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, ehsanelahi@giki.edu.pk

SAJID ANWAR

Institute of Management Sciences, Peshawar, 25000, Pakistan. sajid.anwar@imsciences.edu.pk

BABAR SHAH

Zayed University, Abu Dhabi, UAE. babar.shah@zu.ac.ae

ZAHID HALIM*

Ghulam Ishaq Khan Institute of Engineering Sciences and Technology, Topi, Pakistan, zahid.halim@giki.edu.pk

ABRAR ULLAH

Heriot Watt University, United Kingdom. a.ullah@hw.ac.uk.

IMAD RIDA

Université de Technologie de Compiègne France: BMBI Laboratory, University of Technology of Compiègne, France. imad.rida@utc.fr

MUHAMMAD WAQAS

School of Engineering, Edith Cowan University, Joondalup, WA, 6027, Australia, and College of Information Technology, University of Bahrain, Sakheer, 32038, Bahrain. engr.waqas2079@gmail.com

With the 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 Knowledge Graph Enhanced Contextualized Attention-Based Network (KGCAN). It 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 KG-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.

CCS CONCEPTS • Information systems→Data management systems→Information integration

ACM 2157-6904/2024/01-ART
http://dx.doi.org/10.1145/3641288

Additional Keywords and Phrases: Responsible recommendation system, Graph neural network, Knowledge graph, Attention mechanism.

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. Since the users do not know as what they actually need from the large list of products, the system is displaying. Also, there come the new users which may be unaware of the top trending items in the market, leveraging the recommendation system allow the users to choose from top selling items. That is why recommendation system are being deployed in e-commerce websites to capture the changing needs of their customers. Different users may have different needs 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 to be modified so that changing needs of the users as well as of businesses are fulfilled. In the recent years, researchers are 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, makes MF less applicable in real-world scenarios.

Although the traditional recommendation systems such as MF have been utilized and remain successful, but they have their own limitations of cold start problem. In the cold start problem, the system finds it difficult to recommend some item (can be book, some music, news etc.) for the new user in the system. Additionally, they also suffer from 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 which leverage 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. Firstly, the embedding component transforms the user and item into vector representations. Secondly, 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 Graph (KG) have been utilized and gained much attention from researchers [6, 7]. KG provides the structured and semantic enrich representation of the real-world knowledge. It consists of different entities which are connected among each other with some meaningful relations that

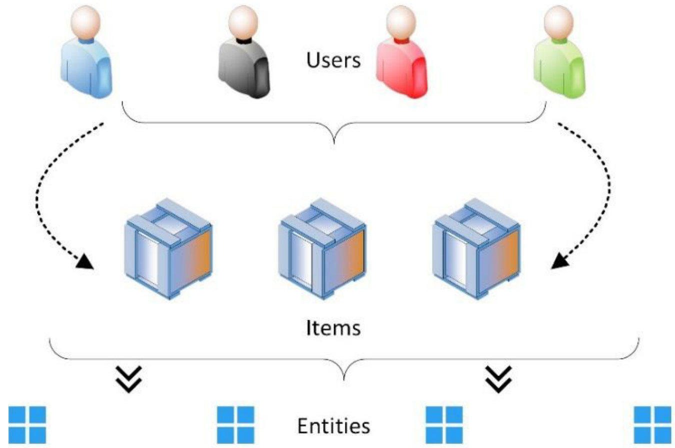

Figure 1: A toy example of user-item interaction and knowledge graph

give more power to them. These semantic relations in KG provide the ability of reasoning and explaining of various recommendation related tasks. In the recommendation scenario, there are widely being used in the recent literature owing to their ability to provide rich information and to overcome the data sparsity as well as the cold start problem. 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, user-item interactions, which improves the 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 knowledge graph 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 over-smoothing 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, 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 Fig. 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, while 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 as well as 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 consider relation into consideration. As KG has entities which are connected to each other with different types of relations. So, ignoring relational information make them insufficient to model the preferences of user. Relational information is necessary to 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. Lack to consider relational information may result in homogeneous recommendations, limiting the 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 famous 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. In summary, the main contributions of this work can be summarized as follows:

- A responsible recommendation model KGCAN is proposed which explicitly exploits interaction information as well knowledge graph 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 multi-layer 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 paper 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 knowledge graph (KG). Existing GNN-based recommendation models have limitations in addressing these challenges. Firstly, 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. Secondly, while 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 multi-layer propagation. The contextualized representation is obtained by concatenating the learnt representation of user/item with their original representations. By doing so, over-smoothing 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. 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, 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 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. On the other hand, 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 paper is organized as follows. Section 2 provides a description of recent and relevant works that utilize KG for recommendation purposes. Section 3 presents the task formulation of the research problem to be addressed, while 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 the recent work done in the domain of recommendation system, using graph neural network and knowledge graph.

