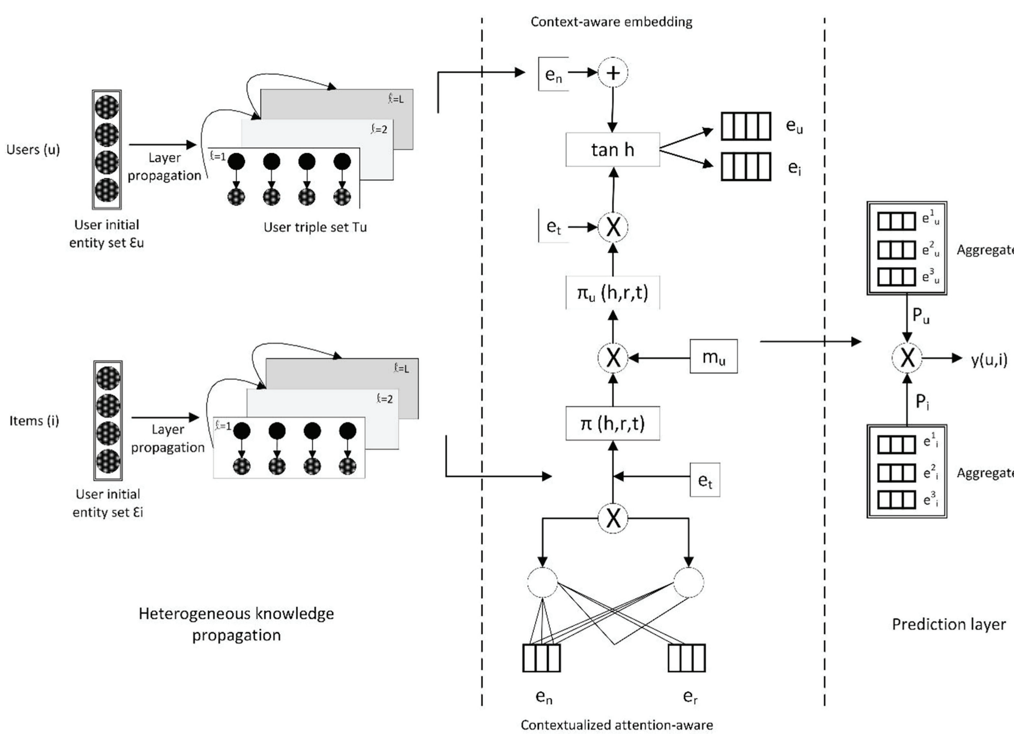

Fig. 2. Overall framework of the proposed recommendation model.

The heterogeneous knowledge propagation layer is heterogeneous, as it is composed of two different modules: collaboration propagation information and KG information. In collaboration propagation, a vital collaborative signal is encoded, which is then used to enrich the representation of both the user and item. In KG information, relevant information is propagated through the links of the KG, thus providing auxiliary side information. The details of these modules are presented next.

ALGORITHM 1: The optimization procedure

Input:
User-item interaction matrix Y,
Knowledge graph G
Output:
Probability score $\hat{y}(u, i)$

Randomly initialize all of the model's parameters
for epoch = 1 to num_epochs **do**
Pick a batch of user triples and item triples from T ;
Perform forward propagation on G;
Compute CTR probability score $\hat{y}(u, i)$;
Compute the gradients for the given batch;
Update models' parameters $\mathbf{\Theta}$ by gradient descent with learning rate η;
end for
return $\hat{y}(u, i)$

Collaboration Propagation Information. CF aims to identify similar users with almost similar item interest. In traditional approaches, users and items are transformed into independent latent vectors, and such vectorized representation is then used to construct the user-item interaction. In this study, we aim to represent the user by the items she has interacted with, as the items interacted by a user u represent her preference and can be used to enrich the representation of that user. For a user u, an initial seed set is constructed from its historical items (interacted set of items), and this initial seed set is then propagated in the KG with the help of alignment set Z. Definition of the initial user u entity set is given as follows:

$$\mathcal{E}_u = \{\mathrm{e} \mid (i, \mathrm{e}) \in Z \text{ and } i \in \{i \mid y_{ui} = 1\}\}, \tag{1}$$

where $\mathcal{E}_u$ represents the initial entity set of user u and Z represents the alignment set that acts as a bridge between item i and entity e.

In the same way, items can also be represented by the set of users who consumed it. More specifically, those users who have interacted with item i will contribute to enrich that item representation. The collaborative item set of items is formed from the users which are basically the collaborative neighbors of item i. The definition of collaborative item set of an item i is given as follows:

$$I_i = \left\{i_u \mid u \in \{u \mid y_{ui} = 1\} \text{ and } y_{ui_u} = 1\right\}. \tag{2}$$

The initial entity set $\mathcal{E}_i$ of item i can be constructed from collaborative item set I_i and alignment set Z. The definition of $\mathcal{E}_i$ is given as follows:

$$\mathcal{E}_i = \{\mathrm{e} \mid (i_u, \mathrm{e}) \in Z \text{ and } i_u \in I_i\}. \tag{3}$$

In the definition of initial item entity set, the entity directly connected to the item is also considered, thus making it sure that original information of the item is preserved and reducing the bias of multihop neighbor information.

KG Information. We build upon the GCN architecture in which feature representation is learned for each entity node in the KG. Unlike traditional KG embedding approaches which consider the directly connected neighbors of an entity, this study aims to find the higher-order neighbors of an entity in a KG with the help of the GCN approach. One layer of the GCN is able to capture only the local neighbor information, so by stacking L-layers, one can able to encode L-hops away neighbor information. In this way, the representation of each entity in enriched, thus capturing more information about the user's preference. The propagation-based methods come with the advantage of no manual feature engineering being required; rather, the feature vector representation of each entity is learned. Moreover, as in the case of path-based methods, there is no need to design paths to explore higher-order neighbors of an entity.

