iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 11 **Issue:** I **Month of publication:** January 2023

DOI: https://doi.org/10.22214/ijraset.2023.48539

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 11 Issue I Jan 2023- Available at www.ijraset.com

Recommendation Systems using Graph Neural Networks

Dhairya Khanna[1], Rishabh Bhushan[2], Khushboo Goel[3], Dr. Sudha Narang[4]
Department of Computer Science, Maharaja Agrasen Institute of Technology, New Delhi, India

Abstract: On the internet as the technology improves, the number of choices is overwhelming due to which there is need to filter, prioritize and efficiently deliver relevant information in order to alleviate the problem of information overload, which has created a potential problem to many Internet users. Recommender systems solve this problem by searching through large volume of dynamically generated information to provide users with personalized content and services. Recently, graph neural network (GNN) techniques have been widely utilized in recommender systems since most of the information in recommender systems essentially has graph structure and GNN has superiority in graph representation learning. This paper aims to provide a comprehensive application of GNN-based recommender systems.

I. INTRODUCTION

A. Recommendation System

Recommendation is a critical, complex and challenging issue in artificial intelligence research. The existing recommendation methods could be divided into content-based methods, collaborative filtering methods, and hybrid methods.

1) The proposal of the content-based recommendation method originated from information retrieval research and is a kind of widely used traditional recommendation system. Content-based recommendation method mainly involves two types of data: content features of items and user profiles (interests). Its basic idea is to calculate the similarity between the items in a subset and the items in the whole set. The former is selected from the whole set according to explicit feedback (ratings, likes, etc.) and implicit feedback (search, click, buy, etc.). Then, this method sorts them into a recommendation list according to their similarity. The disadvantage is that it relies heavily on domain experts and complex feature engineering when constructing content features and user profiles. In addition, the content-based recommendation method suffers from the 'cold start' problem for new users. Recommender systems are a sort of information filtering technology that aims to offer information items that are likely to be of interest to the user. The cold start problem occurs when the system is unable to form any relation between users and items for which it has insufficient data. • Collaborative filtering recommendation is another widely used traditional recommendation system, which can provide interested content for a specific user. It firstly finds the users who are similar to this user, and then recommend the content to this user according to those users' interests. The collaborative filtering recommendation method can be further classified into memory-based collaborative filtering recommendation and model-based collaborative filtering recommendation. The former can be subdivided into item-based collaborative filtering and user-based collaborative filtering. The latter mainly includes SVD [4] and its variant, which can map data to a low dimension and calculate the similarities between items. However, the collaborative filtering recommendation method still suffers from a data sparse problem. In other words, when the interaction matrix of items and users is a sparsity matrix, it is difficult to find users' neighborhood information in the sparsity matrix, which leads to a lower recommendation accuracy. In general, the larger the data size, the smaller the item overlap and the sparser the matrix. In addition, it also has a 'cold start' problem. It cannot recommend a new product without historical data by collecting a user's behavior, such as browsing, clicking or buying.

2) To further improve the recommendation performance, especially in the case of sparse data and 'cold start', researchers proposed the hybrid recommendation method, whose core idea is to combine recommendation methods with auxiliary information. The sources of this auxiliary information are diverse, of which the more prominent is the knowledge graph . Knowledge graph is a directed heterogeneous graph where the node represents entity and the edge represents relation , so that it can provide rich semantic relations between entities. Compared with KG-free methods, incorporating KG into recommendation benefits the results. The rich semantic relatedness among items in a KG can help explore their latent connections and improve the precision of results. Therefore, the KG-based hybrid recommendation methods have better performance than the KG-free methods

©IJRASET: All Rights are Reserved

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 11 Issue I Jan 2023- Available at www.ijraset.com

II. METHOD

The dataset used for the project is Movielens - 25M Dataset and the repository can be seen at https://github.com/Dhairya3 recommendation-system.

