# Recommendation System using Graph Neural Networks

**Mughees Asif** (180288337)
Supervisor: **Dr Angadh Nanjangud**
MSc. Artificial Intelligence
*School of Electronic Engineering and Computer Science*
*Queen Mary, University of London*

***Abstract*—Recently, neural networks have been used for developing recommendation systems that can parse graph-like data structures to develop meaningful representations of user-item relationships. Additionally, recommendation systems have been developed with Graph Neural Networks (GNNs) to expedite the aggregation process of macro (e.g., topological structure) and micro (e.g., node information) operations, and therefore enhance the overall information filtering capabilities of the system. However, the representation learning process is non-linear as social relationships combined with item interactions, both, need to be considered for optimal results. This research project aimed to address this by proposing a recommendation system that is capable of using the underlying social connections between users and items. The system was also split into three variations where several metrics were used to draw comparisons with published academic recommendation systems. The training of the models was done by using two real-world datasets that contain user-to-user and user-to-item information. The results show the system performing with equal efficiency as the sourced academic models, and also highlights the suitability of the system for recommendation tasks.**

***Index Terms*—Artificial intelligence, Recommendation system, Graph Neural Networks, Social Networks**

## I. Introduction

Recommendation systems are a vital tool incorporated into every business model to streamline user experiences and mitigate information overload by pinpointing areas of interest. Indeed, the business model of global organisations including Amazon, Google, Netflix, and Spotify relies on propagating items/services of interest to users, not only to increase customer satisfaction, maintain market dominance, and business profitability but to also simplify the addition of new services that would be pertinent to a specific user base.

Graph Neural Networks (GNNs) leverage deep learning methodologies on non-Euclidean data structures for node classification and link estimation [1]. Graphs are fundamental data structures consisting of nodes connected by edges, where each node can be denoted as a set of objects and the edges are defined as the relationship between the nodes. The information contained in the nodes and edges can be used by GNNs in recommendation systems to model relationships based on user-to-user and/or user-to-item interactions. Recently, the representation of these relationships with the expressive power of graphs has been receiving wide attention in research to enable predictive modelling through recommendation systems [2]–[4]. The process of gathering information represented in a graph-like data structure is underpinned by the iterative aggregation of features from local neighbourhoods which demonstrates the fidelity between representation learning and GNNs [5, 6].

This research project aims to develop, explore, and test a GNN framework for recommendation tasks by incorporating social information and item interactions from two real-world datasets. The remaining project is structured into several sections: Section II details the preliminaries for GNNs techniques and recommendation systems. Section III explains the architecture of the final framework and the datasets used for training and testing. Section IV and V highlight the achieved results with accompanying empirical analysis and suggested future work, and finally, Section VI delivers a conclusion to the research project.

## II. Background

### A. Graph Neural Networks (GNNs)

GNNs have been popularised recently by the advancement made in performances related to many tasks including predicting protein structure [7, 8], analysing dynamic systems [9], and predicting knowledge through base completion [10]. A graph $\mathcal{G}$ is represented by the notation $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is representative of the set of available nodes and $\mathcal{E}$ is the set of edges. Furthermore, $v_i \in \mathcal{V}$ is a node with an edge $e_{ij} = (v_i, v_j) \in \mathcal{E}$ extending from $v_j$ to $v_i$, and the local neighbourhood of the node $v$ can be denoted as $\mathcal{N}(v) = \{u \in \mathcal{V} | (v, u) \in \mathcal{E}\}$ [6]. There are two main categories of graphs:

1) **Directed/undirected graphs**: A directed graph consists of edges that only point in one direction, while an undirected graph consists of edges that can point in both directions.
2) **Homogeneous/heterogeneous graphs**: Homogeneous graphs consist of one type of nodes and edges, while heterogeneous graphs can have different types of nodes and edges.

The main intuition behind GNNs is the iterative aggregation of the feature information from a neighbourhood of nodes that is updated with the information from a current node during the propagation process [11]. The architecture of GNNs is based on stacking several layers that allow propagation through aggregation and update operations [6]:

$$\text{Aggregation}: \quad \mathbf{n}_v = \text{Aggregator}_l\left(\left\{\mathbf{h}_u^l, \forall u \in \mathcal{N}_v\right\}\right) \tag{1}$$

$$\text{Update}: \quad \mathbf{h}_v^{(l+1)} = \text{Updater}_l\left(\left\{\mathbf{h}_v^{(l)}, \mathbf{n}_v^{(l)}\right\}\right) \tag{2}$$

where:

$\mathbf{h}_u^l$ = represents the node at the $l^{th}$ layer
$\text{Aggregator}_l$ = aggregation operation
$\text{Updater}_l$ = update operation
$\mathcal{N}(v)$ = neighbourhood of the node

The aggregation step (Eq. 1) enables equal treatment of each neighbour with a mean-pooling operation or differentiating between the importance of the neighbours through an attention framework [12, 13]. The update step (Eq. 2) is "... the representation of the central node and the aggregated neighbourhood [that] will be integrated into the updated representation of the [current] node" [6]. Various integration techniques exist to alleviate misrepresentation between the aggregator and updater operations including gated mechanisms and applying non-linear transformations [12, 14].