Initial entity sets are extended by the KG propagation, and these extended entity sets of user and item are able to capture the latent relationship between user and item. Formally, the definition for a user and item entity set is given as follows:

$$\mathcal{E}_u^{l+1} = \left\{t \mid (h, r, t) \in \mathcal{G} \text{ and } h \in \mathcal{E}_u^l\right\}, \tag{4}$$

$$\mathcal{E}_i^{l+1} = \left\{t \mid (h, r, t) \in \mathcal{G} \text{ and } h \in \mathcal{E}_i^l\right\}, \tag{5}$$

where l in superscript represents the layer number, which essentially means how far the entity set is from initial entity set, whereas t represents the directly connected neighbor of a given entity h. Once we have an extended entity set obtained from KG propagation, we can formally define the lth triple set for user u and item i.

$$T_u^{l+1} = \left\{(h, r, t) \mid (h, r, t) \in \mathcal{G} \text{ and } h \in \mathcal{E}_u^l\right\}. \tag{6}$$

$$T_i^{l+1} = \left\{(h, r, t) \mid (h, r, t) \in \mathcal{G} \text{ and } h \in \mathcal{E}_i^l\right\}. \tag{7}$$

Using the KG as an additional source of information is reasonable. This is because neighboring entities, as well as their neighbors, enhance the representation of users and items. This enhancement enables the model to capture latent relationships between users and items more effectively.

4.2 Contextualized Attention-Aware Embedding

Consider that an entity h is connected to its neighbor entity t with a relation r, then this fact can be denoted as the triple (h, r, t). Since there can be many different relations with which an entity h can be connected to entity t, there is a need to distinguish the tail or neighboring entities and filter out those irrelevant or noisy entities. For this purpose, an attention-aware mechanism is proposed in this article, with which not only the proximity structure of graph is exploited but also elaborates the importance of different neighboring nodes. The attentive embedding of neighbor entity t is given as follows:

$$e_u = \sum_{T^l} \pi(h, r, t)\mathbf{e}_t, \tag{8}$$

where $\pi(h, r, t)$ measures the attention score for each tail entity. In other words, it is considered as a decay factor that decides how important the neighbor is to the head entity. We have implemented this decay factor as follows:

$$\pi u\,(h, r, t) = (\mathrm{W}_r \mathrm{e}_t)^\top \tanh\left((W_r e_h + e_r)\right) m_u^\top. \tag{9}$$

Here, tanh [31] is the activation function to provide nonlinearity. This relational attention mechanism propagates more information to those entities which are closed to it, as it relies on the distance between $\mathbf{e}_h$ and $\mathbf{e}_t$. In Equation (9), m_u is responsible for calculating different attention scores for each user u, and it is given as follows:

$$m_u = \mathrm{ReLU}\left[p_u W + \mathrm{b}\right]. \tag{10}$$

p_u in Equation (10) represents the users' embedding that we have acquired from the embedding table with user indices for every user. Then, we have utilized the softmax function [32] to normalize the coefficients across all triples in the triple set.

$$\pi(h, r, t) = \frac{\exp\left[\pi_u(h, r, t)\right]}{\sum_{(h, \tilde{r}, \tilde{t}) \in T^l} \exp\left[\pi_u\left(h, \tilde{r}, \tilde{t}\right)\right]} \tag{11}$$

Consequently, each entity is able to decide which neighboring node is more important and relevant for capturing the collaborative signal. Thus, parts of the data are put in focus, which is reasonable in the recommendation scenario. Once we have normalized coefficients of the neighboring entity for the user as well as the item, we calculate the attention score for the tail entity e^t as follows:

$$e_u = \sum_{T^l} \pi(h, r, t)\, \mathrm{e}_j^t, \tag{12}$$

$$e_i = \sum_{T^l} \pi(h, r, t)\, \mathrm{e}_j^t. \tag{13}$$

To have contextualized representation for the entity h, we have aggregated with its neighborhood embedding, given as follows:

$$e_u = \tanh\left[\left(e_h \parallel e_u\right) W + b\right]. \tag{14}$$

In the same way, for the item, the contextualized representation is given as follows:

$$e_i = \tanh\left[\left(e_h \parallel e_i\right) W + b\right]. \tag{15}$$

In the recommendation scenario, embeddings are utilized which represent user and item information. These embeddings are continuous vector representations of users and items which capture their core features. A user embedding, for example, may reflect their previous preferences and behaviors, whereas an item embedding may represent the characteristics and attributes of items. The contextual information is encoded through the concatenation operation that straight away integrates the original user and item embeddings with the embedding generated from the layers of the GNN. In this way, context-aware embedding is generated that does not lack original user and item uniqueness.

4.3 Model Prediction

Multiple representations of the user and item are aggregated from L-layer propagations to form a single vector representing the user and the item. In this study, three different aggregators are utilized, which are discussed in the following. In recent literature, these aggregators are widely used in recommendation system domain.

Pooling Aggregator. In the pooling aggregator, the maximum vector from the representation set is selected, followed by an activation function, to provide nonlinearity:

$$agg^u_{\text{Pooling}} = LeakyReLU\left(\mathrm{W} \cdot \mathrm{pool}_{max}\left(R_u\right) + \mathrm{b}\right),$$

$$agg^i_{\text{Pooling}} = LeakyReLU\left(\mathrm{W} \cdot \mathrm{pool}_{max}\left(R_i\right) + \mathrm{b}\right).$$

Sum Aggregator. This aggregator sum ups the multiple representations to form a single vector. Hereafter, nonlinearity is applied to the aggregated representation:

$$agg^u_{\text{Sum}} = LeakyReLU\left(\mathrm{W} \cdot \sum_{\mathrm{e}_u \in R_u} \mathrm{e}_u + \mathrm{b}\right),$$

$$agg^{i}_{\text{Sum}} = LeakyReLU\left(\mathrm{W} \cdot \sum_{\mathrm{e}_i \in R_i} \mathrm{e}_i + \mathrm{b}\right).$$

Concatenation Aggregator. In this type of aggregator, multiple representations are concatenated to form a single vector, capturing the information of all vectors. Afterward, nonlinearity is applied to the concatenated vector:

$$agg^{u}_{\text{concatenation}} = LeakyReLU\left(\mathrm{W} \cdot \left(\mathrm{e}_u^{(i_1)} \parallel \cdots \parallel \mathrm{e}_u^{(i_n)}\right) + \mathrm{b}\right),$$

$$agg^{i}_{\text{concatenation}} = LeakyReLU\left(\mathrm{W} \cdot \left(\mathrm{e}_i^{(i_1)} \parallel \cdots \parallel \mathrm{e}_i^{(i_n)}\right) + \mathrm{b}\right).$$