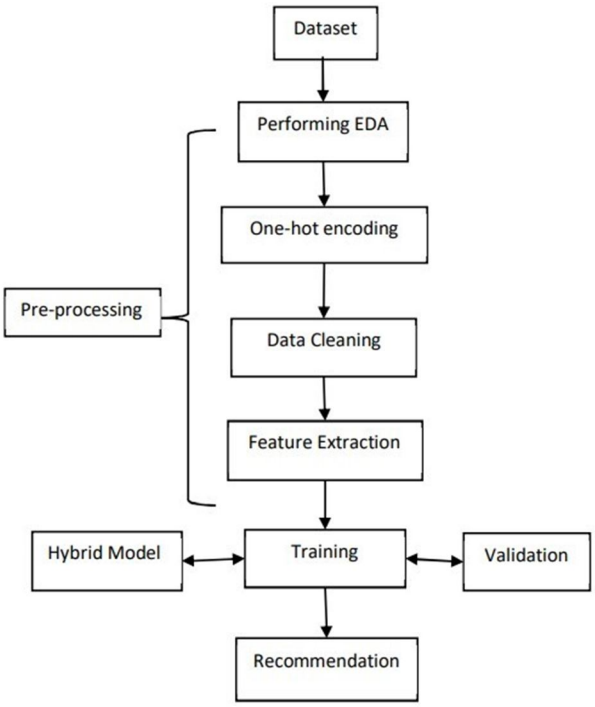


Figure 1: Flowchart for creation of Recommendation System

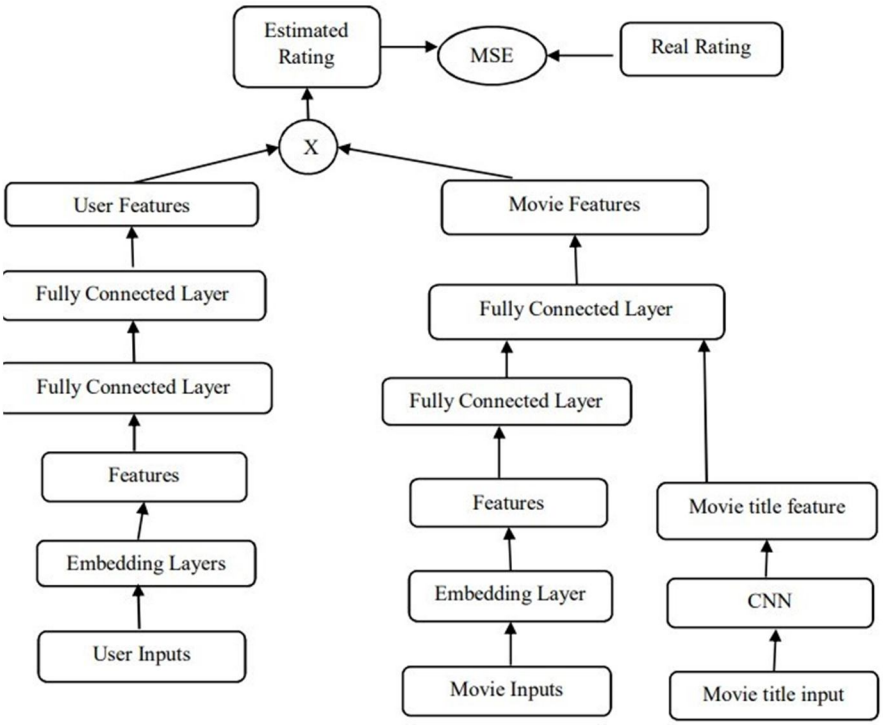


Figure 2: Flowchart for proposed convolution network of Recommendation System

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 11 Issue I Jan 2023- Available at www.ijraset.com