### B. Recommendation Systems

Recommendation systems learn the preferences of a user by analysing the user-to-item interactions or other static features that highlight the interest of the user in a particular item. The task can be defined as [6]:

$$y_{u,i} = f(h_u^*, h_i^*) \tag{3}$$

where:

$i \in I$ = item in the directory
$h_u^*$ = user representation
$h_i^*$ = item representation
$f(\cdot)$ = score function
$y_{u,i}$ = preference score for a user on an item

Using GNNs in recommendation systems resulted in an increase in the precision of the predictions as incorporating social information from the underlying social connections of the user helped to boost performance [5, 6, 15, 16]. The social recommendation framework assumes that the users' social relationships present a novel approach to enhancing user representations by leveraging the social influence theory that connected people influence each other [17]. The social network can be integrated into a user-item graph as a unified graph where the social information and the aggregation operation are combined during the propagation process.

## III. Methodology

### A. Problem formulation

Consider a system comprised of sets of users $U = \{u_1, u_2, \ldots, u_n\}$ and items $V = \{v_1, v_2, \ldots, v_m\}$, where $\mathbf{R} \in \mathbb{R}^{n \times m}$ is the user-item graph represented as a rating matrix. If a rating is passed from $u_i$ to $v_j$, $r_{ij}$ equals the rating score, and conversely for an unknown rating $r_{ij} = 0$. Consider also $\mathcal{O} = \{\langle u_i, v_j \rangle \,|\, r_{ij} \neq 0\}$ as the set of observed/known ratings, and $\mathcal{T} = \{\langle u_i, v_j \rangle \,|\, r_{ij} = 0\}$ as the set of unknown ratings. Additionally, users can create social relations, where $\mathbf{T} \in \mathbb{R}^{n \times n}$ highlights the user-user social graph and is given a value of 0 or 1 dependent on if $u_j$ has a relation to $u_i$. Therefore, in combination with a user-item graph $\mathbf{R}$ and $\mathbf{T}$, the aim of the project is to predict the missing value in $\mathbf{R}$.

### B. User modelling

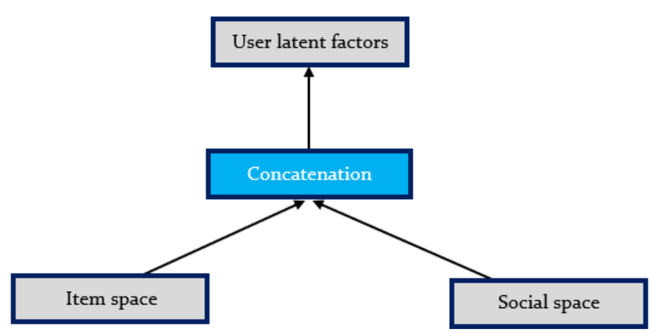


Fig. 1. User modelling

TABLE I
MATHEMATICAL NOTATIONS

| Symbol | Description |
|---|---|
| $r_{ij}$ | Rating values for the user $u_i$ and the item $v_j$ |
| $\mathrm{p}_i$ | Embedding of the user $u_i$ |
| $\mathrm{q}_j$ | Embedding of the item $v_j$ |
| $\mathrm{e}_r$ | Opinion embedding of the rating level $r$ i.e., 5-star rating, $r \in 1, 2, 3, 4, 5$ |
| $d$ | Length of the embedding vector |
| $C(i)$ | Set of items the user interacted with |
| $N(i)$ | Set of social friends that the user connected with |
| $B(j)$ | Set of users that interacted with item $v_j$ |
| $\mathbf{h}_i^I$ | Item-space user latent factor |
| $\mathbf{h}_i^S$ | Social-space user latent factor |
| $\mathbf{h}_i$ | User latent factor |
| $\mathbf{z}_j$ | Item latent factor |
| $\mathbf{x}_{ia}$ | Opinion-aware interaction representation of item $v_a$ for user $u_i$ |
| $\mathbf{f}_i^I$ | Opinion-aware interaction representation of user $u_t$ for item $v_j$ |
| $\alpha_{ia}$ | Item attention of item $v_a$ |
| $\mu_{jt}$ | User attention of user $u_t$ |
| $\beta_{io}$ | Social attention of the neighbouring user $u_o$ |
| $r'_{ij}$ | Predicting rating of the item $v_j$ by user $u_i$ |
| $\mathbf{T}$ | User-user social graph |
| $\mathbf{R}$ | User-item rating graph |
| $\oplus$ | concatenation of two vectors |

