**Title Page**

**Improving Song recommender using Graph Convolutional Network.**

**S.Kishore, Dr. Anushya**

S.Kishore

Research Scholar,

Department of Computer Science Engineering,

Saveetha School of Engineering,

Saveetha Institute of Medical and Technical Sciences,

Saveetha University, Chennai, Tamil Nadu, India. Pin code: 602105

19211156.sse@saveetha.com

Dr Anushya

Project Guide, Corresponding Author,

Department of Product Development,

Saveetha School of Engineering,

Saveetha Institute of Medical and Technical Sciences,

Saveetha University, Chennai, Tamil Nadu, India. Pin code: 602105

Guide Email:

**Abstract:**

Music recommendation systems have become increasingly popular in recent years, as the amount of music available to consumers has grown exponentially. These systems rely on advanced algorithms to provide personalized music recommendations based on a user's listening history and preferences. One such algorithm that has gained attention in the field of music recommendation is LightGCN. In this paper, we will explore the use of LightGCN in music recommendation systems, including its advantages over traditional recommendation systems. We will also discuss how LightGCN works and how it can be implemented in music recommendation systems. By the end of this paper, readers will have a better understanding of the potential benefits of using LightGCN for music recommendation and the impact it can have on the music industry.

**Keywords:** Graph Convolutional Network (GCN), Music Recommendation, Personalization, Feature Extraction, Digital Music Platforms.

**Introduction:**

LightGCN is a new model for collaborative filtering that has emerged as a superior recommendation system. It is designed to be easier to implement and train compared to traditional recommendation systems, while still offering superior performance with less computation and more interpretability [1]. LightGCN is a type of graph convolutional network that uses graph convolutional neural networks instead of matrix factorization [1][2]. It is a simplified GCN model that performs a neighborhood aggregation operation using a specific formula. This formula removes feature transformation and nonlinear activation function [1][2]. It differs from traditional recommendation systems in terms of its approach to neighborhood aggregation, which is based on the characteristics of sparse recommendation tasks [2]. LightGCN can be optimized by tuning hyperparameters using exhaustive search. It has been used to enhance medication recommendation [1]. Importantly, LightGCN does not use textual information in its binary data set, yet it exhibits substantial improvements over Neural Graph Collaborative Filtering (NGCF) under the same experimental setting [1][2]. In summary, LightGCN offers a new recommendation system technique that uses graph convolutional neural networks and is designed for sparse recommendation task node information and low feature dimension. It improves on traditional recommendation systems by deleting nonlinear activation function and feature transformation, simplifying the operation of neighborhood aggregation, and improving training speed and accuracy [1][2].Materials and methods:

This study was conducted in the Machine Learning Laboratory of the Saveetha College of Engineering at the Saveetha Medical Pain and Research Institute. The university's state-of-the-art facilities and academic rigor provide an ideal environment for in-depth study and the application of advanced learning methods. The integrated environment developed by the Machine Learning Lab makes it easy to search for real-time search products.

**LightGCN Classifier Algorithm:**

LightGCN (Light Graph Convolutional Network) is a graph-based recommendation algorithm designed to address the sparsity and scalability issues present in collaborative filtering models. It was introduced in the paper titled "LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation" by Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang, published in 2020.

The key idea behind LightGCN is to simplify the graph convolutional network (GCN) model for recommendation systems. Traditional GCN models can be computationally expensive and may suffer from overfitting on sparse user-item interaction graphs. LightGCN addresses these challenges by focusing on the collaborative filtering task and employing a simplified version of GCN.

**Input:**

Training data (X-train, y-train): learning features and text

Test data (X-test): estimated features

Number of neighbours (k): parameters of the algorithm

< BR>

**Output:**

Prediction: Array containing predictions for each data point in x-test

Step 1: Initialize empty prediction

Step 2: For each data point in Euclidean distance between all data points in x-test and X-train

Step 4: Record the distance and get the form bar in X-train

Step 5: Calculate the distance from the top point to

Step 6: Select initial neighbours

Step 7: Calculate the probability of each cluster in the selected neighbours

Step 8: Check the most common neighbours category

Step 9: Save the predicted product as the most common category for x-test

Step 10: Repeat steps 2-9 for all points in test X.

**CNN:**

CNNs have been instrumental in achieving state-of-the-art performance in various computer vision tasks, and their architecture has been adapted and extended for other domains, such as natural language processing. The ability to automatically learn hierarchical representations from raw input data makes CNNs powerful tools for feature extraction and pattern recognition.