2.1 Graph neural network-based recommendation

Recently, convolution neural network (CNN) is extended to graph neural network (GNN) [16], to model the graph structure data. Since graph data is of irregular structure having no fixed number of nodes in each graph, so it may not be appropriate to apply CNN on the graph data, having no fixed size matrices. The operation of 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 to capture 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; 1) the spectral methods, and 2) the spatial methods.

Table 1: Notations along with their description

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

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

In spatial methods, 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 the 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 previous layer, residual connection is utilized at each layer. The number of neighboring nodes may vary for each node, so to tackle this, sampling approach is proposed in the literature. These sampling approaches sample the fixed number of neighboring nodes and then 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.

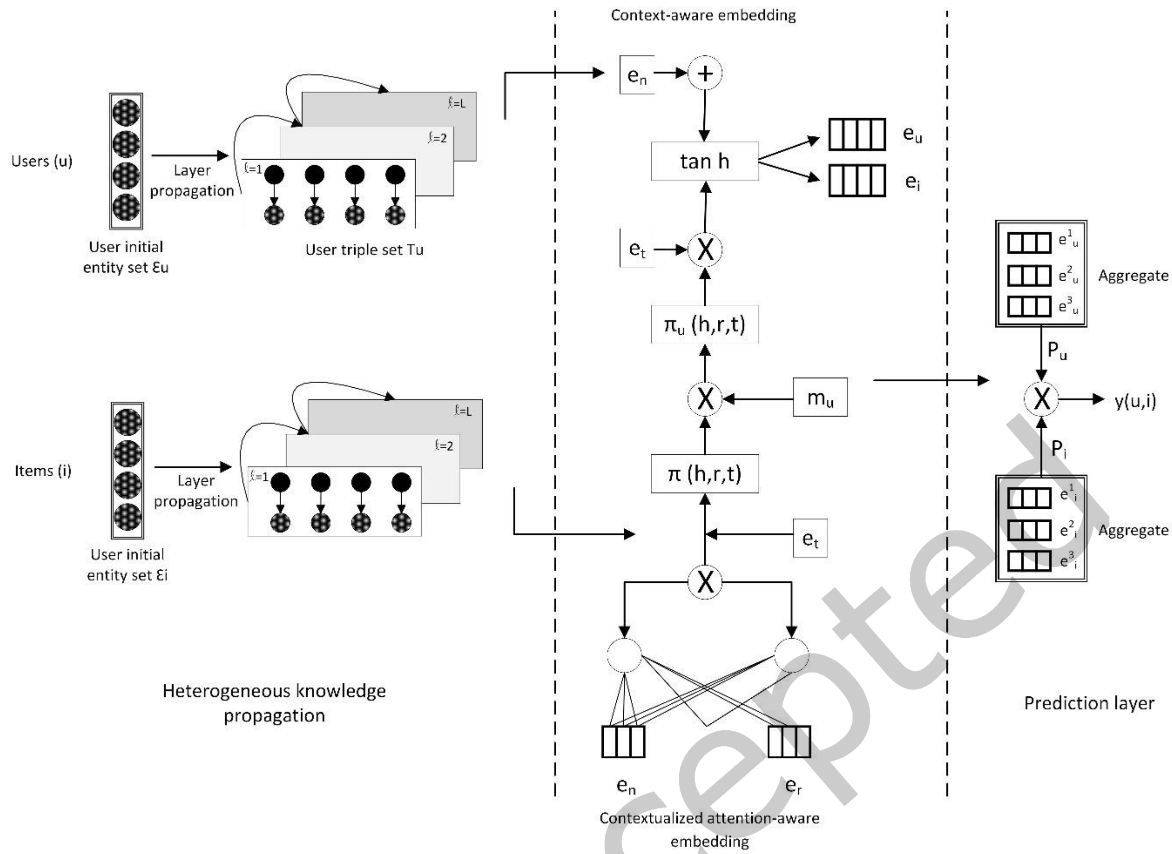

Figure 2: Overall framework of the proposed recommendation model

The authors in [22] introduce the concept of attention mechanism GAT, which argue 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 next item based on the user's previous interacted items. LightGCN [23] is an approach in which an attempt is done to make the GCN a light model. The authors argue that feature transformation as well as no-linearity (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 try to reduce the over-smoothing problem which 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 attention mechanism and thus treat every neighbor equally, which may not be rational in the real-world scenario.

2.2 Knowledge graph-based recommendation

In literature, KG is being widely adopted in the recommendation scenario. Knowledge graph-based recommendation work is divided into three main categories; 1) Embedding based methods, and 2) path-based methods, and 3) propagation-based methods. The detail of each is given in the following sections.