The user modelling operation aims to learn the latent factors $\mathbf{h}_i \in \mathbb{R}^d$ of users $u_i$. Highlighted by Fig. 1, the operation requires the concatenation of two latent factors to obtain the holistic user latent factors $\mathbf{h}_i$: item space user latent factor $\mathbf{h}_i^I \in \mathbb{R}^d$ from the user-item graph, and a social space user latent factor $\mathbf{h}_i^S \in \mathbb{R}^d$ from the social graph. A Multi-Layer Perceptron (MLP) is introduced to concatenate the two vectors, resulting in[1]:

$$\mathbf{h}_i = \sigma(W_l \cdot c_{l-1} + b_i) \tag{4}$$

where, $c_l = \left[\mathbf{h}_i^I \oplus \mathbf{h}_i^S\right]$.

*1) Item space:* The item space operation utilises the interactions between the users and items and also the users' preferences regarding the item, all encoded as a user-item graph. As shown in Fig. 2, the main premise is to learn item-space user latent factor $\mathbf{h}_i^I$. This can

[1]An accessible side-note on MLP can be found here.

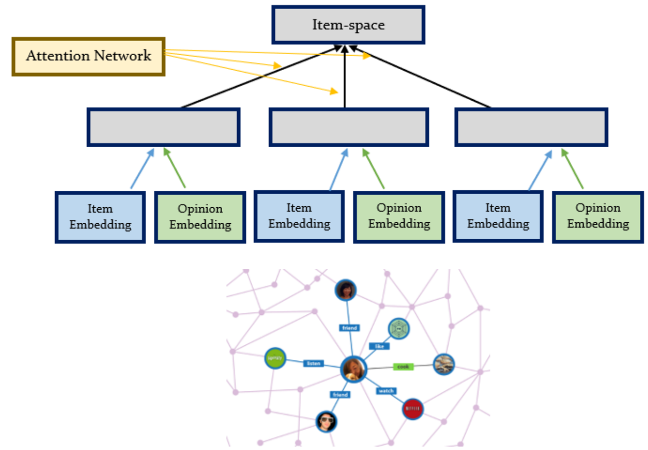


Fig. 2. Item space operation

be defined in the classic $y = mx + c$ equivalent linear function as:

$$\mathbf{h}_i^I = \sigma\left(\mathbf{W} \cdot A_{\text{item}}\left(\mathbf{x}_{ia}, \forall a \in C(i)\right) + \boldsymbol{b}\right) \tag{5}$$

where:

$\sigma$ = rectified linear unit function
$\mathbf{W}$ = weights of the network
$\boldsymbol{b}$ = bias of the network
$A_{\text{item}}$ = the aggregation operation
$C(i)$ = the items the user interacted with
$\mathbf{x}_{ia}$ = representation vector includes the users' opinion

The representation vector $\mathbf{x}_{ia}$ contains the rating $r$ a user gives an item from a 5-star rating i.e., $r \in 1, 2, 3, 4, 5$ or simply a densely embedded vector $\mathbf{e}_r$[2]. This mapping can be used to discern the users preferences and return the user-to-item relationship latent factors. This is achieved by using the direct item embedding $\mathbf{q}_a$, and retrieving the opinion embedding $\mathbf{e}_r$ using an MLP. The output is the representation vector that includes the users' opinion on a certain item $\mathbf{x}_{ia}$:

$$\mathbf{x}_{ia} = g_v\left(\mathbf{q}_a \oplus \mathbf{e}_r\right) \tag{6}$$

where:

$g_v$ = the MLP

The aggregation operation is the element-wise mean of the vectors $\mathbf{x}_{ia}$ and $\forall a \in C(i)$ by using a linear approximation of a localised convolution [5]. However, this produces sub-standard results as each interaction can

[2]An informative and succinct explanation of dense layers can be found here.

not have the same weights due to different users expressing individual preferences. To counteract this linearity, a 2-layer attention mechanism intervenes, where each interaction is given an individual weight dependent on a user's interest in the item. Substituting into Eq. 5:

$$\mathbf{h}_i^I = \sigma\left(\mathbf{W} \cdot \left\{\sum_{a \in C(i)} \alpha_{ia}\mathbf{x}_{ia}\right\} + \boldsymbol{b}\right) \tag{7}$$

where, $\alpha_{ia}$ is representative of the interaction between the user $u_i$ and the item $v_a$.

