Journal of Information and Data Management, 2024, 15:1, doi: 10.5753/jidm.2024.3317

The Impact of Representation Learning on Unsupervised Graph Neural Networks for One-Class Recommendation

Marcos Paulo Silva Gôlo [**University of São Paulo** | ***marcosgolo@usp.br***]
Leonardo Gonçalves de Moraes [**University of São Paulo** | ***leonardo.g.moraes@usp.br***]
Rudinei Goularte [**University of São Paulo** | ***rudinei@icmc.usp.br***]
Ricardo Marcondes Marcacini [**University of São Paulo** | ***ricardo.marcacini@icmc.usp.br***]

Institute of Mathematics and Computer Sciences, University of São Paulo, Av. Trabalhador São Carlense, 400, Centro, São Carlos, Sãp Paulo, Brazil, 13566-590.

Received: 10 April 2023 • **Published:** 22 February 2024

Abstract We present a Graph Neural Network (GNN) using link prediction for One-class Recommendation. Traditional recommender systems require positive and negative interactions to recommend items to users, but negative interactions are scarce, making it challenging to cover the scope of non-recommendations. Our proposed approach explores One-Class Learning (OCL) to overcome this limitation by using only one class (positive interactions) to train and predict whether or not a new example belongs to the training class in enriched heterogeneous graphs. The paper also proposes an explainability model and performs a qualitative evaluation through the TSNE algorithm in the learned embeddings. The methods' analysis in a two-dimensional projection showed our enriched graph neural network proposal was the only one that could separate the representations of users and items. Moreover, the proposed explainability method showed the user nodes connected with the predicted item are the most important to recommend this item to another user. Another conclusion from the experiments is that the added nodes to enrich the graph also impact the recommendation.

Keywords: One-Class Learning, Recommender Systems, Graph Neural Networks, Link Prediction, One-Class Explainability, Graph Explainability

1 Introduction

Finding items that match users' interests is an important feature for any online platform, and recommender systems are indispensable for helping in this task. Still, these systems must find a way to deal with some issues, mainly modeling users' preferences and relations [Wu *et al*., 2020a; Khoali *et al*., 2022]. User preferences and relations essentially have a graph structure since nodes can be user and item, and the edges can be user-user, item-item, and user-item relations. Furthermore, graphs benefit from incorporating structured external information [Wu *et al*., 2020a; Ru *et al*., 2021].

Generally, the graph recommender systems studies model the graph with the user and items using the original modeling. The original modeling considers only user and item relations to generate the graph [Wu *et al*., 2020a]. However, those graph representations are incomplete, i.e., there are no interactions between all users and items. Even so, the graph has interactions the user would like to be recommended (positive) and interactions the user would not like (negative). However, positive interactions are more frequent in the real world than negative ones, i.e., there are few negative interactions [Khoali *et al*., 2022].

Traditional recommender systems need positive and negative interactions to recommend items to users. Therefore, we must cover the positive and negative interactions' scope in this scenario. Still, covering the scope of positive recommendations is easier, given the number of positive recommendations. However, covering the scope of negative interactions is challenging as these interactions have a more extensive scope and few interactions [Khoali *et al*., 2022].

In this scenario, One-Class Learning (OCL) arises as an alternative. OCL algorithms use only one class to train and predict whether or not a new example belongs to the training class. In this sense, OCL is a learning that only needs interest class instances to train and can recommend or not an item for a user. Thus, OCL has the advantage of not having to cover the scope of non-recommendations [Pan *et al*., 2008; Gôlo *et al*., 2021a; Khoali *et al*., 2022]. Furthermore, OCL is adequate for imbalance scenarios, such as recommender systems [Fernández *et al*., 2018]. On the other hand, OCL is more challenging, as OCL only has interesting recommendations in the training step. Part of this challenge relates to the representation directly influencing the OCL [Gôlo *et al*., 2021b; Khoali *et al*., 2022; Gôlo *et al*., 2022]. Therefore, one-class recommendations have the challenge of representing the user, the item, and its iterations, a challenge already known in recommender systems [Wu *et al*., 2020a].

The gaps cover the non-recommendations scope and represent the user, item, and iterations on the OCL scenario, generally, are mitigated individually or by studies that apply one-class recommendation without graphs or by studies that use graph representations considering non-recommendations. In this way, we propose a Graph Neural Network (GNN) for link prediction to the one-class recommendation. GNNs are recently used for representation learning and have obtained state-of-the-art results, even more in the recommender systems literature [Wu *et al*., 2020a], but not in the one-class recommendation. Another novelty of our proposal is an unsupervised GNN for link prediction, which is little explored in the field of recommender systems. This GNN type is directed to the recommendation problem since the link prediction task predicts interactions between nodes, which are interactions between users and items in recommender systems, i.e., recommendations and non-recommendations [Li and Chen, 2013; Yang *et al.*, 2019]. In short, our proposed approach has the following contributions:

1. We model an enriched heterogeneous graph considering only interest class relations (recommendations) to improve the recommendation;
2. We propose an unsupervised GNN via link prediction task for recommender systems;
3. We recommend items considering only recommendations to train without having to cover the wide scope of non-recommendations from users;
4. We propose an explainability model for one-class recommendation through graphs; and
5. We perform a qualitative evaluation through the TSNE algorithm in our learned embeddings.

The extended version of this study builds upon the prior work [Gôlo *et al.*, 2022]. In this extension, we enhance the experimental evaluation by incorporating a more diverse range of datasets, such as Movies, Recipes, and Google recommendations. Additionally, we introduce novel techniques for model interpretability, employing one-class learning in conjunction with graph neural networks. Our investigation into the learned representations is further enriched through both visual inspection using TSNE and statistical significance analyses. We compared our proposal with four other strategies. Three are user and item representations considering the item's review and the graph structure. We use a state-of-the-art algorithm for text representation to represent the reviews. The last strategy is an end-to-end GNN. The results demonstrated that our proposal outperforms other methods to represent users and items in the OCL scenario and outperforms an end-to-end GNN.

We organized the remainder of this paper as follows. First, section 2 discusses related work under OCL and GNN via Link prediction for recommender systems. Second, section 3 presents the proposal to deal with a one-class recommendation. Third, Section 4 presents the explainability model for one-class learning and graphs. Fourth, section 5 presents the experimental evaluation, discussing the results obtained when applying the proposed representations to real-world data for recommender systems. Finally, Section 6 presents our concluding remarks and future work.

2 Related Work

One-class approaches have been used as a tool in recommender systems. [Pan *et al.*, 2008] propose two frameworks to deal with problems with one-class collaborative filtering. The authors explore the user-item matrix as a feature for one-class collaborative filtering. In addition, [Zhao *et al.*, 2015] explore the one-class recommendation through item and user representations generated by a personalized matrix factorization. Also, using one-class collaborative filtering, [He and McAuley, 2016] use matrix factorization and visual features represented by a Deep Convolutional Neural Network.