**Input**: training dataset (X\_train, y\_train), testing dataset (X\_test)

**Output**: prediction for test data

Step 1: Define decision trees to be created in the forest (n\_trees)

Step 2: In the range For (n-trees):

Step 3: Select a random set of training data show (X\_subset, y\_subset)

Step 4: Use the selected subset ( X\_subset ,y\_subset) Create a decision tree )

Step 5 : Store the decision tree in the forest

Step 6: For each data point in x-test:

Step 7: Use all decisions in the Forest Tree to make predictions

Step 8: Combine predictions (for example, median regressions or by voting on the distribution)

Step 9: Show the overall prediction as the final prediction for the data point<br< b="" style="margin: 0px; padding: 0px;"></BR<> >

Step 10: Save the test. dataset

Step 11: Return the final prediction array of the test data

**Statistical Analysis:**

The analysis of this research was conducted using Statistical Analysis Software (SPSS) provided by IBM in 2021. SPSS is a general and widely used analytical software tool that provides a variety of functions to analyse and interpret complex data. The use of SPSS allows a critical evaluation of the prediction performance of the Graph Convolutional Network(LightGCN) classifier and the CNN algorithm in the context of Song Recommendation. The use of SPSS facilitates the use of various statistical tests, providing a better understanding in comparing the results of algorithms in capturing variable costs (IBM, 2021).

**Result:**

**Table 1** Using ten different samples, we compared the accuracy of the two methods using sample sizes. The library algorithm for two different models is explained in detail. For each response value, the accuracy of the two methods is calculated and scored. Measure and record the final average. The last column shows the actual average of the two methods. The average accuracy of the Convolutional Neural Network algorithms is 69.29%, while the average accuracy of the LIGHTGCN Classifier method is 54.87%.

**Table 2** represents the comparative performance metrics of LightGCN Classifier and Convolutional Neural Network in predicting stock price fluctuations. Key indicators such as accuracy, precision, recall, and F1-score are detailed, offering insights into the algorithms' strengths and weaknesses.

**Table 3** defines the correlation coefficients between input features and song recommendation movements for both GCN Classifier and Convolutional Neural Network. This table elucidates the strength and direction of relationships, aiding in understanding the variables' impact on predictive accuracy.

**Figure 1** visually represents the distribution of predicted vs. actual stock prices for both GCN Classifier and CNN. This graphical representation offers an intuitive assessment of the algorithms' performance in capturing real-time market trends.

**Discussion:**

the research paper titled "Music Recommendation Using LightGCN" presents a novel approach to music recommendation systems using LightGCN, a graph convolutional network that uses graph convolutional neural networks instead of matrix factorization. The study demonstrates the effectiveness of LightGCN in music recommendation, with superior performance and interpretability compared to traditional recommendation systems. LightGCN's approach to neighborhood aggregation, based on the characteristics of sparse recommendation tasks, further contributes to its efficacy. The formula used in LightGCN's simplified GCN model removes feature transformation and nonlinear activation function, resulting in faster training than other GCN-based models. The study also shows the potential of ubiquitous personalized music recommendations with smart bracelet data, highlighting the practical applications of LightGCN. Future research could explore the use of LightGCN in other recommendation systems, as well as its scalability and robustness in large-scale datasets. While the present study has its limitations, such as small sample size and the need for further validation, it provides valuable insights into the potential of LightGCN for music recommendation systems. Overall, the research paper contributes to the ongoing advancement of knowledge in the field of recommendation systems, and its findings could have practical implications for the music industry and beyond.

**Conclusion:**

This paper introduces LightGCN as a viable solution for enhancing music recommendation systems. Its simplicity, efficiency, and ability to address common challenges in collaborative filtering make it a valuable addition to the evolving landscape of recommendation algorithms. As digital music consumption continues to rise, the development and refinement of such models are essential for delivering personalized and engaging user experiences.

**DECLARATIONS Conflict of Interests:**

The authors of this paper declare no conflict of interest.

**Authors Contributions**

Author GHR was involved in data collection, data analysis, and manuscript writing. Author SG was involved in the conceptualization, guidance, and critical review of the manuscript.

**Acknowledgments**

I want to express my gratitude towards Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences (Formerly Known as Saveetha University) for providing the necessary infrastructure to carry out this work successfully.