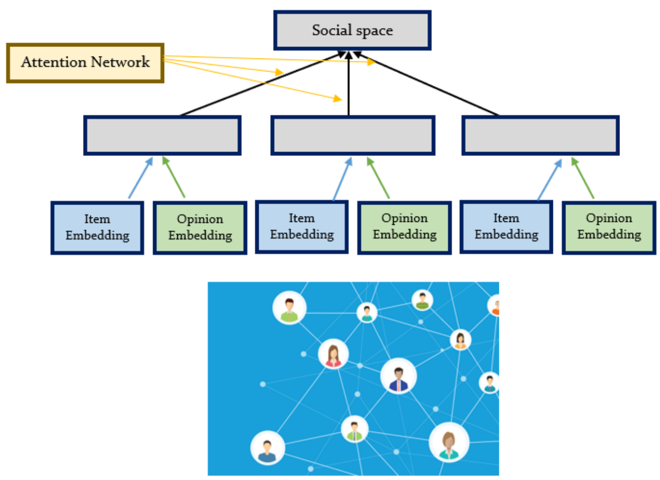


Fig. 3. Social space operation

*2) Social space:* Well-established social correlation theories have highlighted the impact of a user's social relationships on user preferences, where a specific category of items appeals to a specific social group, the user tends to associate with [18, 19]. This highlights the influence of the social connections on a user's decision-making process. To encode heterogeneous strengths of social relations into the GNN, an attention mechanism is introduced "... to select social friends that [can] characterise users' social information and then aggregate [the] information" [5]. This is achieved by incorporating social space latent factors $\mathbf{h}_i^S$ that are an aggregation of the neighbouring users item space $A_{\text{neighbours}}(\cdot)$, as defined in Section III-B1 and displayed in Fig. 3:

$$\mathbf{h}_i^S = \sigma\left(\mathbf{W} \cdot A_{\text{neighbours}}\left(\mathbf{h}_o^I, \forall o \in N(i)\right) + \boldsymbol{b}\right) \tag{8}$$

The aggregation operation on the user's neighbours follows a similar structure to the item space aggregation operation where a mean operator takes the element-wise mean of the vectors [5]. To diminish the impact of assuming that all neighbours contribute equally, an attention mechanism using a 2-layer neural network is introduced that develops the correlation between the user-to-user relationships and user-to-item interactions:

$$\mathbf{h}_i^S = \sigma\left(\mathbf{W} \cdot \left\{\sum_{o \in N(i)} \beta_{io}\mathbf{h}_o^I\right\} + \boldsymbol{b}\right) \tag{9}$$

where, $\beta_{io}$ is representative of the interaction between the user's social circle $u_i$ and the item $v_a$.

## C. *Item modelling*

This section highlights the incorporation of the item latent factors $\mathbf{z}_j$ by using information from the user-to-item graph as shown in Fig. 4. For each item, the user's preferences are aggregated i.e., the mean of the ratings for all items $(R = r \in 1, 2, 3, 4, 5)$, and using a MLP, the two vectors holding information regarding plain user embedding $\mathbf{p}_t$ and opinion embedding $\mathbf{e}_r$ are concatenated to develop a final user representation $\mathbf{f}_{jt}$:

$$\mathbf{f}_{jt} = g_u(\mathbf{p}_t \oplus \mathbf{e}_r) \tag{10}$$

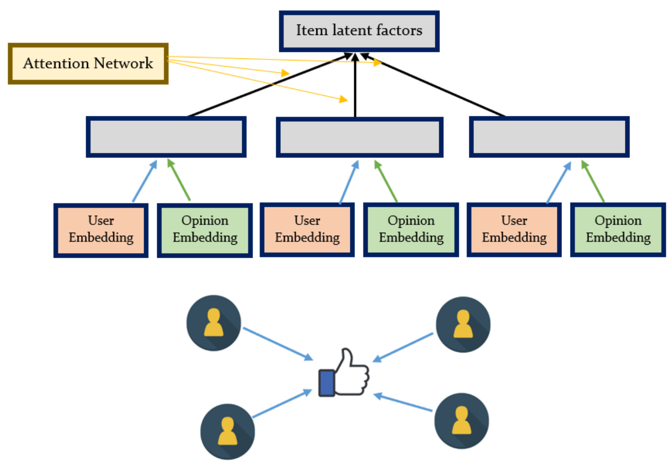


Fig. 4. Item modelling

The latent factors are derived similarly to the previous two sections by the introduction of an attention mechanism:

$$\mathbf{z}_j = \sigma\left(\mathbf{W} \cdot A_{\text{users}}\left(\mathbf{f}_{jt}, \forall t \in B(j)\right) + \boldsymbol{b}\right) \tag{11}$$

$$\mathbf{z}_j = \sigma\left(\mathbf{W} \cdot \left\{\sum_{t \in B(j)} \mu_{jt}\mathbf{f}_{jt}\right\} + \boldsymbol{b}\right) \tag{12}$$

where $\mu_{jt}$ "... captures heterogeneous influence from user-item interactions on learning item latent factor" [5].