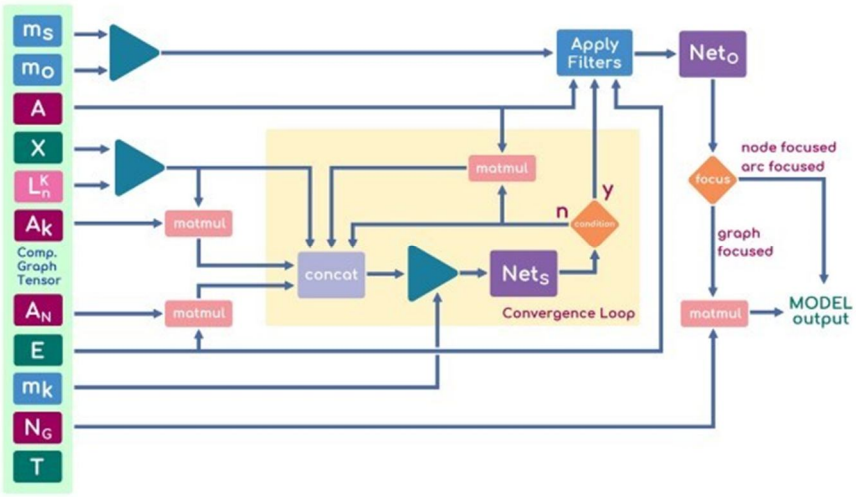


Figure 3: Composite GNN model.

The GraphSequencer generates from GraphObjects batches of GraphTensor which are presented to the model as input. All quantities pass through multiple operations (matrix multiplications, boolean mask filtering, and concatenating processes) to form the input to nets and neto

GNN Model Since GNNkeras is a Keras-based software, where the GNN classes inherit from the Keras.Model class, it comes with all the functionalities provided by TensorFlow 2.x and Keras (TensorFlow backend). To parallelize software execution on modern CPUs and GPUs, all the operations have been based on matrix multiplications. Fig. 3 shows the processing scheme of a heterogeneous graph by a Composite GNN model.

III. IMPACT

Graph data are nowadays ubiquitous, allowing to represent relational information between data entities in many different research domains. Furthermore, the importance of graphs is increasing as they permit to directly merge information from different sources, which is very natural in modern applications.

On the one hand, graphs are a powerful form of data representation, in particular for relational information, and applications of machine learning techniques in this domain become more important every day. On the other hand, standard machine learning methods were based on flat vectorial data, thus their application on graph domains was difficult. This implied that graph data had to be preprocessed, losing relevant pieces of information, in particular on the relational side. This limit stimulated the proposal and the success of GNNs, which are a class of machine learning models that can process graph data directly. In the last few years, the interest of researchers in the field has known a recent and steady increase, leading to the development of a large number of new models [15], several theoretical studies , so as a huge number of real-world applications. GNNkeras has been designed in this context with the aim of simplifying the use of the GNNs.

Graph-based networks can be classified into two broad classes, recurrent and convolutional . GNNkeras is focused on the former class of GNNs, on which our group has accumulated long expertise. As far as we know, this software is the first solution for TensorFlow 2.x specifically designed for recurrent GNNs. The main difference between these two kinds of models is the way they operate directly on graphs. In the Convolutional case, the graph is processed by means of convolutional and pooling techniques, which are carried out by a fixed number of stacked layers, using independent parameters in the various layers. Instead, in the recurrent case, the MLP units share the architecture and parameters, and the state computation can be carried out for a fixed number of times or until convergence to a steady state.

IV. ACKNOWLEDGEMENTS

Please add any relevant acknowledgements to anyone else that assisted with the project in which the data was created but did not work directly on the data itself.