Embedding based methods: In existing embedding-based methods, entities in KG are embedded to vectorized form, while 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 recommender system. In CKE [10] and DKN [25], semantic embedding of the nodes of KG is generated, through Knowledge Graph Embedding (KGE) approaches. Afterwards, these generated embeddings are then fed into recommender system to regularize the user and item representation learning. The authors in RippleNet [6] and KGCN [26] utilize graph neural network to embed the item in 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 the KGCN by emphasizing on the label smoothness to ensure the regularization over the edges of neighboring entities.

Path-based methods: In path-based methods, paths and meta paths are designed to infer the preferences of user. More formally, the representation of each entity in KG is enriched with higher order connectivity information, by designing paths form start entity to *L*-hops away entity. Since there can be many paths form a given entity to other entities, two approaches are there. First approach is selective approach [8], in which most significant paths are selected and thus used to enrich the entity representation. Second approach is based on meta-paths patterns to provide a limit of paths [9]. The authors in [28] have proposed attribute-rich model known as Heterogeneous Information Network (HIN) for improving recommendation quality. In [29], the authors have employed meta-path based random walk component, which encode the heterogeneous neighborhood information for a node. CGAT [30] introduced the concept of biased random walk, which employ 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 KG size may reach up to million entities, this make the situation even impossible to efficiently design paths.

Propagation-based methods: As the name suggest, these methods operate on KG where information is propagated iteratively to provide auxiliary information for recommendation. These methods have attained much attention in the research community. For example, RippleNet [6] model which enrich the potential preferences of the user by propagating along KG links. However, 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 which present the collaborative knowledge graph (CKG) to integrate the user item interaction and KG. In KGAT, user and entities are treated in the same manner which may not be rational as users have different meaning in user-item bipartite graph whereas entities in KG represents different meaning. Moreover, KGAT model need to retrain itself for new upcoming user, so that recommendation can be provided to her. This make KGAT highly computationally expensive in real world where there are million entities. CKAN [14] presents the heterogeneous propagation in which collaboration propagation and KG propagation are integrated. This end to end framework leverage attentive embedding to learn varying weights of neighboring nodes, for each node. As the KG have diverse relations which is essential to be incorporated to have personalized recommendation, attention mechanism is tailored to incorporate this diverse information. But it does not encode the contextual information which 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 same type of entities while the relation between the two is treated as interaction. In recent years, more focus is being given to construction of subgraph i.e., only those entities and relations are considered in the subgraph which are relevant to the user-item pair. The authors in [40] works on the assumption that shorter distance between the two nodes represents 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. Since these methods rely on subgraph construction to have relevant user and item pair, but 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 which are being used in 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 the user has interacted with the item e.g. user u has purchased an item i or view an item i etc. 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 having relation information to have auxiliary information and thus it is utilized to alleviate the sparsity problem of 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 triple (Christopher, film.film.actor, Batsman Begins), the relation is actor and Christopher is head entity while Batsman Begins is the tail entity. This triple is described as Christopher is the film actor of the movie Batsman Begins. One thing to highlight here is that movie is an item for which semantic rich information is obtained by utilizing 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 $\hat{y}_{ui} = \mathcal{F}(u, i \mid Y, G, \theta)$, where θ denotes all the model's parameters of probability function $\mathcal{F}$. Table 1 shows all the notations used in this article.

4 PROPOSED FRAMEWORK

In this work, a KG-based recommendation framework, i.e., KGCAN is presented which assists in the selection of relevant information to have user-specific recommendations. The overall framework of the proposed recommendation model is shown in the Fig. 2. As depicted in the figure there are three main components of the framework: (1) heterogeneous knowledge propagation, (2) contextualized attention-aware embedding, and (3) prediction layer. The details of these components are elaborated in the following sections.

Optimization algorithm

Input:

User-item interaction matrix Y,

Knowledge graph G

Output:

Probability score $\hat{y}(u,i)$

```
Randomly initialize all 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 ŷ(u,i);
    Compute the gradients from Eq.  for the given batch;
    Update models' parameters Θ by gradient descent with learning rate η;
end for
return ŷ(u,i)
```

Algorithm 1: The optimization procedure

4.1 Heterogeneous knowledge propagation

This layer is heterogeneous as it is composed of two different modules; collaboration propagation information and knowledge graph information. In collaboration propagation, vital collaborative signal is encoded which is then used to enrich the representation of both user and item. In knowledge graph information, relevant information is propagated through the links of KG, thus provide auxiliary side information. The detail of these modules is given in the following sections.