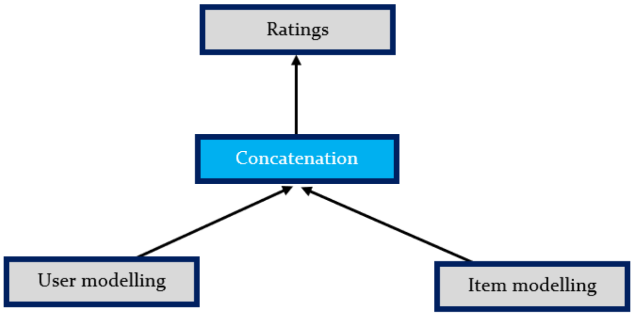


Fig. 5. Ratings

### D. Ratings

Lastly, the rating predictions are obtained as displayed in Fig. 5. The latent factors gained from the previous sections are concatenated using an MLP, $\mathbf{h}_i \oplus \mathbf{z}_j$, and passed through to get the final rating predictions $r_{ij}^{'}$:

$$r_{ij}^{'} = \mathbf{w}^T \cdot \boldsymbol{g}_{l-1} \tag{13}$$

### E. Objective function

The objective/loss function was an optimised version of the vanilla gradient descent algorithm, in the form of Root Mean Squared Propagation (RMSProp). The premise of RMSProp is to establish an adaptive learning rate that responds to the gradient of the learning function by storing and using the moving average of the squared gradients for each weight initialisation [20]. The mathematical notation follows as:

$$\text{Loss} = \frac{1}{2|\mathcal{O}|} \sum_{i,j \in \mathcal{O}} \left( r_{ij}^{'} - r_{ij} \right)^2 \tag{14}$$

where, $\mathcal{O}$ is the observable ratings, and $r_{i,j}$ is the ground truth as expressed by the user.

### F. Datasets

Two datasets from prominent product review websites were used: Epinions and Caio[3]. The contents of the datasets contained several data points including information on allowing the user to add friends, product names, varieties and the rating score. Table II shows the statistics of the datasets and also highlights the densely embedded user-to-user and user-to-item information that was used to train and test the GNN model in discerning suitable recommendations.

[3]Available here.

TABLE II
DATASET STATISTICS

| Characteristic | Epinions | Caio |
|---|---|---|
| Users | $7,000$ | $18,000$ |
| Items | $100,000$ | $250,000$ |
| Ratings | $7,000$ | $18,000$ |

### G. Training

The model was built using the `PyTorch` library in the Python programming language ecosystem, as per industry standard. The three embeddings (user, item and opinion) were stochastically initialised to allow the model to learn concurrently. For training purposes and adhering to a finite computational cost, dropout was added to the network that neglected specific weights which did not add sufficient value to the overall result i.e, anomalous and minute updates. Each component of the system contained a maximum of three hidden layers (same size as the embedding size) with the ReLU activation function. The embedding size was chosen from a range of $8$ to $256$, the batch size was kept to $32$, and the variable learning rate ranged from $1e^{-3}$ to $0.1$.

### H. Testing

The framework with the associated variants was benchmarked against published recommendation systems from academic literature, where the results were averaged to enable comparison[4]:

- **Social Recommendation** (SoRec): Hao et al. developed a recommendation system to use a factor analysis approach based on probabilistic matrix factorisation to incorporate users' social network information [21].
- **Social Regularization** (SoReg): Consequently, Hao et al. also suggested constraining the matrix factorisation objective function by employing average-based regularisation (ratings from all the social connections), and an individual-based regularisation (constrains the framework to use only similar social connections) [22].
- **Deep neural network model on Social Relations for recommendation** (DeepSoR): Fan et al. suggested improving the accuracy of a recommender system by integrating a deep neural network to learn complex and non-intrinsic features from social relations [23].

[4]The implementations used for this research from the mentioned academic papers can be found in Appendix B.

- **Graph Convolutional Matrix Completion** (GC-MC): Berg et al. proposed a matrix completion framework: an auto-encoder that produced latent features of users and items at the node level on a bipartite interaction graph, which is then decoded by reconstructing rating links bi-linearly [24].
- **Variants**:
  1) $X_a$: Framework as described in this section.
  2) $X_b$: The item-space operation is disabled to evaluate the elimination of user-item interactions on the predicted ratings.
  3) $X_c$: Same as $X_b$ but instead of the item-space operation, the social-space operation is eliminated.