Hereafter, having user p_u and item q_i aggregated representations, we calculate the inner product of both representations to get the probability score, which is given as follows:

$$\widehat{y}\,(u, i) = \mathrm{p}_u^{\top}\ \mathrm{q}_i. \tag{16}$$

4.4 Model Training

In our work, we have extracted the same number of negative samples, for each user, as that of positive samples. In this way, we can check the effect of model training, thus the positive and negative sample ratio is balanced. The negative samples are extracted from the unwatched/uninteracted items for each user. We have the following loss function for our proposed model:

$$\mathcal{L}_{\text{CE}} = \sum_{u \in \mathcal{U}} \left(\sum_{(u,\ i) \in I^+} \mathcal{P}\left(y_{(u,\ i)}, \hat{y}_{(u,\ i)}\right) - \sum_{(u,\ j) \in I^-} \mathcal{P}\left(y_{(u,\ j)}, \hat{y}_{(u,\ j)}\right) \right), \tag{17}$$

where I^+ represents the interacted items of the user u, whereas I^- represents the uninteracted items which we have acquired by random negative sampling for each user u, and $\mathcal{P}$ represents the cross-entropy loss. Hence, the following objective function that learns the model's parameters is given as follows:

$$\min_{\Theta}\ \mathcal{L}_{\text{CE}} + \lambda \parallel \Theta \parallel_2^2 .$$

Here, Θ denotes all parameters of the model, whereas $\parallel \Theta \parallel_2^2$ is the L2 regularizer, being parameterized by λ. The optimization of the model as well as calculation of the prediction score are described in Algorithm 1. The architectural diagram of the proposed model is given in Figure 3 for better understanding.

5 EXPERIMENTS

This section describes the experimental results obtained by conducting extensive experiments on three benchmark datasets. These experiments are conducted to answer the following research questions:

RQ1: Does KGCAN perform better in terms of performance when compared with state-of-the-art KG-based recommendation models?

RQ2: How do different variants of KGCAN affect performance of the base model?

RQ3: How effectively does the user-specific component in the attention mechanism capture useful information for the recommendation?

RQ4: How do different hyperparameter settings (e.g., depth of the layer, embedding size, and aggregation function) influence the model's performance?

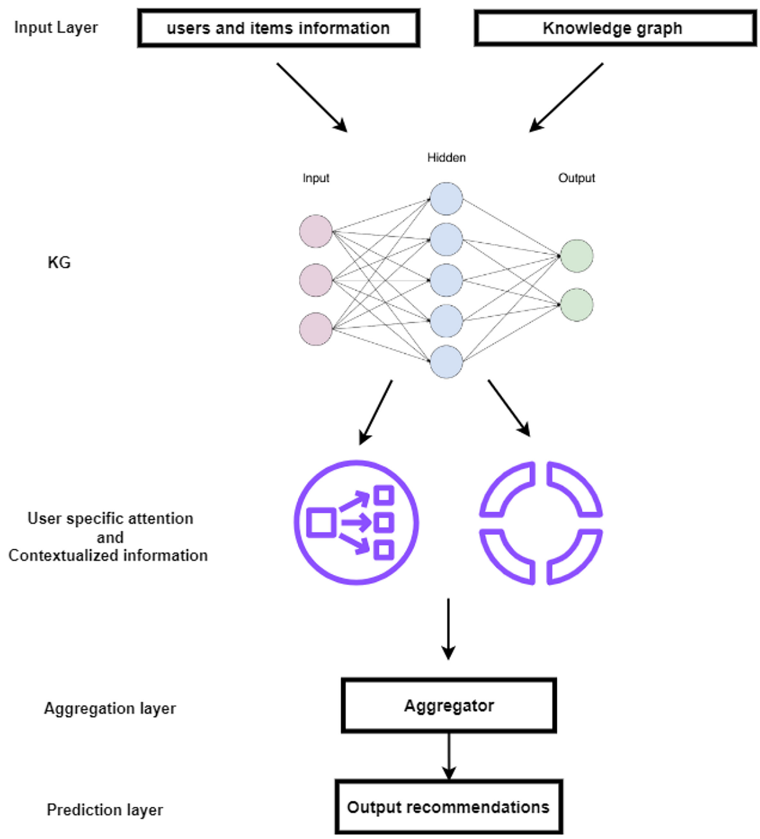

Fig. 3. Architectural diagram of the KGCAN model.

RQ5: Is KGCAN suitable for the healthcare domain?

5.1 Datasets

In this work, three different benchmark datasets from different domains are used to validate the effectiveness of our proposed recommendation model. We ensured diversity in the benchmark datasets to assess the generalizability of our method. These datasets cover three different domains and thus have data characteristics which validate the generalizability of our proposed method. Moreover, our aim is to assess the proposed model against state-of-the-art recommendation models. Therefore, these benchmark datasets are well suited, as they are widely used in state-of-the-art recommendation models, for purposes of comparison. A brief description of these benchmark datasets is given as follows:

- *MovieLens-20M*: The **MovieLens-20M (ML)** dataset consists of almost 20 million ratings, on a scale of 1 to 5, which are given by more than 138,000 users. This is a widely used dataset in movie recommendation scenarios.
- *Last.FM*: The **Last.FM (FM)** dataset consists of around 2,000, which provides information on their music track count. This dataset is provided by the FM online music system.
- *Book-Crossing*: The **Book-Crossing (BC)** dataset consists of more than 17,000 user ratings and is provided by the BC community. The scale of rating is from 0 to 10, which is for different books, treated as items.