Addressing some common issues of one-class recommendation and commonly used strategies to tackle such issues, [Khoali *et al.*, 2022] propose a Bayesian personalized ranking based on a neural network using as input the user-item matrix as features. [Raziperchikolaei and Chung, 2022] aim to predict positively-related user-item pairs by training several state-of-art methods with only similar pairs, addressing the challenges of relying on dissimilar pairs by introducing two new terms to the objective functions. Dealing with users' one-class feedback problems, [Liu *et al.*, 2020] introduce rich interactions and exploits complementarity between generative and discriminative training, using generative adversarial networks to enhance the accuracy of modeling users' behaviors.

Using users' textual feedback, [Wan and McAuley, 2018] propose a ranking method to incorporate personalized textual information in implicit feedback settings. Considering only positive feedback data, the proposed method models texts in topics and incorporates it with implicit positive behavior feedback, such as purchases or check-ins. Also, in a one-class approach, [He *et al.*, 2022] propose a method to recommend dealing with heterogeneous user behavior, such as different types of interaction with an item. The authors propose a behavior attention layer to represent different behaviors of a user in the same inner structure, modeling the user's next behavior relationship with different historical interactions. This way, a task-specific layer can use modeled users' real behavior to predict the next interaction, such as purchasing or examining an item.

With a different one-class classification approach, [Li and Chen, 2013] convert a recommendation problem into a link prediction one using a generic kernel-based ML approach to map the transactions into a bipartite user-item graph, with the prediction model built from the one-class SVM algorithm. Graph representation approaches have been used in recommender systems [Li and Chen, 2013; Wu *et al.*, 2020a]. The study [Wu *et al.*, 2020a] is a survey of Graph Neural Networks (GNNs) in recommender systems. This study shows the extensive use of GNNs in this field. The authors highlighted the advantages of using heterogeneous graphs to represent relations and user-item iterations to obtain a robust representation of the user and the items. Furthermore, GNNs allow link prediction approaches that are adequate for the recommender systems problem [Wu *et al.*, 2021].

Considering GNN for link prediction in the recommender systems, we can cite studies such as [Zhang and Chen, 2018] that propose a novel method that uses GNN to predict edges in graphs, indicating the GNN for link prediction as promising in the recommender systems. Also, [Islam *et al.*, 2020] suggest that GNN for link prediction is an alternative to the recommender systems field. Furthermore, [Wu *et al.*, 2021] also cite recommender systems as promising applications for these GNNs.

As the above discussion shows, pioneers' one-class recommendation studies use traditional and non-enriched representations and do not explore graphs for recommendations modeling. On the other hand, previous one-class recommendation studies use enriched heterogeneous representations with feedback/reviews, however, without graph modeling and its advantages. Following a different path, [Li and Chen, 2013] explore one-class recommendations with a graph for link prediction. However, this study does not explore the GNNs. Studies that explore GNNs for link prediction indicate recommender systems as a promising application. However, using GNNs for link prediction in one-class recommendations is scarce. Thus, there exists a need for more studies on GNNs via link prediction for the one-class recommendation. Therefore, with the use and advantages of OCL, heterogeneous graph modeling, and GNN for link prediction to represent users and items, we propose the one-class recommendation through unsupervised graph neural networks via link prediction.

3 Unsupervised Graph Neural Networks via Link Prediction for One-Class Recommendation

We propose learning a more robust and adequate representation for the one-class recommendation considering the advantages of the graph structure. Thus, we separate the proposal into three steps: (i) create an enriched graph for the users, items, and metadata; (ii) learn representations through a graph neural network via link prediction task; and (iii) use the One-Class Support Vector Machines to perform the one-class recommendation. Figure 1 illustrates each proposal step. First, we model our recommender problem with a heterogeneous enriched graph. Second, we learn representations for the graph nodes with a GNN for link prediction. Finally, we use one-class learning to recommend or not an item for a user. The following sections explain each step.

3.1 Heterogeneous Enriched Graph Modeling

We model our user-item recommendation problem with graphs since graphs naturally model the recommendation problem. First, we create a heterogeneous graph. Then, we model the graph:

1. **User-Movie dataset**: five nodes (users, items, keywords, genre, and review) and four edge types (user-item, keyword-item, genre-item, and review-item);
2. **User-Recipe dataset**: four nodes (users, items, tags, and review) and four edge types (user-item, tag-item, review-user, and review-item);
3. **User-Google dataset**: four nodes (users, items, category, and review) and four edge types (user-item, category-item, review-user, and review-item);

The keywords represent the words related to the item, the genre is the movie genre, and the reviews are the reviews for the items. We add this metadata to enrich the graph and make it more connected. Finally, we model our graph as a one-class graph since we only add edge user-item of the interest, i.e., user evaluations for items with a 5 rating, with the intuition that learns a representation only with the set of interest labeled. After modeling the graph, we can learn a representation with graph neural networks considering the link prediction task.

3.2 Graph Neural Networks for Link Prediction

Before applying the graph neural network for link prediction, we adopted a strategy for all graph nodes aiming to reach representations to improve the representation learning of our Graph Neural Network. Thus, we use a graph regularization framework from [Rossi *et al.*, 2014; do Carmo and Marcacini, 2021]. Equation 1 defines regularization and has two terms. The first term determines that neighboring nodes in the graph have similar embedding vectors. The second term preserves the initial node representation according to a factor. Finally, our goal is to minimize the $Q(\boldsymbol{F})$:

$$Q(\boldsymbol{F}) = \frac{1}{2} \sum_{o_i,o_j \in O} w_{o_i,o_j} \Omega(\boldsymbol{f}_{o_i}, \boldsymbol{f}_{o_j}) + \mu \sum_{o_i \in O_e} \Omega(\boldsymbol{f}_{o_i}, \boldsymbol{k}_{o_i}) \quad (1)$$

in which, $\boldsymbol{F}$ are the graph node representations generated by the regularization, O is the set of all graph nodes, O_e is the set of graph nodes with initial representations, w_{o_i,o_j} indicates the weight of the connection between the nodes o_i and o_j, Ω is a distance function between embedding vectors, $\boldsymbol{f}_{o_i}$ is a generated embedding, $\boldsymbol{k}_{o_i}$ is an initial representation for o_i, and μ is a factor of preserving, in which $\mu > 0$.