### *I. Evaluation*

The evaluation criteria included two negatively-oriented metrics (the lowest numerical value is considered the best) to analyse the classifiers' precision [25]:

- **Mean Absolute Error** (MAE): Measures the average magnitude of the prediction $y_i$ and true $x_i$ errors without any directional considerations:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - x_i| \tag{15}$$

- **Root Mean Squared Error** (RMSE): Same as MAE but with the caveat of square-rooting the average of the squared errors which translates into increased sensitivity for large errors:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(|y_i - x_i|)^2} \tag{16}$$

## IV. Results

TABLE III
TEST METRICS

| Dataset | Metric | Algorithm | | | |
|---|---|---|---|---|---|
| | | SoRec | SoReg | DeepSoR | GC-MC |
| Ciao | *MAE* | 0.87 | 0.85 | 0.83 | 0.81 |
| | *RMSE* | 1.04 | 1.06 | 1.03 | 1.02 |
| Epinions | *MAE* | 1.09 | 1.07 | 0.89 | 1.01 |
| | *RMSE* | 1.14 | 1.17 | 1.09 | 1.07 |

TABLE IV
VARIANT METRICS

| Dataset | Metric | Algorithm | | |
|---|---|---|---|---|
| | | $X_a$ | $X_b$ | $X_c$ |
| Ciao | *MAE* | 0.73 | 0.79 | 0.88 |
| | *RMSE* | 1.00 | 0.98 | 1.01 |
| Epinions | *MAE* | 1.01 | 1.04 | 1.02 |
| | *RMSE* | 0.87 | 1.02 | 0.99 |

TABLE V
WEIGHTED AVERAGE OF MAE AND RMSE

| Algorithm | Dataset | |
|---|---|---|
| | Ciao | Epinions |
| SoRec | 0.925 | 1.115 |
| SoReg | 0.950 | 1.120 |
| DeepSoR | 0.940 | 0.990 |
| GC-MC | 0.945 | 1.040 |
| $X_a$ | 0.985 | 1.000 |
| $X_b$ | 0.885 | 1.000 |
| $X_c$ | 0.945 | 0.945 |