Table 2. Datasets and Their Statistics

	ML	FM	BC
No. of Users	138,159	1,872	17,860
No. of Items	16,954	3,846	14,967
No. of Interactions	13,501,622	42,346	139,746
No. of Avg. Interactions	98	23	8
Entities	102,569	9,366	77,903
Relations	32	60	25
Triples	499,474	15,518	151,500

The interactions given in the ML,[1] FM,[2] and BC[3] datasets are in the form of explicit feedback. This explicit feedback is converted to implicit one, where 1 represents positive interaction. To obtain negative interaction for each user, we have randomly sampled uninteracted items from his historical information. To reduce the biasedness, the negative samples are of equal size as that of the positive interaction's size, for each user. In ML, we only considered those ratings as positive where the rating is greater than 4. In the case of FM and BC, no such threshold is considered because of their sparsity. In this work, a KG-based recommendation model is proposed, and the KGs of the used datasets are taken from a public repository (https://github.com/xiangwang1223). For each dataset, from its whole KG, a sub-KG is prepared on the basis of triples in the KG having a confidence level greater than 0.9. Moreover, we have removed the entities and items which are matching other entities. Different statistics for the experimental datasets are summarized in Table 2.

5.2 Baselines

NCF [4]: This method is based on a CF approach, which replaces the inner product of the user and item with a neural network architecture. By the use of neural network architecture, an arbitrary matching function is learned from the data.

RippleNet [6]: This method utilizes KG information to enrich the user's potential preferences along the KG links which are rooted at the user's interacted items.

KGNN-LS [33]: In this model, a GCN is applied on the KG that propagates to compute the item embedding, then aggregation is done to aggregate the neighboring nodes information. Moreover, label smoothness regularization is also applied.

KGCN [26]: This is the extension of the GCN approach where a KG is utilized to aggregate the neighborhood information as well as to encode the item-item relatedness.

KGAT [15]: Unlike KGCN, this model utilizes an attention module that assigns varying weights to neighboring nodes in the CKG composed of a user-item interaction graph and a KG.

CKAN [14]: This is a heterogeneous propagation model that makes use of a relation-aware attention mechanism. This relation-aware attention mechanism considers the relational information of neighboring entities when aggregating them.

5.3 Implementation Details

Each dataset is divided in the proportion of 6:2:2 for training, validation, and testing sets, respectively, as this proportion is widely adopted in the recommendation literature (e.g., [11, 34]). For performance evaluation, **Click-Through Rate (CTR)** prediction is used in which the performance of the recommendation model is evaluated using AUC and F1 scores. The reason for

Table 3. Performance of Different Recommendation Algorithms (w.r.t. AUC and F1 Score)

Dataset	Model	AUC	F1 Score
ML	NCF	0.966	0.918
	Ripple Net	0.969	0.921
	KGNN-LS	0.972	<u>0.926</u>
	KGCN	0.974	0.924
	KGAT	0.973	0.925
	CKAN	<u>0.974</u>	**0.927**
	KGCAN	**0.984**	0.922
FM	NCF	0.759	0.701
	Ripple Net	0.768	0.702
	KGNN-LS	0.801	0.715
	KGCN	0.820	0.704
	KGAT	0.822	<u>0.758</u>
	CKAN	<u>0.840</u>	0.697
	KGCAN	**0.869**	**0.782**
BC	NCF	0.710	0.628
	Ripple Net	0.712	0.631
	KGNN-LS	0.667	0.631
	KGCN	0.706	0.632
	KGAT	0.721	0.651
	CKAN	<u>0.749</u>	<u>0.668</u>
	KGCAN	**0.771**	**0.684**

employing AUC and F1 score aligns with the goals of our KG-based recommendation system. The AUC indicates the model's capacity to rank positive instances higher than negative ones, which is crucial to recommendation accuracy. Meanwhile, the F1 score offers a balanced picture of precision and recall, addressing the tradeoff in the recommendation scenario between lessening false positives and false negatives. In CTR prediction, the model is trained using a training set, and once the model is trained, a testing set is used to predict the probability that a user would like to interact with an item. The Adam optimizer [35] is utilized in this work for the optimization of our model, whereas the Xavier initializer [36] is used for initializing the parameters of the model.

The model is implemented using PyTorch (a deep learning framework), and different hyperparameters are chosen using grid search. The batch size is set to 1024, whereas the embedding size is selected from {8, 16, 32, 64, 128}. Other hyperparameters such as the learning rate are selected among $\{10^{-3}, 4 \times 10^{-3}, 10^{-2}, 4 \times 10^{-2}\}$, whereas the regularization parameter is selected from $\{10^{-5}, 5 \times 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}\}$. The user triple set size and the item triple set size are empirically chosen, and it is 64. The optimal layer size for different datasets is different and is explained in Section 5.7. For the baselines, we have utilized open source implementations to have experimental results on three benchmark datasets.

5.4 Performance Comparison (RQ1)

We have conducted experiments to compare the performance of the proposed model KGCAN with recent recommendation models on three benchmark datasets. The experimental results w.r.t. AUC and F1 score are reported in Table 3, where the best performance is shown in bold type and the second best is underlined. From the table, we have the following findings:

- KGCAN shows significant performance when compared with the state-of-the-art recommendation models using the statistical Wilcoxon signed-rank test [37] ($p < 0.5$). This highlights the importance of taking those challenges into consideration which we mentioned in Section 1. More formally, KGCAN shows improved performance w.r.t. the second-best model by 1.0%, 3.3%, and 2.9% w.r.t. AUC in the ML, FM, and BC datasets. It is important to mention here is that this increase in performance can also be due to the contextual information that is incorporated into the model.
- Except for NCF, all baselines incorporate the KG into the recommendation model. NCF shows the worst performance w.r.t. AUC in all datasets when compared with other baselines. This highlights the importance of incorporating the KG into the recommendation scenario, as it minimizes the data sparsity problem. Moreover, CKAN usually performs better in terms of AUC after KGCAN, which highlights the importance of considering incorporation of heterogeneous propagation as well as the attention mechanism.
- In the ML dataset, the performances of different baselines are not much different because the ML dataset is dense and thus has more average interactions per user as compared to the FM and BC datasets. In this manner, the learning of an entity's representation and the encoding of contextual information do not significantly improve the performance of KGCAN.
- CKAN and KGAT usually show reasonable performance because both employ an attention mechanism in their model. One thing that distinguishes between the two is the heterogeneous propagation in CKAN, which adds more capacity in CKAN model learning, thus giving an edge to CKAN over KGAT. However, KGAT employs a CKG that captures the user interaction as well as a KG in the form of a unified relational graph.
- The dataset-wise comparison of all baselines shows that top performance (above 96.6%) is exhibited by the ML dataset, followed by the FM and BC datasets. Since these datasets belong to different domains (movie website, book community, and music platform), they have different instances and different average interactions per user. Moreover, the ML dataset is dense, with around 98 interactions per user, so the model has more to learn from the dataset that that of latent vector embedding propagation.