After generating initial representations for all nodes, we will apply a Graph Neural Network (GNN). The GNN considers the structured representation of each node in the graph and the adjacency matrix $\boldsymbol{A}$ as input for learning the representations of the nodes. The initial representation of the nodes will be called $\boldsymbol{f}_{o_i} \in \boldsymbol{F}$. We denote $g(\boldsymbol{F}, \boldsymbol{A}; \boldsymbol{\mathcal{W}})$ to represent a GNN with trainable weights $\boldsymbol{\mathcal{W}} = \{\boldsymbol{W}^{(1)}, \cdots, \boldsymbol{W}^{(L)}\}$ in L hidden layers. For the l-th layer, the GNN propagation rule can be summarized as [Wu *et al.*, 2020b]:

$$\boldsymbol{H}^{l+1} = g(\boldsymbol{H}^{(l)}, \boldsymbol{A}; \boldsymbol{W}^{(l)}) \quad (2)$$

in which, $\boldsymbol{H}^{(l)}$ is the input to the l-th layer of the GNN, and $\boldsymbol{H}^{l+1}$ is the output of this layer. It is worth noting that the $\boldsymbol{F}$ representations are the input to the first layer, which is equivalent to $\boldsymbol{H}^{(0)}$. The embeddings learned for each object in the graph are represented by $\boldsymbol{H}^{(L)}$. Therefore, the information, i.e., characteristics present in the representations of adjacent nodes (neighbors), are aggregated through a neural network.

For unsupervised representation learning via GNN for link prediction, we pass the last layer $\boldsymbol{H}^{(L)}$ for a *LinkEmbedding* layer that predicts a link between nodes. Equation 3 defines this layer:

$$\boldsymbol{z} = \sigma \left(\sum_{o_i,o_j \in O} \boldsymbol{h}_{o_i}^{(L)} \cdot \boldsymbol{h}_{o_j}^{(L)} \right) \quad (3)$$

in which, $\boldsymbol{h}_{o_i}^{(l)}$ and $\boldsymbol{h}_{o_j}^{(l)}$ are the final generated embeddings from the neural network for the nodes o_i and o_j, σ is an activation function and $\boldsymbol{z}$ are the link predictions.

For this GNN for link prediction, we learn embeddings in an unsupervised way. First, we generate fake edges and use

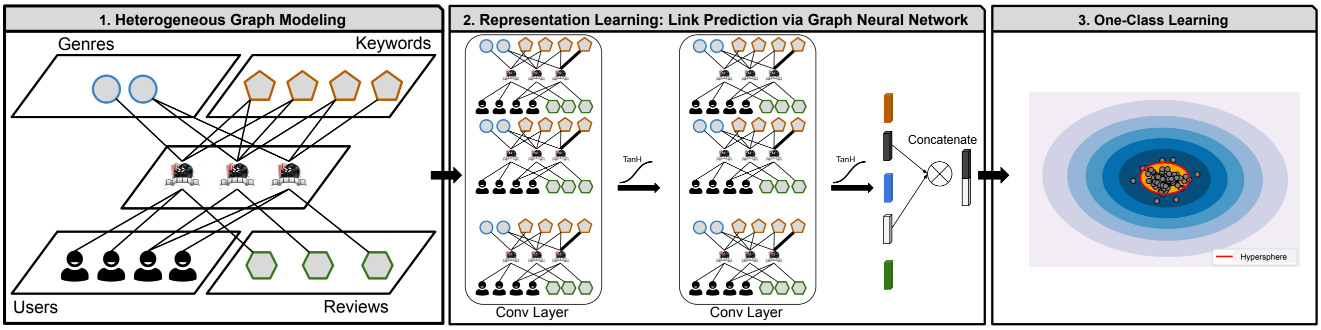

Figure 1. Proposed steps illustration.

original edges to train through Equations 2 and 3. Finally, we use binary accuracy as a loss function. We use the $\boldsymbol{H}^{(\boldsymbol{L})}$ layer to obtain the embedding nodes. After generating the user and item representations with the GNN, we use OCL to recommend or not an item for a user.

3.3 One-Class Learning

Given a user node o_i and an item o_j, we represent the user and item concatenating the representations, which generate a new representation $\boldsymbol{ui_q}$:

$$\boldsymbol{ui}_q = concatenation(\boldsymbol{h}_{o_i}^{(L)}, \boldsymbol{h}_{o_j}^{(L)}). \quad (4)$$

Given the $\boldsymbol{ui}_q$ representation, we can apply the One-Class Learning (OCL) in the 5 rating representations.

In OCL [Tax, 2001; Alam *et al.*, 2020], the training of the algorithms is only with examples of the interest class (rating 5), i.e., in the absence of counterexamples (other ratings). Therefore, in OCL for recommendations, the algorithm's objective is to inform if the item should be recommended for the user or not, learning only with data of items recommended for users. We use the One-Class Support Vector Machines (OCSVM) to perform the classification [Tax and Duin, 2004]. The OCSVM of [Tax and Duin, 2004] classifies a new instance belonging to the interest class if this example is inside a hypersphere. Formally, the center of the hypersphere [Tax and Duin, 2004] is:

$$\mu_{(\boldsymbol{c})} = \arg\min_{\boldsymbol{\mu} \in U} \max_{1 \leq q \leq m} \|\varphi(\boldsymbol{ui}_q) - \boldsymbol{\mu}\|^2, \quad (5)$$

in which $\boldsymbol{ui}_q$ is the representation for a user and item for a rating 5, m is the number of examples, U is the feature space associated with the function kernel φ, $\boldsymbol{\mu}_{(\boldsymbol{c})}$ is the center of the hypersphere in which the greater distance between $\varphi(\boldsymbol{ui}_q)$ to $\boldsymbol{\mu}_{(\boldsymbol{c})}$ is minimal and $\varphi(\boldsymbol{ui}_q)$ map $\boldsymbol{ui}_q$ into another feature space defined according to the kernel chosen. [Tax and Duin, 2004] define the hypersphere through the Equation 6:

$$\min_{\mu,\varphi,r} \quad r^2 + \frac{1}{m}\sum_{q=1}^{m} \frac{\varepsilon_{\boldsymbol{ui_q}}}{\nu} \quad (6)$$

subject to:

$$\|\varphi(\boldsymbol{ui}_q) - \boldsymbol{\mu}_{(\boldsymbol{c})}\|^2 \leq r^2 + \varepsilon_{\boldsymbol{ui_q}}, \forall i = 1, ..., m. \quad (7)$$

r is the radius of the hypersphere, $\varepsilon_{\boldsymbol{ui_q}}$ is the external distance between $\varphi(\boldsymbol{ui_q})$ and the surface of the hypersphere, and $\nu \in (0, 1]$ defines the smoothness level of the hypersphere volume.

4 Explainability for One-Class Recommendation and Graph Neural Networks

Studies use explainability in graph neural networks in the literature to explain the representation learning of these methods, for instance, explain which type of graph node is more important for the solved task [Yuan *et al.*, 2022]. We follow [Yuan *et al.*, 2022]'s taxonomy for our explainability, proposing an explainability method for unsupervised GNNs in the one-class learning scenario. Our method is at the level of instance-level explanations through perturbations.