Collaboration propagation information: Collaborative filtering aims to identify the similar users having almost similar interests of items. In traditional approaches, users and items are transformed into independent latent vectors and these 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, since the items interacted by a user u represents her preference and can be used to enrich the representation of that user. For a user u, initial seed set is constructed from its historical items (interacted set of items) and this initial seed set is then propagated in KG with the help of alignment set $\boldsymbol{Z}$. The definition of initial user u entity set is given as follows;

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

where $\mathcal{E}_u$ represents the initial entity set of user u, Z represents the alignment set which 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;

$$I_i = \{i_u \mid u \in \{u \mid y_{ui} = 1\} \text{ and } y_{ui_u} = 1\} \qquad (2)$$

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;

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

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

Knowledge graph information: We build upon the GCN architecture in which feature representation is learnt for each entity node in the KG. Unlike traditional KG embedding approaches which considers the directly connected neighbors of an entity, this study aims to find the higher-order neighbors of an entity in KG with the help of GCN approach. One layer of 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 is required rather the feature vector representation of each entity is learnt. 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 user and item entity set is given as follows,

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

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

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 extended entity set obtained from KG propagation, we can formally define the lth triple set for user u and item i.

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

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

Utilizing KG as an auxiliary source of information is reasonable since neighbor entities (as well as neighbors of neighbors) enrich the user an item feature representation and making it capable of capturing latent relationship between user and item.

4.2 Contextualized attention-aware embedding

Consider 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*, so there is a need to distinguish the tail or neighboring entities and filter out those irrelevant or noisy entities. For this purpose, attention-aware mechanism is proposed in this paper, with which not only the proximity structure of graph is exploited but also elaborate the importance of different neighboring nodes. The attentive embedding of neighbor entity *t* is given as follows;

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

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

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

Here, tanh [31] is the activation function to provide non-linearity. 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 which calculates different attention scores for each user *u*, and it is given as follows;

Table 2: Datasets and their statistics

	ML	FM	BC
No. of users	138,159	1872	17,860
No. of items	16,954	3846	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

Table 3: Performance of different recommendation algorithms w.r.t AUC an F1 score

Dataset	**Model**	**AUC**	**F1 score**
ML	NCF	0.966	0.918
	Ripple Net	0.969	0.921
	KGNN-LS	0.972	0.926
	KGCN	0.974	0.924
	KGAT	0.973	0.925
	CKAN	0.974	**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	0.758
	CKAN	0.840	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	0.749	0.668
	KGCAN	**0.771**	**0.684**

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

p_u in equation 10 represents the users' embedding which we have acquired from the embedding table having user indices for every user. After this, we have utilized softmax function [32] to normalize the coefficients across all the triples which are there in the triple set.

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

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 item, we calculate the attention score for the tail entity e^t as given by;

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

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

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

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

In the same way, for the item, the contextualized representation given as,

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

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 which straight away integrates the original user and item embeddings with the embedding generated from the layers of graph neural network. In this way, the context aware embedding is generated which do not lack original user and item uniqueness.

4.3 Model prediction

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

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

$$agg_{\text{Pooling}}^{u} = LeakyReLU(\mathrm{W} \cdot \mathrm{pool}_{max}(R_u) + \mathrm{b}),$$

$$agg_{\text{Pooling}}^{i} = LeakyReLU(\mathrm{W} \cdot \mathrm{pool}_{max}(R_i) + \mathrm{b})$$

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

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

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

Concatenation aggregator: In this type of aggregator, multiple representations are concatenated to form a single vector, capturing the information of all vectors. Afterwards, non-linearity is applied to the concatenated vector.

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

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

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

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

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/un-interacted items for each user. We have the following loss function for our proposed model.

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

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

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

Here, Θ denotes the all the 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 prediction score is described in the Algorithm 1. The architectural diagram of the proposed model is given in the Fig. 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 given research questions:

RQ1. Whether does KGCAN perform better in terms of performance, when compared with state-of-the-art KG based recommendation models?

RQ2. How does different variants of KGCAN affect the performance of base model?

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

RQ4. How do different hyper-parameter settings e.g. depth of layer, embedding size, aggregation function influence the model's performance?

RQ5. How does 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 such 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 the state-of-the-art recommendation models, for comparing purposes. A brief description of these benchmark datasets is given as follows;