5.5 Ablation Study (RQ2)

In this work, the effectiveness of the proposed model is verified by conducting an ablation study. For this purpose, two variants of the proposed model are introduced, the details of which are given in the following. The experiments are conducted to compare the performance of the proposed model and the following variants on different datasets:

$KGCAN_{/contextual}$: In this variant, we have disabled the contextual representation of the user and item and only the encoded representation of propagation is considered. The experiments are conducted to compare the performance of this variant with that of the original model.

$KGCAN_{/attention}$: In this variant, the attentive module is disabled in the proposed model, and therefore it is assumed that each neighboring node of an entity is contributing equally. More specifically, the average of the neighboring nodes is taken, then the performance is compared with that of original model having an attention module.

In Table 4, the experimental results are presented, highlighting the significance of contextual representation of the user and item, as well as of the attention module. To summarize, we can obtain the following from the table:

Table 4. Performance Comparison of KGCAN and Its Two Variants (in terms of AUC)

	$\text{KGCAN}/_{\text{contextual}}$	$\text{KGCAN}/_{\text{attention}}$	KGCAN
ML	0.977	0.981	**0.986**
FM	0.826	0.843	**0.868**
BC	0.727	0.738	**0.771**

Table 5. Performance Comparison of User-Specific Component in the KGCAN

	Without User-Specific Component	With User-Specific Component
ML	0.972	0.984
FM	0.843	0.868
BC	0.741	0.772

- The performance of $\text{KGCAN}_{/\text{contextual}}$ is compared with that of KGCAN, which depicts the significance of contextualized representation in the original model. One possible reason for this is that the original information of the user and item is preserved and the bias due to multilayer propagation is minimized in contextualized representation.
- The attention module is significant and plays a vital role in the performance of the recommendation model. It is clear from Table 4 that $\text{KGCAN}_{/\text{attention}}$ has shown poor performance when compared with KGCAN with an attention module. The possible reason for this is due to the heterogeneity of the KG, which should not be ignored during the aggregation of neighboring nodes information.

5.6 Significance of User-Specific Component (RQ3)

In the recommendation scenario, item recommendation is done depending on the user's historical information and that of closely related neighboring entities. Since the proposed recommendation model is equipped with a user-specific component, it is crucial to determine its significance. For this purpose, experiments are conducted for each dataset by removing the user-specific component (by removing m_u from Equation (9)), then the results are compared with those of the original model. The performance is recorded in Table 5. It is clear that the performance is enhanced when a user-specific component is considered, thus highlighting the significance of a user-specific component.

5.7 Hyperparameters Study (RQ4)

Extensive experimentations are conducted to validate the performance of KGCAN on different hyperparameters such as propagational layer size, selection of aggregation function, and dimension of embedding. In the following section, experimental results are reported, along with a discussion.

Depth of Layer. We conducted experiments to check the effect of different depth of layer on the model's performance. The experimental results are recorded in Table 6. It can be seen from the table that each dataset behaves differently on the different depth of layer, whereas the best performance is highlighted in bold type. In the case of the ML dataset, depth of single layer shows best performance, whereas in the case of the FM and BC datasets, depth of layer is set to 3 for achieving better performance. One possible explanation for this is that as we increase the depth of layer, more information is encoded and thus performance increases. However, as we increase the depth of layer, noise is also being added into the entity representation, thus performance de-

Table 6. Effect of Depth of Layer (w.r.t. AUC)

No. of Layers	1	2	3	4
ML	**0.9866**	0.9861	0.9852	0.9847
FM	0.8612	**0.8634**	0.8586	0.8540
BC	0.7627	**0.7633**	0.7587	0.7532

Table 7. Effect of Different Aggregators on Performance (w.r.t. AUC)

	Pool	Sum	Concatenation
ML	0.9264	0.9836	**0.9866**
FM	0.7916	0.8452	**0.8634**
BC	0.7121	0.7345	**0.7632**

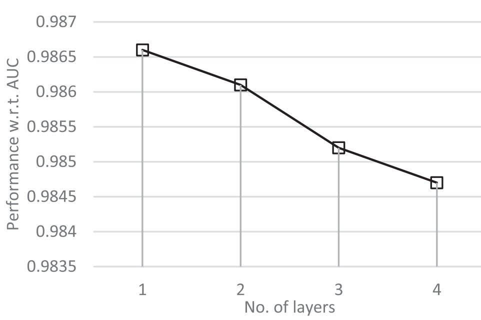

Fig. 4. Effect of the number of layers in the ML dataset.

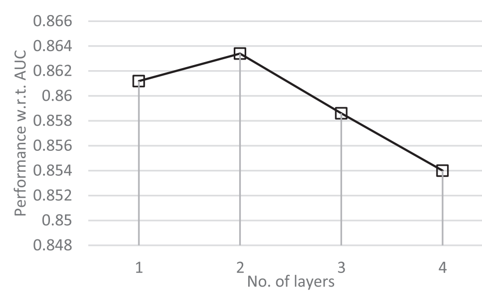

Fig. 5. Effect of the number of layers in the FM dataset.