The strategy of perturbation-based methods is to analyze the GNN output with perturbed GNN inputs [Yang *et al.*, 2019; Liu *et al.*, 2020]. In our case, we analyze the output of the OCSVM classification after perturbing its inputs, i.e., we perturb the representations generated by our unsupervised graph neural network. Since we divided our proposal into two steps, we also divided our explainability method into (i) perturbation of the representations generated by our enriched GNN; and (i) analyzing the OCSVM classification of the perturbed representation.

Explainability methods such as GNNExplaner [Yang *et al.*, 2019] mask edges and nodes to disturb the aggregations between node representations performed by the GNN representation learning during its training. In the same way, PG-Explaner [Liu *et al.*, 2020] masks the edges of the graph. We used a similar strategy in our perturbation step. We disturb the graph node representations by considering each node type. We perform the perturbation by subtracting the neighbor's representations of a given type from a target node representation. Our perturbation can be defined by Equation 8:

$$\hat{\boldsymbol{h}}_{o_i}^{(l)} = \sum_{o_j \in O_p} \boldsymbol{h}_{o_i}^{(l)} - \boldsymbol{h}_{o_j}^{(l)}, \quad (8)$$

in which, $\hat{\boldsymbol{h}}_{o_i}^{(l)}$ is the new target node representation and O_p is the target node neighbors set considering a single node type.

Studies on explainability for classification tasks generally use the probability of the instance belonging to one of the problem classes to estimate how much the perturbation influenced the classification of the instance [Ribeiro *et al.*, 2016].

To generate a score in the one-class learning scenario similar to the probability in the multi-class scenarios, we use the distance from the example to the hypersphere, in which, if the example is classified as belonging to the class of interest, the score has a positive sign, and if not, it has a negative sign [da Silva *et al.*, 2022; Gôlo *et al.*, 2023]. Figure 2 illustrates an example of the scores generated for two examples with distances 5 and 7 from the hypersphere. The green point has distance 7 from the hypersphere, while the blue point has distance 5 from the hypersphere. The green point is outside of the hypersphere. Thus we add a negative signal for the green point distance (-7). The blue point is inside of the hypersphere. Thus we add a positive signal for the blue point distance ($+5$). After perturbing the graph's node representations, sending the representations to the OCSVM, and receiving a distance value from that representation to the hypersphere, we can analyze how much each type of node influenced a certain recommendation considering the value generated through the strategy presented above.

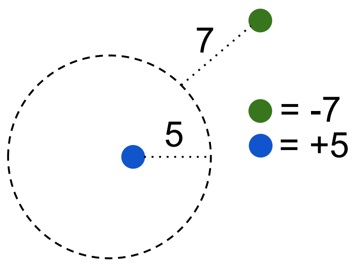

Figure 2. Illustration of the value generated through our explainability method.

5 Experimental Evaluation

In the experimental evaluation, we compare the representations generated by our proposal with three baselines in the one-class learning scenario for recommendations considering the OCSVM algorithm and an end-to-end GNN for link prediction baseline. We also compare the 2D TSNE projections of the representation methods and analyze our explainability model. Our research goal is to demonstrate that our proposal outperforms other methods used in the user and item representation and the recommendation and generates more robust representations. The following sections present the dataset, experimental settings, results, and discussion. All source codes and datasets are available.

5.1 Dataset

We use the recommendation dataset for movies from [Rana *et al.*, 2022]. The dataset contains 289853 ratings from users of movies. We use the ratings 5 (19668) and 1 (6624) to represent the recommendation and non-recommendation classes. Each movie has reviews, and the dataset with only ratings 1 and 5 contains 1915 users and 1612 movies. Furthermore, the dataset contains the IMDB movie id. Thus, we enriched the dataset with movie metadata to enrich our graph[1]. We collect the metadata from. We add the movie genre, keywords, and overview.

We also used the dataset from [Li, 2019], which contains 1132367 ratings and reviews from users of food Recipes online, with ratings re-scaled from $[0, 5]$ to $[1, 5]$. Recipes tags and description metadata are also present in the dataset, thus, allowing for its usage to enrich our graph. Also, to make our graph more connected, we kept only reviews from users within the top 1000 users with the most reviews posted and Recipes in the top 1000 most reviewed, with only ratings 5 and 1 kept for representing recommendations. We sample the 1132367 ratings to run experiments.

Finally, we used the dataset from [Yan *et al.*, 2022; Li *et al.*, 2022], containing 4838887 user reviews from places on Google. Reviews with ratings 1 and 5 were kept to represent recommendations, filtering only items with both user and item in the top 1000 most common. We enrich the graph using the metadata from descriptions and categories of places. We sample the 4838887 ratings to run experiments. Table 1 presents the details of the datasets.

Table 1. Details of the datasets considering the number of users, items, rating 5, and rating 1.

Datasets	Users	Items	Rating 5	Rating 1
Movies	1915	1612	19668	6624
Recipes	967	978	10000	634
Google	959	974	10000	1497
Average	1280	1188	13223	2918

5.2 Experimental Settings

We use three baselines based on the Bidirectional Encoder From Transformers (BERT) [Devlin *et al.*, 2019]. BERT is a pre-trained neural network that we use for text representation that generates embeddings to represent the text. The BERT model was trained in a large textual corpus that represents sentences based on their context and outperforms other natural language pre-processing models. In this way, BERT applies correlation techniques, compares the embedding, and extracts semantic and syntactic characteristics from the text [Otter *et al.*, 2020]. We represent the item reviews and descriptions with the BERT to generate the baselines. The first baseline is the BERT representation for the item (BERT-i). We use the review representation in the movie dataset since we have one review for each item and the description representation in the other datasets since we have more than one review for each item in these datasets. The second and third are the BERT representation for the movie concatenated with the user representation generated by the regularization for two graphs: the original graph with item-user relations (BERT-i-u-o) and our enriched graph (BERT-i-u-e).

[1]We collect the metadata from `https://www.kaggle.com/datasets/rounakbanik/the-movies-dataset?select=keywords.csv&page=2`.

For our graph modeling, the item node content is the item description, and the review node content is the reviews for the item. Both contents are in a text format. We represent these texts with the BERT embeddings. After representing these nodes, we apply the regularization for all nodes to have representations. Another method we used to compare with our proposal was the prediction of the GNN, i.e., in an End-To-End (ETE) way. We use our enriched graph to train the GNN-ETE.