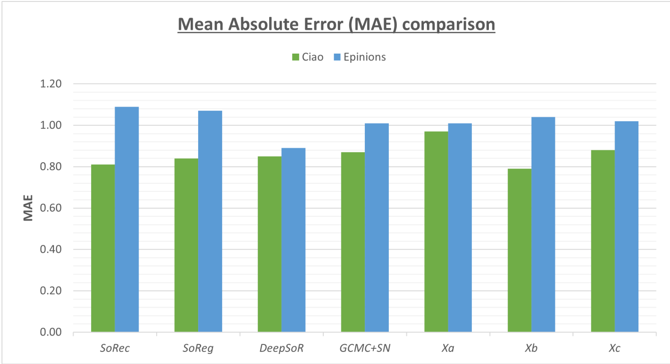


Fig. 6. MAE for both datasets

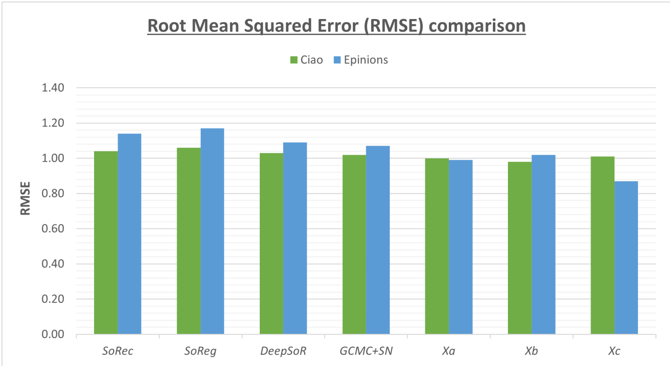


Fig. 7. RMSE for both datasets

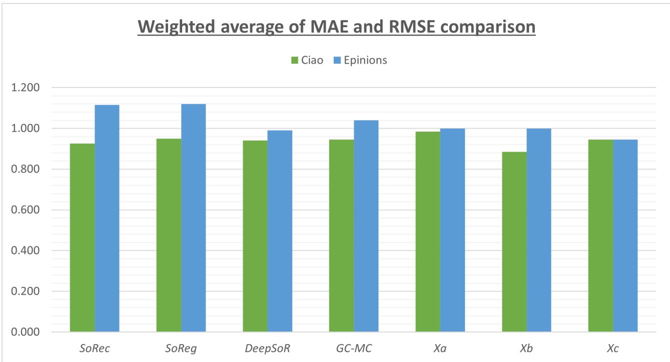


Fig. 8. Averaged metrics for all the tested models

## V. Discussion

The aim of this research was to develop a recommendation system using GNNs that utilises user-to-user relationship and user-to-item interactions information to output a predicted rating of how likely a user is to choose an item based on external factors such as the users social connections' influence. The methodology to develop the system required several neural networks to be connected in a modular framework with each module performing self-contained computations on graph-like data. These computations were then aggregated and integrated with the results of the other modules to update the final results, thereby highlighting the effect of using, (a) neural networks for graph processing, (b) learning features associated with the recommendation task, and (c) integrate latent features by mapping user-to-user relationships. The training and testing of the models to derive the evaluation metrics were done on a Google `Colaboratory` instance with standard GPU allocations. It should be noted that the sourced models were not implemented from first principles as that would require additional computational and academic resources that are beyond the scope of this project.

### A. Analysis

Table III and IV display the MAE and RMSE for all the models that were involved in this study whilst Table V highlights the weighted average of both the evaluation metrics. Evidently, the three deep learning frameworks outperform the the traditional models where the metrics are consistently lower in value and the percentage difference between the two metrics is also smaller. This further strengthens the case for the relevancy of GNNs in recommendation tasks due to increased computation ability and the incorporation of more features resulting in accurate inferences during testing.

The MAE for all the models is displayed in Fig. 6 and shows the $X_a$ framework outperforming the other architectures, yet again emphasising the importance of GNNs in representation learning, as the $X_a$ model contained both user-to-user and user-to-item information. The small difference between the MAE on both the datasets also displays the model's capability to handle fluid topological information i.e., nodes forming or changing and/or edges changing direction or associated connections. Subsequently, Fig 7 also shows the superiority of model $X_a$ while the DeepSoR model comes in second for the MAE metric and the $X_b$ model for the RMSE metric. Notably, the $X_c$ variation performs slightly worse on both metrics for both datasets, thereby highlighting the importance of incorporating user-to-user latent factors in the overall results as opposed to using only item latent factors.

Overall, the final weighted average results from Table V and Fig. 8 confirm the $X_a$ and DeepSoR frameworks as the most accurate within this research with a percentage difference of $4.14\%$ and $5.66\%$ for the Ciao and Epinions datasets, respectively. This further validates the need to incorporate GNNs and embed social context of a user in the recommendation framework to develop an accurate system.

### B. Further work

The first enhancement to this project can be ensuring that all the frameworks are implemented from first principles to establish constancy with regards to architectural designs and operating system idiosyncrasies. Furthermore, the classes or methods should also be unit and integration tested to increase the validity of the results and increase the applicability of the proposed framework in real-world settings.

This research can also be built upon by considering different types of datasets that contain more than social network information including, but not limited to, the strength of the connection between users or user preference for a specific category of products. This might require changing the architecture of the models to be capable of working with dynamic graph structures with multiple user-item associated vertices.

Additionally, combining GNNs and recurrent neural networks using a diffusion-convolutional process has shown promise in increasing the accuracy of the score matrix whilst being computationally efficient [26]. This technique could be employed to the models in this research, as during training, the GPU did time out several times due to the immense computational load.

## VI. Conclusion

This research project is a clear indicator to the importance of using item interaction information augmented with social network information in developing a holistically robust recommender system. The original framework was split into three variants to test the effect of removing either information source, where the results have been presented and examined to elucidate representative baselines. Additionally, further improvements have also been suggested and the applicability of each has been examined. Overall, the project was successful where a recommendation system was developed and compared with established and peer-to-peer reviewed frameworks.

## VII. Acknowledgements

I would like to thank my supervisor, Dr Angadh Nanjangud, for the amazing support, mentorship, and facilitating my request to research this domain. I would also like to extend a big thank you to the School of Electronic Engineering and Computer Science for the brilliant postgraduate programme in Artificial Intelligence.

I would also like to thank myself for putting in the required effort. Seldom does such unwavering loyalty to a cause come.

## References

[1] Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C. and Sun, M., 2020. Graph Neural Networks: A review of methods and applications. *AI Open*, 1, pp.57-81.

[2] Cai, D., Hu, J., Qian, S., Fang, Q., Zhao, Q. and Xu, C., 2021. GRecX: An Efficient and Unified Benchmark for GNN-based Recommendation. *arXiv preprint* arXiv:2111.10342.

[3] Guo, Z. and Wang, H., 2020. A Deep Graph Neural Network-based mechanism for Social Recommendations. *IEEE Transactions on Industrial Informatics*, 17(4), pp.2776-2783.

[4] Feng X., Liu Z., Wu W., and Zuo W., 2022. Social Recommendation via Deep Neural Network-based multi-task learning. *Expert Systems with Applications*. 10.1016/j.eswa.2022.117755. 206. (117755). Online publication date: 1-Nov-2022.