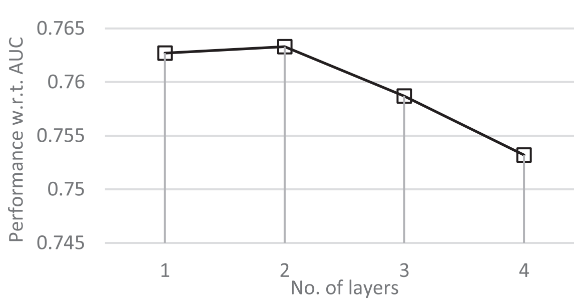

Fig. 6. Effect of the number of layers in the BC dataset.

creases. Moreover, as we increase the depth of layer, training time of the model also increases, whereas performance decreases. Figures 4, 5, and 6 show model performance for the ML, FM, and BC datasets on different depth of layer, respectively.

Selection of Aggregation Function. To determine the effect of aggregation function on the proposed model, we have conducted experiments and the results are reported in Table 7. From the table, it is clear that the concatenation aggregator significantly outperforms from the sum or pool aggregator. The possible explanation for this is that the concatenation aggregator captures more information as compared to the sum or pool aggregator. The concatenation aggregator occupies

Table 8. Effect of the Embedding's Dimensions

d	8	16	32	64	128
ML	0.972	0.974	0.981	**0.984**	0.978
FM	0.854	0.859	0.863	0.868	**0.871**
BC	0.757	0.768	0.771	**0.772**	0.770

more memory and thus is able to capture each single tensor. However, the sum and pool aggregators yield only a single value after incorporating tensors from multiple layers.

Dimension of Embedding. To verify the effect of varying dimensions of embedding, we kept the same dimensional parameters for the entity as well as the relation embeddings, thus reducing the bias that may be induced due to varying dimensional parameters. The experimental results are recorded in Table 8. From the table, it is clear that as we increase the embedding dimension d, the performance of the model also increases up to a certain threshold, which is 64 for the ML dataset. The performance starts decreasing when we increase the embedding dimension d further. One possible reason for this change in the performance is due to the fact that more information is captured as we increase d, but increasing d also causes the model to overfit, which essentially means that more information is captured than that of model's capacity. It is worth mentioning here that KGCAN shows strong tolerance to the change in the embedding dimensions and thus fewer fluctuations in the performance. This makes our proposed model less dependent on the hyperparameter's settings.

5.8 Health Domain: A Case Study (RQ5)

A lot of healthcare data is scattered on the internet that may hinder patients in acquiring useful and relevant health-related information. Besides patients, even medical professionals find it difficult to align their resources in a patient-oriented way. For this purpose, it becomes the need of the hour to design and deploy recommendation systems in the healthcare domain. In this way, not only patients but also medical professionals are facilitated and thus channelize the available resources to have accurate decisions related to the overall well-being of patients.

In the healthcare domain, the drug rating dataset [38] is widely used in the recommendation scenario. From this real-world dataset, we randomly selected four patients (*p1, p2, p3, p4*) who have interactions with different drugs. The patients have given ratings to the drugs on the scale of 1 to 10. From the drug rating dataset, a real example of patient interaction with the drug is shown in Figure 7, along with the reasoning to recommend the drug to the given patient with history similar to that of another patient. The following key observations are drawn from the figure:

- High-order connectivity incorporated into the recommendation system enriches the patient representation, as relevant drugs are recommended to her and to other patients with similar medical history. Thus, the patient's uninteracted drugs are recommended by the CF. For example, in Figure 7, *p2* interacted with *Nexplanon*, whereas *p3* interacted with *Nexplanon* and *Augmentin XR*. Due to CF, *Augmentin XR* is recommended to *p2* as well (represented by the dotted line) since the similar user *p3* also consumed it.
- The attention mechanism plays a vital role in discriminating the importance of different neighboring nodes. For example, *p2* has interacted with *Ativan* as well as *Nexplanon*, but *p2* has a strong attentive weight with *Nexplanon* as compared to *Ativan* (the attentive weights are obtained from the *p2* rating of these drugs). This essentially means that in drug rating data, the attention mechanism is also equally applicable to describe the different importance of the interacted drugs for the given patient.

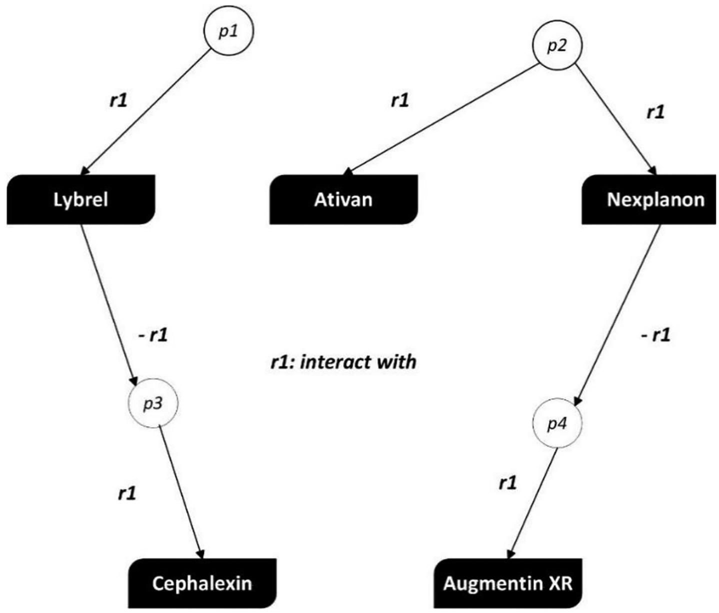

Fig. 7. Real example from a drug rating dataset.

- Since the given data is sparse, the patients may have rated or interacted with few drugs. Therefore, the utilization of side information assists in enhancing the recommendation performance. That is why, in the proposed KGCAN model, KG-enhanced recommendations are produced to provide reasoning of the user's possible items of interest.

5.9 Case Study: Personalized Recommendations of Medication

Background. A healthcare center with various patient groups aims to increase medication adherence and lower the risk of adverse drug reactions in its patients. Many patients are given several medications, making it hard to make sure that they understand what is prescribed, take them as advised, and avoid potentially hazardous interactions. Electronic Health Records (EHRs) contain a patient's medical history, diagnosis, prescription drugs, lab findings, and demographic information.

Implementation. The healthcare center deploys the KGCAN recommendation model, which is based on a KG and a user-specific attention mechanism. A KG is created by combining medical ontologies and historical patient information. It contains medication interactions, medical issues, and patient attributes. The system employs the KGCAN model, which gives personalized medication recommendations to each patient. This model takes input of the patient's medical history, current medicines, and ‘demographics to adapt its recommendations.