For our proposals, GNN-enriched and GNN-original, we use the regularization for all nodes to have an initial representation in our enriched and original graph (only user-items relations). Then, finally, we obtain the embedding with the unsupervised GNN with the link prediction task. In the final step, we classify a recommendation or not through the OCSVM that has as input the concatenation for the user and item representations, except for BERT-i, which uses only the item representations, and GNN-ETE, which is end-to-end. The parameters for the representation methods and OCSVM were:

- **BERTs**: parameter free;
- **GNNs**: layer sizes = $\{64, 32\}$ and $\{32\}$, epochs = $\{200\}$, patience = $\{20, 50\}$, activation functions = $\{sigmoid, relu, tanh\}$, and learning rates = $\{1^{-3}, 1^{-4}\}$;
- **OCSVM**: kernel = $\{rbf, poly, sigmoid, linear\}$, $\nu = \{0.001, 0.005, 0.01, 0.05\}$ and $0.1 * \nu, \nu \in [1..7]$, and $\gamma = \frac{1}{n}\}$, in which n is the input dimension.

We use the procedure k-Fold Cross-Validation for One-Class Learning. In this procedure, we apply a k-Fold Cross-Validation considering only the interest class (rating 5) since, in the OCL, we have only interest examples labeled. The procedure consists of dividing the interest class into folds and using $k - 1$ folds to train and the remaining fold to test iteratively. We also add the not-interest set (rating 1) in the test set. We chose the $k = 5$ to maintain the test set with interest and non-interest sets containing similar sizes. In addition, we use as the test only user-item pairs that are in the train set. Finally, we use the Accuracy (Acc), f_1-score, Precision (P), and Recall (R) as evaluation measures:

$$Acc = \frac{tp + tn}{tp + tn + fp + tn}, \quad (9) \qquad f_1 = \frac{2 \cdot P \cdot R}{P + R}, \quad (10)$$

$$P = \frac{tp}{tp + fp}, \quad (11) \qquad R = \frac{tp}{tp + fn}, \quad (12)$$

in which tp (True Positives) is the number of ratings 5 that the OCSVM has correctly classified; tn (True Negatives) is the number of ratings 1 that the OCSVM has correctly classified; fp (False Positives) is the number of ratings 1 incorrectly classified; and fn (False Negatives) is the number of ratings 5 incorrectly classified;

5.3 Results and Discussion

Tables 2, 3, and 4 present the best results considering all parameters used considering the precision, recall, f_1-Score, and accuracy for the OCSVM with the representations generated from BERT and GNN variations and for the GNN-ETE. The best results are in bold font, and we tie equal values by the standard deviation values. Each value in the table is the average of all executions for each fold in the k-fold cross-validation.

BERT-i presented the lowest results, indicating that only the item representation is insufficient to recommend an item to a user. We reinforced this indicative when the BERT results with a representation of users after the regularization through the graph structure improved the recommendations (BERT-i-u-o and BERT-i-u-e). Another interesting point is the recommendation improvement when the regularization is performed on the enriched graph since BERT-i-u-e got the best results in relation to BERT-i-u-o in the Movies and Recipes dataset. On the other hand, the GNN-ETE presented the highest values of precision, recall, accuracy, and f_1-score in relation to the BERT methods with the OCSVM in the Movies and Google datasets.

In addition to obtaining better f_1-score and precision values than the baselines in the Recipes datasets, GNN-ETE obtained better results than our proposals. Furthermore, BERT-i-u-e obtains the highest values considering recall and accuracy in this dataset. Interestingly, these two methods are based on enriched graphs, i.e., enriching the graph improved the recommendation performance. In the other datasets, our proposals with the OCSVM outperform BERT methods and the GNN-ETE, considering precision, recall, accuracy, and f_1-score.

One advantage of our proposals besides the highest metrics values is the dimensionality reduction of the representation of the user and item based on the user's review of an item. Our representations have 32 dimensions and BERT 384. The results and advantages reinforce the representation learning relevance for the one-class recommendation. Considering our proposals, GNN-original performed better than GNN-enriched. However, the GNN-enriched performed better than all baselines and the GNN-original, considering recall and accuracy in the Google dataset.

In our intuition, the GNN-original is better since this GNN is based on link prediction trains with real and fake links only between users and items, i.e., with recommendations and not recommendations. On the other hand, the GNN-enriched trains with different types of edges and fake edges. Even with worse metric values than the GNN-original, the GNN-enriched has some advantages. First, the greater connectivity of the graph generates paths between users and items, which we can use in other recommender system tasks, such as user clustering based on graph relations. Second is the recommendation's explanatory power through the interactions of the different nodes that we explore with our expandability method for unsupervised graph neural networks in the one-class learning scenario.

We performed Friedman's statistical test with Nemenyi's post-test to compare the methods considering precision, recall, f_1-score, and accuracy in the three datasets [Trawinski *et al.*, 2012]. We show the test result in Figure 3, which presents a critical difference diagram. The diagram presents the methods' average rankings and the methods connected by a line do not present statistically significant differences. Baselines obtain the worsts rankings, GNN-ETE has a better ranking than the BERT, and our proposals obtain the best

Table 2. Precision, recall, F_1-Score and Accuracy for the OCSVM considering all representations methods and the end-to-end GNN. Bold fonts indicate the best values of the metric. These results are for the **Movies** dataset.

Methods/Metrics	precision	recall	f_1-score	accuracy
BERT-i	0.541±0.007	0.536±0.013	0.527±0.014	0.537±0.013
BERT-i-u-o	0.525±0.006	0.532±0.007	0.525±0.006	0.532±0.006
BERT-i-u-e	0.581±0.009	0.574±0.015	0.553±0.021	0.574±0.015
GNN-ETE	0.703±0.004	0.694±0.013	0.693±0.014	0.694±0.013
GNN-original	**0.729±0.001**	**0.730±0.001**	**0.729±0.001**	**0.730±0.001**
GNN-enriched	0.709±0.002	0.707±0.002	0.708±0.002	0.708±0.002

Table 3. Precision, recall, F_1-Score and Accuracy for the OCSVM considering all representations methods and the end-to-end GNN. Bold fonts indicate the best values of the metric. These results are for the **Recipes** dataset.

Methods/Metrics	precision	recall	f_1-score	accuracy
BERT-i	0.635±0.008	0.723±0.011	0.660±0.005	0.724±0.011
BERT-i-u-o	0.648±0.007	0.731±0.005	0.668±0.004	0.731±0.005
BERT-i-u-e	0.641±0.008	**0.735±0.006**	0.663±0.003	**0.735±0.006**
GNN-ETE	**0.664±0.008**	0.712±0.012	**0.679±0.004**	0.712±0.012
GNN-original	0.661±0.003	0.719±0.005	0.678±0.002	0.719±0.005
GNN-enriched	0.662±0.005	0.725±0.004	0.678±0.003	0.725±0.004

Table 4. Precision, recall, F_1-Score and Accuracy for the OCSVM considering all representations methods and the end-to-end GNN. Bold fonts indicate the best values of the metric. These results are for the **Google** dataset.