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

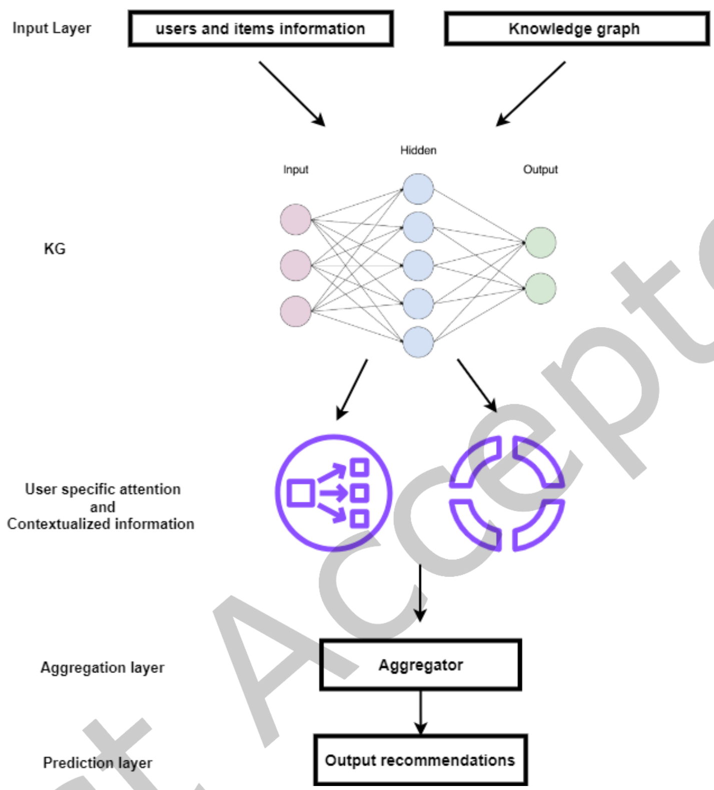

Figure 3: The architectural diagram of the KGCAN model

The interaction is given in the MovieLens-20M[1] (ML), Last.FM[2] (FM) and Book-Crossing[3] (BC) 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 un-interacted items from his historical information. To reduce the biasedness, the negative samples are of equal size as that of 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, KG based recommendation model is proposed and the KG of the used datasets are taken from the public repository https://github.com/xiangwang1223. For each dataset, from its whole KG, sub-KG is prepared on the basis of triples in KG having confidence level greater than 0.9. Moreover, we have removed the entities and items which are matching with other entities. Different statistics for the experimental datasets are summarized in Table 2.

5.2 Baselines

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

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

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

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

KGAT [15]: Unlike KGCN, this model utilizes attention module which assign varying weights to neighboring nodes in the collaborative knowledge graph which comprises of user-item interaction graph and KG.

CKAN [14]: It is a heterogeneous propagation model which make use of 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 [11] and [34]. For performance evaluation, click-through rate (CTR) prediction is used in which the performance of recommendation model is evaluated using AUC and F1 scores. The reason for employing AUC and F1 score aligns with the goals of our knowledge graph-based recommendation system. 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 trade-off in the recommendation scenario between lessening false positives and false negatives. In CTR prediction, the model is trained using training set and once the model is trained, testing set is used to predict the probability that a user would like to interact with an item. Adam [35] optimizer is utilized in this paper for the optimization of our model, whereas Xavier initializer [36] is used for initializing the parameters of the model.

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

5.4 Performance comparison (RQ-1)

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

- KGCAN shows significant performance when compared with the state-of-the-art recommendation models using statistical test Wilcoxon signed rank test [37] ($p<0.5$). This highlights the importance of taking those challenges into consideration which we mentioned in the introduction part. More formally, KGCAN shows improved performance with respect to the second-best model by 1.0%, 3.3% and 2.9% w.r.t AUC in ML, FM and BC datasets. It is important to mention here is that this increase in the performance can also be due to the contextual information which is incorporated into the model.
- Except NCF, all the baselines incorporate the KG into the recommendation model. NCF shows worst performance w.r.t AUC in all datasets when compared with other baselines. This highlights the importance of incorporating KG into the recommendation scenario as it minimizes the data sparsity problem. Moreover, CKAN usually perform better in terms of AUC after KGCAN which highlights the importance of incorporating heterogeneous propagation as well as attention mechanism into the consideration.
- In ML dataset, the performances of different baselines are not as much different due to the fact that ML dataset is dense and thus have more average interactions per user as compared to FM and BC datasets. In this way, entity's representation learning as well as encoding of contextual information does not give the major addition in the performance of KGCAN.
- CKAN and KGAT usually shows reasonable performances due to the fact that both employ attention mechanism in their model. One thing that distinguishes between the two is the heterogeneous propagation in CKAN which add more capacity in CKAN model learning, thus giving an edge to CKAN over KGAT. On the other hand, KGAT employs the collaborative KG which capture the user interaction as well as KG in the form of unified relational graph.
- The dataset wise comparison of all the baselines shows that top performance (above 96.6%) is exhibited by the ML dataset followed by FM and BC dataset. Since these datasets belong to different domains (movie website, books community and music platform) thus have different instances as well as different average interactions per user. Moreover, ML dataset is dense having 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 (RQ-2)

In this work, the effectiveness of the proposed model is verified by conducting the 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 as well as of these 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, attentive module is disabled in the proposed model, therefore it is assumed that each neighboring node of an entity is contributing equally. More specifically, the average of the neighboring nodes is taken and then the performance is compared with that of original model having attention module.

In Table 4, the experimental results are presented which highlights the significance of contextual representation of user and item as well as of the attention module. To summarize, the table following findings.

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

Table 4: Performance comparison of KGCAN and its two variants (in terms of AUC).