REFERENCES

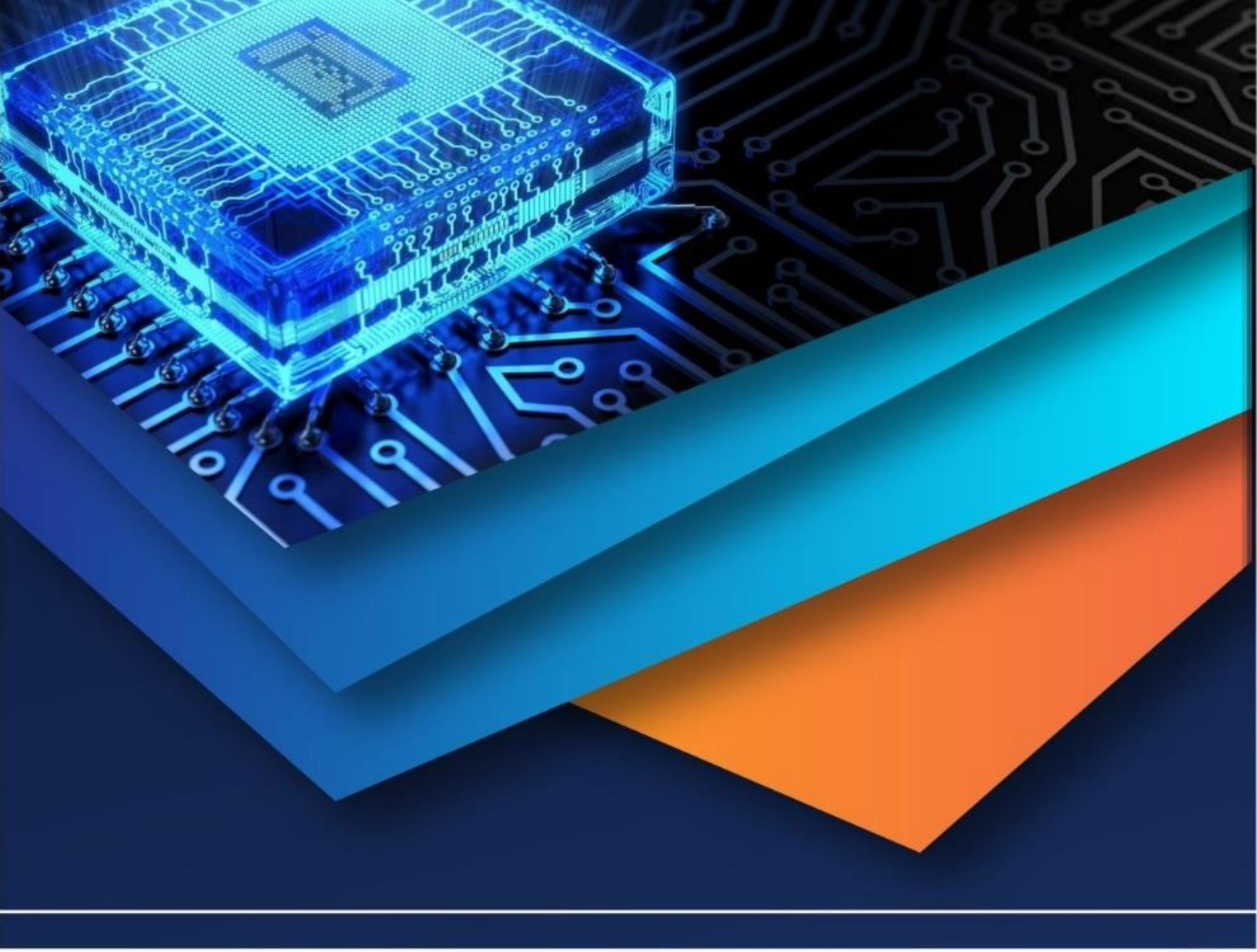
[1] Niccol`o Pancino , Pietro Bongini, Franco Scarselli, Monica Bianchini (2022, March 17) GNNkeras: A Keras-based library for Graph Neural Networks and homogeneous and heterogeneous graph processing, Retrieved from https://www.softxjournal.com/action/showPdf?pii=S2352-7110

[2] SHIWEN WU, FEI SUN, WENTAO ZHANG, XU XIE, BIN CUI (2022 April 02) Graph Neural Networks in Recommender Systems: A Survey, Retrieved from https://arxiv.org/pdf/2011.02260.pdf

[3] Jian Weia, Jianhua He, Kai Chen, Yi Zhou, Zuoyin Tanga (2017 March 01) Collaborative filtering and deep learning based recommendation system for cold start items, Retrieved from https://www.sciencedirect.com/science/article/abs/pii/S0957417416305309

[4] Oweys Momenzada, Michael Palk and Stefan Voß () GRAPH NEURAL NETWORKS FOR EFFICIENT RECOMMENDER SYSTEMS

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)