Methods/Metrics	precision	recall	f_1-score	accuracy
BERT-i	0.514±0.018	0.533±0.006	0.512±0.026	0.533±0.006
BERT-i-u-o	0.525±0.014	0.539±0.012	0.527±0.014	0.539±0.012
BERT-i-u-e	0.497±0.008	0.502±0.008	0.499±0.008	0.502±0.008
GNN-ETE	0.660±0.010	0.631±0.017	0.576±0.054	0.631±0.017
GNN-original	**0.669±0.005**	0.670±0.005	**0.669±0.005**	0.670±0.005
GNN-enriched	0.668±0.004	**0.670±0.004**	0.668±0.004	**0.670±0.004**

rankings. Our proposals are not statistically different from GNN-ETE. However, our proposals have statistically significant differences between the BERT methods.

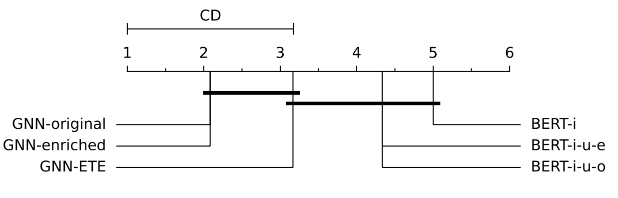

Figure 3. Critical difference diagram with the average rankings of the Friedman test with Nemenyi's post-test considering all metrics and datasets.

We applied our explainability method to the movie recommendation dataset. We chose a user and item pair correctly classified by our OCSVM as a recommendation. The second step was to generate the subgraph related to the two observed nodes. Third, we perturbed the representation of the item considering neighbors of a single node type, i.e., we perturbed with user, keywords, genres, and review nodes. This undisturbed instance has a score of $+13.77$. After perturbing the item node considering the user nodes connected with this item, we obtained a score of -764.31. Considering the keyword nodes, we obtained a score of -53.04, for genres, -65.53, and finally, for the review node $+7.12$. We present a visual result of our method showing the subgraph. Each node used to perturb the item representation has a size proportional to how much it influenced the recommendation. Thus, the user nodes were more important for the recommendation since they were the nodes that most modified the score when removed. Movies genre nodes and keywords also influenced but less significantly. The type of node that had the least influence was the review node.

We expected that user nodes would influence more because they are the most important and item nodes since they are the nodes used directly to perform recommendations. The keyword and genre nodes influenced the recommendation less than the user nodes but were also important as they changed the recommendation to non-recommendation and the user nodes. However, the review node was the least influential since even modifying the distance of the instance did not change the recommendation. This fact can be explained because the review node that disturbed the item's representation is the only one among the disturbed nodes that has an initial representation. Thus, this type of node presents less similar representations than the other types, which led to this node having a smaller influence on the recommendation, even though it is one of the most important nodes due to its function in regularization.

Figure 5 presents two-dimensional projections of the best representation model obtained by each method used in our experimental evaluation, considering the Movies datasets. We generated the projections using the algorithm t-Distributed

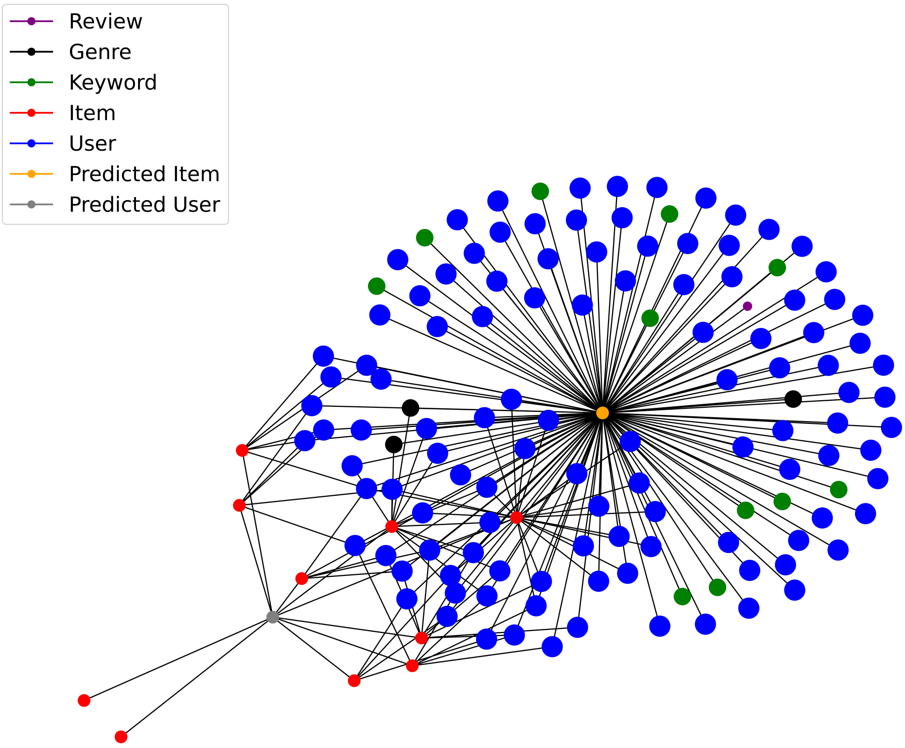

Figure 4. Subgraph generated by our explainability method for the movies dataset. The higher the node, the more important for predicting the item's recommendation (orange node) for the user (gray node), and the smaller, the less important.