	KGCAN $/_{contextual}$	KGCAN $/_{attention}$	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	With user-specific
ML	0.972	0.984
FM	0.843	0.868
BC	0.741	0.772

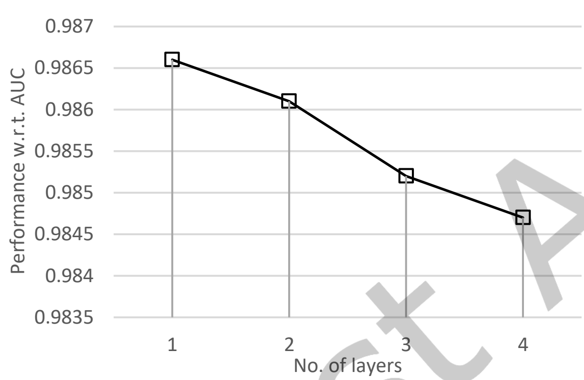

Figure 4: Effect of no. of layers in ML dataset

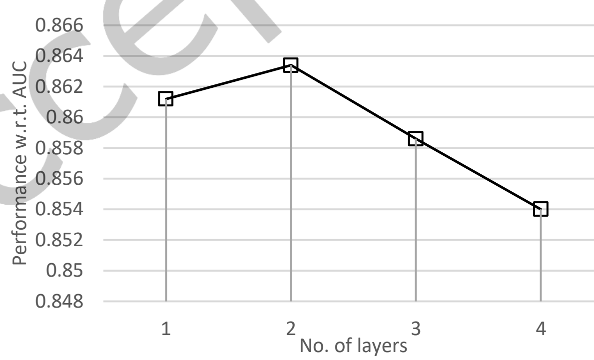

Figure 5: Effect of no. of layers in FM dataset

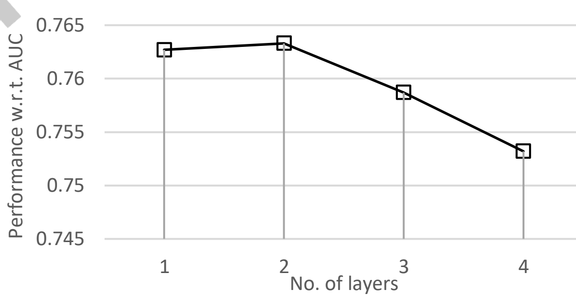

Figure 6: Effect of no. of layers in BC dataset

5.6 Significance of user-specific component (RQ-3)

In the recommendation scenario, item recommendation is done depending on the user's historical information as well as of closely related neighboring entities. Since the proposed recommendation model is equipped with 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 form equation 9) and then the results are compared with that of the original model. The performance is recorded in the Table 5. It is clear that the performance is enhanced when user-specific component is considered, thus highlighting the significance of user-specific component.

5.7 Hyper-Parameters Study (RQ-4)

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

Depth of layer: We have conducted the experiments to check the effect of different depth of layer on the model's performance. The experimental results are recorded in the Table 6. It can be seen from Table 6 that each dataset behaves differently on the different depth of layer, whereas, the best performance is highlighted in bold face. In case of ML dataset, depth of single layer shows best performance whereas in the case of FM and BC dataset, depth of layer is set to 3 for achieving better performance. One possible explanation for this is as we increase the depth of layer more information is encoded and thus performance increases. But as we increase the depth of layer, noise is also being added into the entity representation thus performance decreases. Moreover, as we increase the depth of layer, training time of the model also increases whereas performance decreases. Fig. 4, Fig. 5, and Fig. 6 shows the model performance for ML, FM and BC datasets on different depth of layers.

Table 6: Effect of depth of layers (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

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 the Table 7. From Table 7, it is clear that concatenation aggregator significantly outperforms from sum or pool aggregator. The possible explanation for this is that concatenation aggregator captures more information as compared to sum or pool aggregator. The concatenation aggregator occupies more memory thus able to capture each single tensor. On the other hand, sum and pool aggregators yields only single value after incorporating tensors from multiple layers.

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**

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 which may be induced due to varying dimensional parameters. The experimental results are recorded in the Table 8. From the Table 8, 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 ML dataset. The performance start 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 to mention here is that KGCAN shows strong tolerance to the change in the embedding dimensions thus less fluctuations in the performance. This make our proposed model less dependent on the hyper-parameter's settings.

Table 8: Effect of 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

5.8 Health domain: A case study (RQ-5)

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

In the healthcare domain, 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*) having their interaction with different drugs. The patients have given ratings to the drugs on the scale of 1-10. From the drug rating dataset, the real example of patient interaction with the drug is shown in the Fig. 7, along with the reasoning to recommend the drug to the given patient having similar history as that of her similar patient. The following key observations are drawn from the Fig. 7;

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

- Since the given data is sparse, as the patients may have rated or interacted with the few drugs. Therefore, the utilization of side information assists 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 patient's demographic information.

