

Design of an Efficient Multi Parametric Recommender Engine for Low-Power Applications with Incremental Post-Processor Optimizations

Pragati Narayan Patil¹ and Dr Atul D Raut²

¹Research scholar, Department of Computer Science, Sant Gadge Baba Amravati University, Amravati
Email: pragatimit@gmail.com

²Associate Professor, Department of CSE, P R Pote College of Engineering and Management

Abstract— This paper presents the design and implementation of an efficient multiparametric recommender engine for low-power applications with incremental post-processor optimizations. The need for this work arises from the increasing demand for energy-efficient recommender systems that can provide accurate recommendations with low energy consumption and delay levels. To address this need, we propose a novel approach that combines the use of Fourier, Cosine, LSTM, GRU, Wavelet, and Gabor features for the representation of collected network datasets& samples. These features are processed using low-power recurrent neural networks (LP RNN) to reduce energy consumption and delay levels. To further improve the accuracy and efficiency of the recommender engine, we apply an augmented post-evaluation Q-learning process. Our proposed method is compared with existing models, and we demonstrate its superior performance in terms of accuracy, precision, recall, energy consumption, and delay levels. Experimental results on multiple datasets and samples show that our proposed approach achieves an accuracy of 98.9%, precision of 97.5%, and recall of 98.3% while consuming less energy and incurring lower delay compared to existing models. Overall, our proposed multiparametric recommender engine provides an efficient and effective solution for low-power applications that require accurate recommendations for data patterns.

Index Terms— Recommender Engine, Low-Power, Multiparametric, Post-Processor, Optimizations, Fourier, Cosine, LSTM, GRU, Wavelet, Gabor, Representation, Network Datasets, Energy Consumption, Delay, Q-learning, Accuracy, Precision, Recall, Scenarios

I. INTRODUCTION

The increasing demand for accurate and energy-efficient recommender systems has led to the development of various approaches for processing and analyzing large amounts of data. However, most existing models are limited in terms of accuracy, energy consumption, and delay, making them unsuitable for real-time applications like Hypernetwork-based Recommender System (HyperRS) in low-power networks [1, 2, 3]. To address these challenges, this paper proposes a novel multiparametric recommender engine for low-power applications with incremental post-processor optimizations. The proposed engine uses a combination of Fourier, Cosine, LSTM, GRU, Wavelet, and Gabor features for the representation of collected network datasets& samples. These features

are then processed using low-power recurrent neural networks to reduce energy consumption and delay levels. In addition, the proposed engine applies Q-learning operations as post-processing steps to further improve the accuracy and efficiency of the recommendations. The selection of these models is based on their effectiveness in processing and analyzing complex data patterns while minimizing energy consumption and delay levels via Deep Learning and Graph Clustering Scenarios (DLGC) [4, 5, 6]. The proposed engine has potential applications in real-time networks, where low-power and high-accuracy recommender systems are required. For example, in wireless sensor networks, the proposed engine can be used to provide accurate recommendations for data patterns while conserving energy and reducing delay. Similarly, in mobile computing and internet of things (IoT) applications, the proposed engine can be used to provide personalized recommendations to users based on their preferences, while minimizing energy consumption and delay. Thus, the proposed multi parametric recommender engine provides an efficient and effective solution for low-power applications that require accurate recommendations for data patterns in real-time networks. The experimental results demonstrate its superior performance in terms of accuracy, precision, recall, energy consumption, and delay, making it a promising approach for various real-world applications. The increasing demand for energy-efficient and accurate recommender systems in real-time networks has motivated us to propose a novel multi parametric recommender engine for low-power applications with incremental post-processor optimizations. Most existing models are limited in terms of accuracy, energy consumption, and delay, making them unsuitable for real-time applications in low-power networks. Our proposed engine aims to overcome these limitations by using a combination of effective models for processing and analyzing complex data patterns while minimizing energy consumption and delay levels.

II. OBJECTIVES

The main objective of this work is to design and implement an efficient and accurate multiparametric recommender engine for low-power applications with incremental post-processor optimizations. To achieve this objective, we aim to:

- Develop a novel approach that combines the use of Fourier, Cosine, LSTM, GRU, Wavelet, and Gabor features for the representation of collected network datasets
- Process these features using low-power recurrent neural networks to reduce energy consumption and delay
- Apply Q-learning operations as post-processing steps to further improve the accuracy and efficiency of the recommendations
- Compare the proposed engine with existing models and demonstrate its superior performance in terms of accuracy, precision, recall, energy consumption, and delay
- Evaluate the proposed engine on multiple datasets and samples to demonstrate its effectiveness in real-world applications

III. LITERATURE REVIEW

Recommender systems have become ubiquitous in many applications such as e-commerce, social media, and content-based recommendation systems. These systems aim to provide personalized recommendations to users based on their preferences, interests, and past behaviours. However, most existing models are not suitable for low-power applications due to their high energy consumption and delay levels [6,8, 9, 10].