Observations. The healthcare center sees significant gains in compliance with medications by personalizing medication recommendations to unique patient profiles. Patients are able to understand and adhere to their prescription treatments, resulting in improved health results. The patient-specific data obtained through the system provides essential insights to the healthcare system. For better recommendations, patterns in drug adherence and health outcomes are investigated.

Practical Implications for Health Informatics. The utilization of KGCAN highlights the practical significance of customized medical recommendations. Health informatics allows for better and personalized treatments by taking individual patient information and medical histories into consideration. This case study shows that health informatics could enhance medication safety by identifying potential medicine interactions using a KG. This is a major implication for patient safety as well as for lowering medical costs related to adverse situations. The KGCAN recommendation system, when integrated with a KG and a user-specific attention mechanism, could lead to practical implications in health informatics for increasing medication compliance, patient safety, and personalized healthcare recommendations.

6 CONCLUSION

This work presented an end-to-end responsible recommendation model called *KGCAN*, which explicitly encodes contextual and relational information for entities. KGCAN effectively aggregates collaborative signals latent in user-item interactions and KG information. To achieve this, KGCAN leverages a heterogeneous propagation mechanism for collaborative information and a KG. Additionally, KGCAN employs a user-specific attention mechanism to discern personalized preferences for entities. By aggregating contextual information to preserve entity originality and mitigate propagation bias, KGCAN achieves fruitful results. We compared KGCAN with state-of-the-art KG-based recommendation models on three benchmark datasets, and the experimental results demonstrated the significant superiority of KGCAN over the baselines. In future research, we are interested in selecting the most relevant higher-order neighbors for given entities to enhance recommendation performance. Furthermore, exploring path selection methods that aggregate high-order neighbors for entities holds potential in this domain.

The limitations include the degradation of the model's performance in data scarcity scenarios and the model's sensitivity to the diversity of the training data. KGs can be diverse and complex in nature; thus, they may pose scalability issues. As the KG grows, more computational time may be needed for traversing. Therefore, handling such massive KGs may lead to performance degradation. KGs are domain specific, which essentially means that they may not cover multiple domains. This confines their applicability in different recommendation scenarios and thus may not work well for varied content.

In the future, the robustness of KGCAN may be improved by employing more sophisticated encoding approaches so that the uniqueness of each entity may not disappear due to aggregation of multiple entities. Moreover, more advanced data augmentation techniques need to be explored which address the limitations related to data scarcity, to further improve the fairness and robustness of the proposed model.

ACKNOWLEDGMENTS

The authors would like to thank GIK Institute for providing research facilities.

REFERENCES

[1] C. Yin, J. Wang, and J. H. Park. 2017. An improved recommendation algorithm for big data cloud service based on the trust in sociology. *Neurocomputing* 256 (2017), 49–55. DOI: https://doi.org/10.1016/j.neucom.2016.07.079

[2] Y. Koren, R. Bell, and C. Volinsky. 2009. Matrix factorization techniques for recommender systems. *IEEE Comput.* 42, 8 (2009), 30–37.

[3] M. Jamali and M. Ester. A matrix factorization technique with trust propagation for recommendation in social networks. In *Proceedings of the 4th ACM Conference on Recommender Systems (RecSys'10).* 135–142. DOI: https://doi.org/10.1145/1864708

[4] X. He, L. Nie, X. Hu, and T.-S. Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web (WWW'17).* 173–182. DOI: https://doi.org/10.1145/3038912.3052569

[5] Y. Tay, L. Anh Tuan, and S. C. Hui. 2018. Latent relational metric learning via memory-based attention for collaborative ranking. In *Proceedings of the 2018 World Wide Web Conference.* 729–739.

[6] H. Wang, F. Zhang, J. Wang, M. Zhao, W. Li, X. Xie, and M. Guo. 2018. RippleNet: Propagating user preferences on the knowledge graph for recommender systems. In *Proceedings of the International Conference on Information and Knowledge Management (CIKM'18).* 417–426. DOI: https://doi.org/10.1145/3269206.3271739

[7] Y. Cao, X. Wang, X. He, Z. Hu, and T. S. Chua. 2019. Unifying knowledge graph learning and recommendation: Towards a better understanding of user preferences. In *Proceedings of the 2019 World Wide Web Conference.* 151–161. DOI: https://doi.org/10.1145/3308558.3313705

[8] Z. Sun, J. Yang, J. Zhang, A. Bozzon, L.-K. Huang, and C. Xu. 2018. Recurrent knowledge graph embedding for effective recommendation. In *Proceedings of the 12th ACM Conference on Recommended Systems (RecSys'18).* 297–305. DOI: https://doi.org/10.1145/3240323

[9] B. Hu, C. Shi, W. X. Zhao, and P. S. Yu. 2018. Leveraging meta-path based context for top-N recommendation with a neural co-attention model. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'18).* 1531–1540. DOI: https://doi.org/10.1145/3219819.3219965

[10] F. Zhang, N. J. Yuan, D. Lian, X. Xie, and W. Y. Ma. 2016. Collaborative knowledge base embedding for recommender systems. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'16).* 353–362. DOI: https://doi.org/10.1145/2939672.2939673

[11] R. Yin, K. Li, G. Zhang, and J. Lu. 2019. A deeper graph neural network for recommender systems. *Knowledge-Based Syst.* 185 (2019), 105020.

[12] A. Micheli. 2009. Neural network for graphs: A contextual constructive approach. *IEEE Trans. Neural Netw.* 20, 3 (2009), 498–511.

[13] C. Hsu and C. Te Li. 2021. RetaGNN: Relational temporal attentive graph neural networks for holistic sequential recommendation. In *Proceedings of the Web Conference 2021 (WWW'21).* 2968–2979. DOI: https://doi.org/10.1145/3442381.3449957

[14] Z. Wang, G. Lin, H. Tan, Q. Chen, and X. Liu. 2020. CKAN: Collaborative knowledge-aware attentive network for recommender systems. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval.* 219–228. DOI: https://doi.org/10.1145/3397271

[15] X. Wang, X. He, Y. Cao, M. Liu, and T.-S. Chua. 2019. KGAT: Knowledge graph attention network for recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* 950–958. DOI: https://doi.org/10.1145/3292500

[16] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun. 2020. Graph neural networks: A review of methods and applications. *AI Open* 1 (2020), 57–81.