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

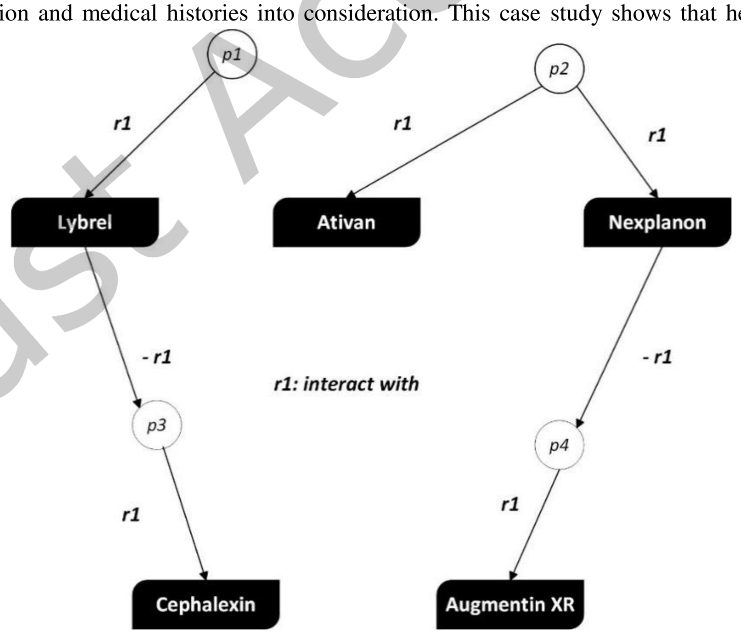

Figure 7: Real example from a drug rating dataset

could enhance medication safety by identifying potential medicine interactions using KG. This is a major implication for patient safety as well as lowering medical costs related to adverse situations. KGCAN recommendation system when integrated with KG and user-specific attention mechanism, could lead to practical implications in the health informatics for increasing medication compliance, patient safety, and personalized healthcare recommendations.

6 CONCLUSIONS

This work presents 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 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 compare KGCAN with state-of-the-art KG-based recommendation models on three benchmark datasets, and the experimental results demonstrate 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. KG 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, to handle such massive KG may lead to performance degradation. KG are domain-specific, which essentially means that they may not cover multiple domains. This confines their applicability in different recommendation scenarios, thus may not work well for varied content.

In future, the robustness of KGCAN may be improved by employing more sophisticated encoding approaches so that uniqueness of each entity may not be vanished 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. This work was supported by the GIK Institute graduate program research fund under the GA-F scheme.

Declarations:

Funding: This work was sponsored by the GIK Institute graduate research fund under GA-F scheme.

Conflict of Interest: All authors declare that they have no conflict of interest.

Ethical approval: No procedures performed in studies involved human participants.

REFERENCES

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

[2] Y. Koren, R. Bell, and C. Volinsky, “Matrix factorization techniques for recommender systems,” *IEEE Comput.*, vol. 42, no. 8, pp. 30–37, 2009.

[3] M. Jamali and M. Ester, “A Matrix Factorization Technique with Trust Propagation for Recommendation in Social Networks,” *Proc. fourth ACM Conf. Recomm. Syst. - RecSys ’10*, doi: 10.1145/1864708.

[4] X. He, L. Liao, H. Zhang, L. Nie, … X. H.-P. of the 26th, and undefined 2017, “Neural collaborative filtering,” *dl.acm.org*, pp. 173–182, 2017, doi: 10.1145/3038912.3052569.

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

[6] H. Wang *et al.*, "RippleNet: Propagating user preferences on the knowledge graph for recommender systems," in *International Conference on Information and Knowledge Management, Proceedings*, Oct. 2018, pp. 417–426. doi: 10.1145/3269206.3271739.

[7] Y. Cao, X. Wang, X. He, Z. Hu, and T. S. Chua, "Unifying knowledge graph learning and recommendation: Towards a better understanding of user preferences," *Web Conf. 2019 - Proc. World Wide Web Conf. WWW 2019*, pp. 151–161, May 2019, doi: 10.1145/3308558.3313705.

[8] Z. Sun, J. Yang, J. Zhang, A. Bozzon, L.-K. Huang, and C. Xu, "Recurrent Knowledge Graph Embedding for Effective Recommendation," *Proc. 12th ACM Conf. Recomm. Syst.*, 2018, doi: 10.1145/3240323.

[9] B. Hu, C. Shi, W. X. Zhao, and P. S. Yu, "Leveraging meta-path based context for top-n recommendation with a neural co-attention model," *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, pp. 1531–1540, Jul. 2018, doi: 10.1145/3219819.3219965.

[10] F. Zhang, N. J. Yuan, D. Lian, X. Xie, and W. Y. Ma, "Collaborative knowledge base embedding for recommender systems," *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, vol. 13-17-August-2016, pp. 353–362, Aug. 2016, doi: 10.1145/2939672.2939673.