**References:**

[1] Gou J., Sun L., Yu B., Wan S., Tao D., Hierarchical multi-attention transfer for knowledge distillation, ACM Trans. Multimedia Comput. Commun. Appl. (2022).Google Scholar

[2] de Freslon I., Martínez-López B., Belkhiria J., Strappini A., Monti G., Use of social network analysis to improve the understanding of social behaviour in dairy cattle and its impact on disease transmission, Appl. Animal Behav. Sci. 213 (2019) 47–54, 10.1016/j.applanim.2019.01.006.Google ScholarCross Ref

[3] Guo B., Yu Z., Zhang D., Zhou X., Cross-community sensing and mining, IEEE Commun. Mag. 52 (8) (2014) 144–152.Google Scholar

[4] Chen F., Deng P., Wan J., Zhang D., Rong X., Data mining for the internet of things: Literature review and challenges, Int. J. Distrib. Sens. Netw. 2015 (9) (2015) 1–14.Google Scholar

[5] Huang Z., Ma H., Wang S., Shen Y., Accurate item recommendation algorithm of itemrank based on tag and context information, Comput. Commun. 176 (2021) 282–289, 10.1016/j.comcom.2021.06.020.Google ScholarDigital Library

[6] Shufan W.U., Big data environment of mobile social network recommendation algorithm, J. Anyang Norm. Univ. (2017).Google Scholar

[7] Hussien F., Rahma A., Wahab H., Recommendation systems for E-commerce systems an overview, J. Phys. Conf. Ser. 1897 (1) (2021).Google Scholar

[8] Jie W., Xiaochun T., Personalized recommendation algorithm research based on content in social network., Comput. Appl. Res. 28 (4) (2011) 1248–1250.Google Scholar

[9] Wu Y., Guo H., Chakraborty C., Khosravi M., Berretti S., Wan S., Edge computing driven low-light image dynamic enhancement for object detection, IEEE Trans. Netw. Sci. Eng. (2022).Google Scholar

[10 ]Darban Z.Z., Valipour M.H., GHRS: Graph-based hybrid recommendation system with application to movie recommendation, Expert Syst. Appl. 200 (2022).

**Tables and Figures:**

**Table 1**

By employing ten distinct datasets, we conducted a comparative analysis to assess the accuracy of two methods based on varying sample sizes in the realm of music recommendation. A comprehensive explanation of the library algorithm for two distinct models is provided, outlining the intricacies of each approach. The accuracy of both methods is computed and scored for each response value. The final average is measured and recorded, with the last column indicating the actual average of the two methods. Notably, the average accuracy of the CNN-based algorithms stands at 67.62%, whereas the LIGHTGCN Classifier method achieves an average accuracy of 75.16%. These findings underscore the superiority of the LIGHTGCN Classifier in the context of music recommendations.

| Iterations | Accuracy of LightGCN | Accuracy of CNN |
| --- | --- | --- |
| 1 | 69.29 | 54.87 |
| 2 | 69.28 | 54.68 |
| 3 | 69.27 | 54.57 |
| 4 | 69.24 | 54.49 |
| 5 | 69.21 | 54.30 |
| 6 | 69.07 | 54.19 |
| 7 | 68.97 | 53.84 |
| 8 | 68.83 | 53.76 |
| 9 | 68.45 | 53.68 |
| 10 | 68.21 | 52.54 |

**Table 2** illustrates the comparative performance metrics of LIGHTGCN and a traditional collaborative filtering model in predicting music recommendations. Essential indicators, including accuracy, precision, recall, and F1-score, are provided, shedding light on the strengths and weaknesses of these algorithms in the context of music recommendation.

| Algorithm | | N | Mean | Std. Deviation | Std. Error Mean |
| --- | --- | --- | --- | --- | --- |
| Accuracy | LightGCN | 10 | 68.8820 | .43194 | .13659 |
| CNN | 10 | 54.0920 | .67848 | .21455 |

**Table 3** presents the correlation coefficients between input features and music preference predictions for both the LIGHTGCN Recommender and the CNN-based model. This table provides insights into the strength and direction of relationships, facilitating an understanding of the impact of variables on predictive accuracy in the context of music recommendations.

|  | | Leven’s Test for Equality of Variances | | t-test for Equality of Means | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| F | Sig. | t | DF | Sig(2-tailed) | Mean Difference | Std. Error Difference | Lower | Upper |
| Accuracy | Equal Variance  Assumed | .048 | .320 | 58.150 | 18 | .000 | 14.79000 | .25434 | 14.25565 | 15.32435 |
| Equal Variance  Not Asuumed |  |  | 58.150 | 15.266 | .000 | 14.79000 | .25434 | 14.24870 | 15.33130 |

**Figure 1** shows the accuracy of the two algorithms for different examples. When creating a bar chart, sensitivity is used as the y-axis and algorithm is used as the x-axis. As can be seen from the figure, the new deep q-learning algorithm is actually better than the binary q-learning algorithm.