[17] T. N. Kipf and M. Welling. 2017. Semi-supervised classification with graph convolutional networks. In *Proceedings of the 5th International Conference on Learning Representations (ICLR'17).*

[18] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun. 2014. Spectral networks and locally connected networks on graphs. In *Proceedings of the 5th International Conference on Learning Representations (ICLR'17).*

[19] M. Defferrard, X. Bresson, and P. Vandergheynst. 2016. Convolutional neural networks on graphs with fast localized spectral filtering. In *Proceedings of the 30th International Conference on Neural Information Processing Systems (NIPS'16).* 3844–3852.

[20] A. Micheli. 2009. Neural network for graphs: A contextual constructive approach. *IEEE Trans. Neural Netw.* 20, 3 (2009), 498–511. DOI: https://doi.org/10.1109/TNN.2008.2010350

[21] W. L. Hamilton, R. Ying, and J. Leskovec. 2017. Inductive representation learning on large graphs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems.* 1025–1035.

[22] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio. 2018. Graph attention networks. In *Proceedings of the 6th International Conference on Learning Representations (ICLR'18).*

[23] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang. 2020. LightGCN: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'20).* 639–648. DOI: https://doi.org/10.1145/3397271.3401063

[24] F. Liu, Z. Cheng, L. Zhu, Z. Gao, and L. Nie. 2021. Interest-aware message-passing GCN for recommendation. In *Proceedings of the Web Conference 2021 (WWW'21).* 1296–1305. DOI: https://doi.org/10.1145/3442381.3449986

[25] H. Wang, F. Zhang, X. Xie, and M. Guo. 2018. DKN: Deep knowledge-aware network for news recommendation. In *Proceedings of the Web Conference 2018 (WWW '18).* 1835–1844. DOI: https://doi.org/10.1145/3178876.3186175

[26] H. Wang, M. Zhao, X. Xie, W. Li, and M. Guo. 2019. Knowledge graph convolutional networks for recommender systems. In *Proceedings of the Web Conference 2019 (WWW'19),* 3307–3313. DOI: https://doi.org/10.1145/3308558.3313417

[27] H. Wang, F. Zhang, M. Zhang, J. Leskovec, M. Zhao, W. Li, and Z. Wang. 2019. Knowledge-aware graph neural networks with label smoothness regularization for recommender systems. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'19).* 968–977. DOI: https://doi.org/10.1145/3292500

[28] X. Yu, X. Ren, Y. Sun, B. Sturt, U. Khandelwal, Q. Gu, B. Norick, and J. Han. Recommendation in heterogeneous

information networks with implicit user feedback. In *Proceedings of the 7th ACM Conference on Recommended Systems (RecSys'13).* 347–350. DOI: https://doi.org/10.1145/2507157

[29] Y. Dong, N. V. Chawla, and A. Swami. 2017. Metapath2vec: Scalable representation learning for heterogeneous networks. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 135–144. DOI: https://doi.org/10.1145/3097983.3098036

[30] Y. Liu, S. Yang, Y. Xu, C. Miao, M. Wu, and J. Zhang. 2023. Contextualized graph attention network for recommendation with item knowledge graph. *IEEE Trans. Knowl. Data Eng.* 35, 1 (2023), 181–195. DOI: https://doi.org/10.1109/TKDE.2021.3082948

[31] B. Zamanlooy and M. Mirhassani. 2014. Efficient VLSI implementation of neural networks with hyperbolic tangent activation function. *IEEE Trans. Very Large Scale Integr. Syst.* 22, 1 (2014), 39–48. DOI: https://doi.org/10.1109/TVLSI.2012.2232321

[32] R. Memisevic, C. Zach, M. Pollefeys, and G. E. Hinton. 2010. Gated softmax classification. In *Advances in Neural Information Processing Systems 23 (NIPS'10).* 1603–1611.

[33] H. Wang, F. Zhang, M. Zhang, J. Leskovec, M. Zhao, W. Li, and Z. Wang. 2019. Knowledge-aware graph neural networks with label smoothness regularization for recommender systems. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'19).* 968–977, DOI: https://doi.org/10.1145/3292500.3330836

[34] L. Sang, M. Xu, S. Qian, and X. Wu. 2021. Knowledge graph enhanced neural collaborative recommendation. *Expert Syst. Appl.* 164 (2021), 113992. DOI: https://doi.org/10.1016/j.eswa.2020.113992

[35] D. P. Kingma and J. Ba. 2014. Adam: A method for stochastic optimization. *arXiv:1412.6980* (2014).

[36] X. Glorot and Y. Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics.* 249–256.

[37] G. Shani and A. Gunawardana. 2011. Evaluating recommendation systems. In *Recommender Systems Handbook.* Springer, 257–297.

[38] F. Gräßer, S. Kallumadi, H. Malberg, and S. Zaunseder. 2018. Aspect-based sentiment analysis of drug reviews applying cross-domain and cross-data learning. In *Proceedings of the 2018 International Conference on Digital Health.* 121–125.

[39] R. Sun, X. Cao, Y. Zhao, J. Wan, K. Zhou, F. Zhang, Z. Wang, and K. Zheng. 2020. Multi-modal knowledge graphs for recommender systems. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management.* 1405–1414.

[40] X. Sha, Z. Sun, and J. Zhang. 2021. Hierarchical attentive knowledge graph embedding for personalized recommendation. *Electron. Commer. Res. Appl.* 48 (2021), 101071.

[41] H. Lin, Y. Liu, W. Wang, Y. Yue, and Z. Lin. 2017. Learning entity and relation embeddings for knowledge resolution. *Procedia Comput. Sci.* 108 (2017), 345–354.

Received 3 July 2023; revised 18 November 2023; accepted 14 December 2023