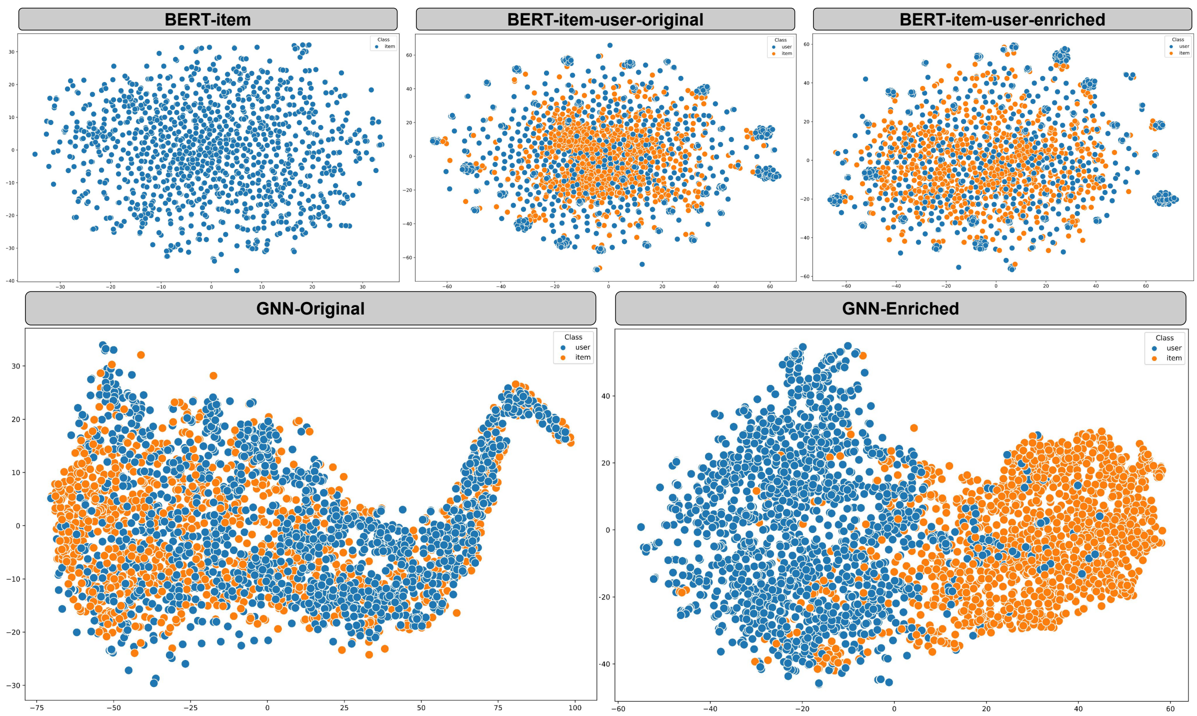

Figure 5. Two-dimensional projection (t-Distributed Stochastic Neighbor Embedding) of BERT-item, BERT-item-user-original, BERT-item-user-enriched, GNN-original and GNN-enriched, in the one-class recommendation scenario. The colors indicate users (blue) and items (orange). Our proposal obtained the best visual result since was able to separate users and items in different regions.

Stochastic Neighbor Embedding (t-SNE) [Van der Maaten and Hinton, 2008]. Representations generated by BERT-item are difficult to analyze since we have only item representations. BERT-item-user-original and BERT-item-user-enriched presented distributed hubs of users and distributed items in different regions. Even obtaining the best results in the movies dataset, the GNN-Original generates distributed representations for items and users in the same area. On the other hand, GNN-Enriched generates the best visual result since TSNE separates the items and users in different regions.

Another important point that our GNN-enriched can contribute is the exploration of other aspects of the recommendation, such as coverage, novelty, and diversity, since the graph can be enriched with appropriate information to cover these aspects. In addition, these aspects may interest the user, which is advantageous in carrying out the recommendation.

6 Conclusions and Future Work

This study presented an extension for the one-class recommendation through graph neural networks. We evaluate our methods in more datasets, with qualitative and statistical analysis and a new explainability model for this scenario. Our proposal has the advantages: (i) heterogeneous and enriched modeling that is useful to recommender systems; (ii) easy extension to different types of heterogeneous graphs such as the original or enriched proposed in this study; (iii) representation learning to obtain more robust representations for one class recommendation; (iv) generate a more appropriate representation for items and users; and (v) statistical difference significance to baselines. Furthermore, our expandability method is easy to extend for other works that combine graphs and one-class learning. Finally, our method is applicable in other graph modeling contexts where one of the edges or node types is of interest.

Our proposal was limited to a concatenation for the user and item representation, a two-step proposal (representation plus OCSVM), and a link prediction proposal that learns representations by predicting all links in the graph. Therefore, in future work, we will create an OCL end-to-end Heterogeneous GNN for link prediction that uses only user-item relations to predict links while using all edges to learn embeddings, thus covering our three main limitations.

Funding

This work was supported by *Coordenação de Aperfeiçoamento de Pessoal de Nível Superior* (CAPES) grant number 88887.671481/2022-00. Also, this work was supported by LATAM Google Ph.D. Fellowship. The authors of this work would like to thank the Center for Artificial Intelligence (C4AI-USP) and the support from the São Paulo Research Foundation (FAPESP grant 2019/07665-4) and the IBM Corporation.

Competing interests

The authors declare that they do not have competing interests.

Availability of data and materials

The datasets analyzed, codes and complete results are available in `https://github.com/GoloMarcos/JIDM-One-Class-Recommendation.git`.

References

Alam, S., Sonbhadra, S. K., Agarwal, S., and Nagabhushan, P. (2020). One-class support vector classifiers: A survey. *Knowledge-Based Systems*, 196:1–19. DOI: https://doi.org/10.1016/j.knosys.2020.105754.

da Silva, A. C. M., Gôlo, M., and Marcacini, R. (2022). Unsupervised heterogeneous graph neural network for hit song prediction through one class learning. In *Proceedings of the Symposium on Knowledge Discovery, Mining and Learning (KDMiLe)*, pages 202–209, Porto Alegre, RS, Brasil. SBC. DOI: 10.5753/kdmile.2022.227954.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186, Minnesota. Association for Computational Linguistics. DOI: 10.18653/v1/N19-1423.

do Carmo, P. and Marcacini, R. (2021). Embedding propagation over heterogeneous event networks for link prediction. In *Proceedings of the International Conference on Big Data (Big Data)*, pages 4812–4821, online. IEEE. DOI: 10.1109/BigData52589.2021.9671645.

Fernández, A., García, S., Galar, M., Prati, R. C., Krawczyk, B., and Herrera, F. (2018). *Learning from imbalanced data sets*, volume 11. Springer, Switzerland. DOI: https://doi.org/10.1007/978-3-319-98074-4.

Gôlo, M., Caravanti, M., Rossi, R., Rezende, S., Nogueira, B., and Marcacini, R. (2021a). Learning textual representations from multiple modalities to detect fake news through one-class learning. In *Proceedings of the Brazilian Symposium on Multimedia and the Web*, pages 197–204, Belo Horizonte, MG, Brazil. ACM. DOI: https://doi.org/10.1145/3470482.3479634.

Gôlo, M. P., Araújo, A. F., Rossi, R. G., and Marcacini, R. M. (2022). Detecting relevant app reviews for software evolution and maintenance through multimodal one-class learning. *Information and Software Technology*, 151:106998. DOI: https://doi.org/10.1016/j.infsof.2022.106998.

Gôlo, M. P. S., de Souza, M. C., Rossi, R. G., Rezende, S. O., Nogueira, B. M., and Marcacini, R. M. (2023). One-class learning for fake news detection through multimodal variational autoencoders. *Engineering Applications of Artificial Intelligence*, 122:106088. DOI: https://doi.org/10.1016/j.engappai.2023.106088.

Gôlo, M. P. S., Rossi, R. G., and Marcacini, R. M. (2021b). Learning to sense from events via semantic variational autoencoder. *Plos One*, 16(12):e0260701. DOI: https://doi.org/10.1371/journal.pone.0260701.