[5] Fan, W., Ma, Y., Li, Q., He, Y., Zhao, E., Tang, J. and Yin, D., 2019, May. Graph Neural Networks for Social Recommendation. *In The World Wide Web conference* (pp. 417-426).

[6] Wu, S., Sun, F., Zhang, W., Xie, X. and Cui, B., 2020. Graph Neural Networks in Recommender Systems: a survey. *ACM Computing Surveys* (CSUR).

[7] Fout, A., Byrd, J., Shariat, B. and Ben-Hur, A., 2017. Protein interface prediction using Graph Convolutional Networks. *Advances in neural information processing systems*, 30.

[8] Marx, V., 2022. Method of the year: Protein structure prediction. *Nature methods*, 19(1), pp.5-10.

[9] Battaglia, P., Pascanu, R., Lai, M. and Jimenez Rezende, D., 2016. Interaction networks for learning about objects, relations and physics. *Advances in neural information processing systems*, 29.

[10] Bi, Z., Zhang, T., Zhou, P. and Li, Y., 2020. Knowledge transfer for out-of-knowledge-base entities: Improving Graph-Neural-Network-based embedding using convolutional layers. *IEEE Access*, 8, pp.159039-159049.

[11] Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C. and Philip, S.Y., 2020. A comprehensive survey on Graph Neural Networks. *IEEE transactions on neural networks and learning systems*, 32(1), pp.4-24.

[12] Hamilton, W., Ying, Z. and Leskovec, J., 2017. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30.

[13] Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P. and Bengio, Y., 2017. Graph Attention Networks. *arXiv preprint* arXiv:1710.10903.

[14] Li, Y., Tarlow, D., Brockschmidt, M. and Zemel, R., 2015. Gated Graph Sequence Neural Networks. *arXiv preprint* arXiv:1511.05493.

[15] Guo, G., Zhang, J. and Yorke-Smith, N., 2015, February. Trustsvd: Collaborative filtering with both the explicit and implicit influence of user trust and of item ratings. *In Proceedings of the AAAI conference on Artificial Intelligence* (Vol. 29, No. 1).

[16] Jamali, M. and Ester, M., 2010, September. A matrix factorization technique with trust propagation for recommendation in Social Networks. *In Proceedings of the fourth ACM conference on Recommender systems* (pp. 135-142).

[17] Song, C., Wang, B., Jiang, Q., Zhang, Y., He, R. and Hou, Y., 2021, July. Social recommendation with implicit social influence. *In Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 1788-1792).

[18] Marsden, P.V. and Friedkin, N.E., 1993. Network studies of social influence. *Sociological Methods & Research*, 22(1), pp.127-151.

[19] McPherson, M., Smith-Lovin, L. and Cook, J.M., 2001. Birds of a feather: Homophily in social networks. *Annual review of sociology*, pp.415-444.

[20] Tieleman, T., and Hinton, G.,. 2012. Lecture 6.5-RMSProp. Neural Networks for Machine Learning [*Coursera online course*]. University of Toronto, Technical Report.

[21] Ma, H., Yang, H., Lyu, M.R. and King, I., 2008, October. SoRec: Social recommendation using probabilistic matrix factorisation. *In Proceedings of the 17th ACM conference on Information and knowledge management* (pp. 931-940).

[22] Ma, H., Zhou, D., Liu, C., Lyu, M.R. and King, I., 2011, February. Recommender systems with social regularization. *In Proceedings of the fourth ACM international conference on Web search and data mining* (pp. 287-296).

[23] Fan, W., Li, Q. and Cheng, M., 2018, April. Deep modelling of social relations for recommendation. *In Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 32, No. 1).

[24] Berg, R.V.D., Kipf, T.N. and Welling, M., 2017. Graph convolutional matrix completion. *arXiv preprint* arXiv:1706.02263.

[25] Wang, S., Tang, J., Wang, Y. and Liu, H., 2018. Exploring hierarchical structures for recommender systems. *IEEE Transactions on Knowledge and Data Engineering*, 30(6), pp.1022-1035.
[26] Monti, F., Bronstein, M. and Bresson, X., 2017. Geometric matrix completion with recurrent multi-Graph Neural Networks. *Advances in neural information processing systems*, 30.

## Appendix A
## Authors implementation

It is preferred that the reader accesses the code on the `NBViewer` extension for rendering and sharing Jupyter Notebooks online: **Link**

The code and associated utilities have been made open-source on the author's GitHub profile: **Link**

## Appendix B
## Academic paper implementations

- **SoRec**: Link
- **SoReg**: Link
- **DeepSoR**: Link
- **GC-MC**: Link