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

[12] A. M.-I. T. on N. Networks and undefined 2009, "Neural network for graphs: A contextual constructive approach," *ieeexplore.ieee.org*, Accessed: Dec. 12, 2021. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/4773279

[13] C. Hsu and C. Te Li, "RetaGNN: Relational temporal attentive graph neural networks for holistic sequential recommendation," *Web Conf. 2021 - Proc. World Wide Web Conf. WWW 2021*, pp. 2968–2979, 2021, doi: 10.1145/3442381.3449957.

[14] Z. Wang, G. Lin, H. Tan, Q. Chen, and X. Liu, "CKAN: Collaborative Knowledge-aware Attentive Network for Rec-ommender Systems," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2020, pp. 219–228. doi: 10.1145/3397271.

[15] X. Wang, X. He, Y. Cao, M. Liu, and T.-S. Chua, "KGAT: Knowledge Graph Attention Network for Recommendation," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 950–958. doi: 10.1145/3292500.

[16] J. Zhou *et al.*, "Graph neural networks: A review of methods and applications," *AI Open*, vol. 1, pp. 57–81, 2020.

[17] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2017.

[18] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and locally connected networks on graphs," 2014.

[19] M. Defferrard, X. Bresson, P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," Advances in neural information processing systems, vol. 29, 2016.

[20] A. Micheli, "Neural network for graphs: A contextual constructive approach," *IEEE Trans. Neural Networks*, vol. 20, no. 3, pp. 498–511, 2009, doi: 10.1109/TNN.2008.2010350.

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

[22] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," 2018.

[23] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang, "LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation," in *SIGIR*, 2020, pp. 639–648. doi: 10.1145/3397271.3401063.

[24] F. Liu, Z. Cheng, L. Zhu, Z. Gao, and L. Nie, "Interest-aware message-passing GCN for recommendation," in *The Web Conference 2021 - Proceedings of the World Wide Web Conference, WWW 2021*, 2021, pp. 1296–1305. doi: 10.1145/3442381.3449986.

[25] H. Wang, F. Zhang, X. Xie, and M. Guo, "DKN: Deep knowledge-aware network for news recommendation," *Web Conf. 2018 - Proc. World Wide Web Conf. WWW 2018*, vol. 2, pp. 1835–1844, 2018, doi: 10.1145/3178876.3186175.

[26] H. Wang, M. Zhao, X. Xie, W. Li, and M. Guo, "Knowledge graph convolutional networks for recommender systems," in *The Web Conference 2019 - Proceedings of the World Wide Web Conference, WWW 2019*, May 2019, pp. 3307–3313. doi: 10.1145/3308558.3313417.

[27] H. Wang *et al.*, "Knowledge-aware Graph Neural Networks with Label Smoothness Regularization for Recommender Systems," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 968–977. doi: 10.1145/3292500.

[28] X. Yu *et al.*, "Recommendation in Heterogeneous Information Networks with Implicit User Feedback *," *Proc. 7th ACM Conf. Recomm. Syst.*, doi: 10.1145/2507157.

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

[30] Y. Liu, S. Yang, Y. Xu, C. Miao, M. Wu, and J. Zhang, "Contextualized Graph Attention Network for Recommendation with Item Knowledge Graph," *IEEE Trans. Knowl. Data Eng.*, vol. 4347, no. c, pp. 1–14, 2021, doi: 10.1109/TKDE.2021.3082948.

[31] B. Zamanlooy and M. Mirhassani, "Efficient VLSI implementation of neural networks with hyperbolic tangent activation function," *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 22, no. 1, pp. 39–48, 2014, doi: 10.1109/TVLSI.2012.2232321.

[32] R. Memisevic, C. Zach, M. Pollefeys, and G. E. Hinton, "Gated softmax classification," in *Advances in neural information processing systems*, 2010, pp. 1603–1611.

[33] H. Wang *et al.*, "Knowledge-aware graph neural networks with label smoothness regularization for recommender systems," *dl.acm.org*, vol. 19, pp. 968–977, Jul. 2019, doi: 10.1145/3292500.3330836.

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

[35] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” 2014.

[36] X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks,” in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 2010, pp. 249–256.

[37] G. Shani and A. Gunawardana, “Evaluating recommendation systems,” in *Recommender systems handbook*, Springer, 2011, pp. 257–297.

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

[39] R. Sun et al., “Multi-modal knowledge graphs for recommender systems,” in Proceedings of the 29th ACM international conference on information & knowledge management, 2020, pp. 1405–1414.

[40] X. Sha, Z. Sun, and J. Zhang, “Hierarchical attentive knowledge graph embedding for personalized recommendation,” Electron Commer Res Appl, vol. 48, p. 101071, 2021.

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