Gôlo, M., Moraes, L., Goularte, R., and Marcacini, R. (2022). One-class recommendation through unsupervised graph neural networks for link prediction. In *Proceedings of the Symposium on Knowledge Discovery, Mining and Learning (KDMiLe)*, pages 146–153, Porto Alegre, RS, Brasil. SBC. DOI: 10.5753/kdmile.2022.227810.

He, M., Pan, W., and Ming, Z. (2022). Bar: Behavior-aware recommendation for sequential heterogeneous one-class collaborative filtering. *Information Sciences*, 608:881–899. DOI: https://doi.org/10.1016/j.ins.2022.06.084.

He, R. and McAuley, J. (2016). Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *Proceedings of the International Conference on World Wide Web*, pages 507–517, Republic and Canton of Geneva, Switzerland. ACM. DOI:

https://doi.org/10.1145/2872427.2883037.

Islam, K., Aridhi, S., and Smaïl-Tabbone, M. (2020). A comparative study of similarity-based and gnn-based link prediction approaches. In *Graph Embedding and Mining Workshop*, Ghent, Belgium. HAL. DOI: https://hal.science/hal-03473689/.

Khoali, M., Laaziz, Y., Tali, A., and Salaudeen, H. (2022). A survey of one class e-commerce recommendation system techniques. *Electronics*, 11(6):878. DOI: https://doi.org/10.3390/electronics11060878.

Li, J., Shang, J., and McAuley, J. (2022). Uctopic: Unsupervised contrastive learning for phrase representations and topic mining. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 6159–6169. DOI: 10.18653/v1/2022.acl-long.426.

Li, S. (2019). Food.com recipes and interactions.

Li, X. and Chen, H. (2013). Recommendation as link prediction in bipartite graphs: A graph kernel-based machine learning approach. *Decision Support Systems*, 54(2):880–890. DOI: https://doi.org/10.1016/j.dss.2012.09.019.

Liu, J., Pan, W., and Ming, Z. (2020). Cofigan: Collaborative filtering by generative and discriminative training for one-class recommendation. *Knowledge-Based Systems*, 191:105255. DOI: https://doi.org/10.1016/j.knosys.2019.105255.

Otter, D., Medina, J., and Kalita, J. (2020). A survey of the usages of deep learning for natural language processing. *IEEE Transactions on Neural Networks and Learning Systems*, 32(2):604–624. DOI: 10.1109/TNNLS.2020.2979670.

Pan, R., Zhou, Y., Cao, B., Liu, N. N., Lukose, R., Scholz, M., and Yang, Q. (2008). One-class collaborative filtering. In *Proceedings of the International Conference on Data Mining*, pages 502–511, Pisa, Italy. IEEE. DOI: 10.1109/ICDM.2008.16.

Rana, A., D'Addio, R. M., Manzato, M. G., and Bridge, D. (2022). Extended recommendation-by-explanation. *User Modeling and User-Adapted Interaction*, 32(1):91–131. DOI: https://doi.org/10.1007/s11257-021-09317-4.

Raziperchikolaei, R. and Chung, Y.-j. (2022). One-class recommendation systems with the hinge pairwise distance loss and orthogonal representations. *arXiv preprint arXiv:2208.14594*. DOI: https://doi.org/10.48550/arXiv.2208.14594.

Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). "why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining*, pages 1135–1144, San Francisco, CA, USA. DOI: https://doi.org/10.1145/2939672.2939778.

Rossi, R. G., Lopes, A. A., and Rezende, S. O. (2014). A parameter-free label propagation algorithm using bipartite heterogeneous networks for text classification. In *Proceedings of the Annual Symposium on Applied Computing*, pages 79–84, Gyeongju Republic of Korea. ACM. DOI: https://doi.org/10.1145/2554850.2554901.

Ru, S., Zhang, B., Jie, Y., Zhang, C., Wei, L., and Gu, C. (2021). Graph neural networks for privacy-preserving recommendation with secure hardware. In *Proceedings of the International Conference on Networking and Network Applications (NaNA)*, pages 395–400, Lijiang City, China. IEEE. DOI: 10.1109/NaNA53684.2021.00075.

Tax, D. and Duin, R. (2004). Support vector data description. *Machine Learning*, 54(1):45–66. DOI: https://doi.org/10.1023/B:MACH.0000008084.60811.49.

Tax, D. M. J. (2001). *One-class classification: concept-learning in the absence of counter-examples*. PhD thesis, Delft University of Technology.

Trawinski, B., Smetek, M., Telec, Z., and Lasota, T. (2012). Nonparametric statistical analysis for multiple comparison of machine learning regression algorithms. *Applied Mathematics and Computer Science*, 22(4):867–881. DOI: https://doi.org/10.2478/v10006-012-0064-z.

Van der Maaten, L. and Hinton, G. (2008). Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(11):2579–2605. DOI: http://jmlr.org/papers/v9/vandermaaten08a.html.

Wan, M. and McAuley, J. (2018). One-class recommendation with asymmetric textual feedback. In *Proceedings of the SIAM International Conference on Data Mining*, pages 648–656. SIAM. DOI: https://doi.org/10.1137/1.9781611975321.73.

Wu, S., Sun, F., Zhang, W., Xie, X., and Cui, B. (2020a). Graph neural networks in recommender systems: a survey. *ACM Computing Surveys (CSUR)*, 55(97):1–37. DOI: https://doi.org/10.1145/3535101.

Wu, W., Li, B., Luo, C., and Nejdl, W. (2021). Hashing-accelerated graph neural networks for link prediction. In *Proceedings of the Web Conference*, pages 2910–2920, Ljubljana Slovenia. ACM. DOI: https://doi.org/10.1145/3442381.3449884.

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., and Philip, S. Y. (2020b). A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32(1):4–24. DOI: 10.1109/TNNLS.2020.2978386.

Yan, A., He, Z., Li, J., Zhang, T., and McAuley, J. (2022). Personalized showcases: Generating multi-modal explanations for recommendations. *arXiv preprint arXiv:2207.00422*. DOI: https://doi.org/10.48550/arXiv.2207.00422.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., and Le, Q. V. (2019). Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in Neural Information Processing Systems*, 32:5753–5763. DOI: ISBN: 9781713807933.

Yuan, H., Yu, H., Gui, S., and Ji, S. (2022). Explainability in graph neural networks: A taxonomic survey. *Transactions on Pattern Analysis and Machine Intelligence*. DOI: 10.1109/TPAMI.2022.3204236.

Zhang, M. and Chen, Y. (2018). Link prediction based on graph neural networks. *Advances in Neural Information Processing Systems*, 31:11. DOI: ISBN: 9781510884472.

Zhao, T., McAuley, J., and King, I. (2015). Improving latent factor models via personalized feature projection for one class recommendation. In *Proceedings of the International Conference on Information and Knowledge Management*, pages 821–830, Melbourne, Australia. ACM.

DOI: https://doi.org/10.1145/2806416.2806511.