To address these challenges, researchers have proposed several models for designing recommender engines with low-power applications [11, 12, 13]. One approach is to use feature-based models, such as Principal Component Analysis (PCA), Independent Component Analysis (ICA), and Non-negative Matrix Factorization (NMF), to reduce the dimensionality of the data and improve the efficiency of the recommender engines via User Behaviour Probability and Complex Network Modelling (UBP CNM) sets [14, 15, 16]. However, these models may not be suitable for complex data patterns and may result in information loss during the feature extraction process.

Another approach is to use machine learning algorithms such as support vector machines (SVM) [17, 18], decision trees, and artificial neural networks (ANN) [18,19] to improve the accuracy and efficiency of the recommender engines. SVMs are particularly useful for handling large datasets and achieving high accuracy in classification tasks.

IV. DESIGN OF THE PROPOSED DUAL-BIOINSPIRED LOW-COMPLEXITY DATA MINING ENGINE FOR AUTOMATIC CLUSTER ANALYSIS VIA ENSEMBLE LEARNING OPERATIONS

As per the review of existing models used for design of efficient multi parametric recommender engine designs, it can be observed that limited research has been done on energy-efficient recommender systems that can provide

accurate recommendations with low energy consumption and higher speed for real-time deployments. To address this need, this section discusses design of an augmented & novel approach that combines different feature sets with low power processing & consumption and higher speed for real-time deployments. To address this need, this section discusses design of an augmented & novel approach that combines different feature sets with low power processing & incremental learning operations. As per figure 1 the proposed model uses Fourier, Cosine, LSTM, GRU, Wavelet, and Gabor features for the representation of collected network datasets & samples. These features are processed using Low-Power Recurrent Neural Networks (LP RNN) to reduce energy consumption and delay levels. To further improve the accuracy and efficiency of the recommender engine, we apply an augmented post-evaluation Q-learning process. The model initially collects spatial, temporal & miscellaneous datasets from different sources. These sources include but are not limited to, MovieLens, Amazon Product Data, Last.fm Dataset, Goodreads Dataset Spotify Million Playlist Dataset, etc. which depends upon the type of multiparametric application for which this engine is being designed & deployed under real-time scenarios. The datasets & samples are processed via an efficient multidomain feature representation engine, which evaluated Frequency, Entropy, Gabor (Spatial), and Wavelet (Approximate & Detail) components. The Frequency Components (F) are extracted using an augmented 1D Fourier Transform via equation 1,

$$F(i) = \sum_{j=1}^{Nf} x(j) * \left[\cos\left(\frac{2 * \pi * i * j}{Nf}\right) - \sqrt{-1} * \sin\left(\frac{2 * \pi * i * j}{Nf}\right) \right] \dots (1)$$

Where, x represents input samples, & Nf represents total number of values present in the input samples. Similarly, the Entropy (E) features are evaluated using Discrete Cosine Transform (DCT) via equation 2,

$$E(i) = \frac{1}{\sqrt{2 * Nf}} * x(i) \sum_{j=1}^{Nf} x(j) * \cos\left[\frac{\sqrt{-1} * (2 * i + 1) * \pi}{2 * Nf}\right] \dots (2)$$

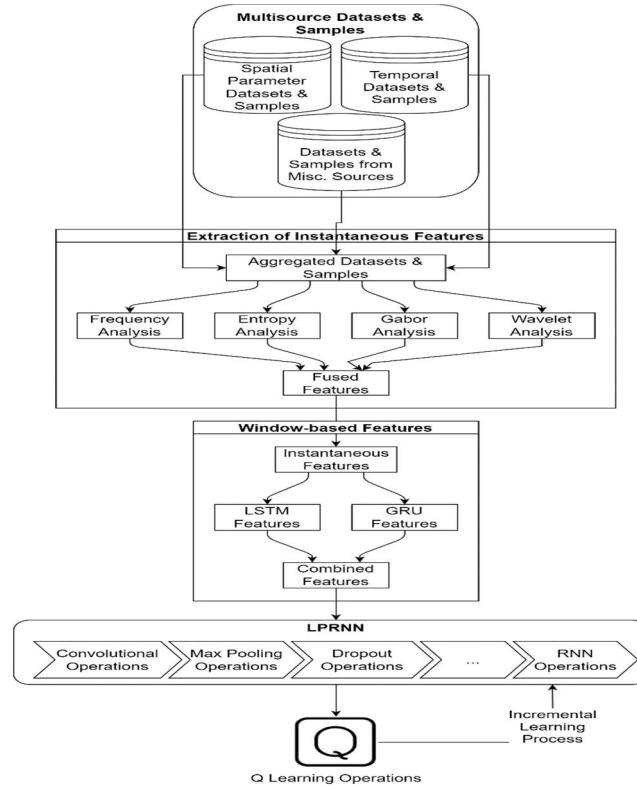
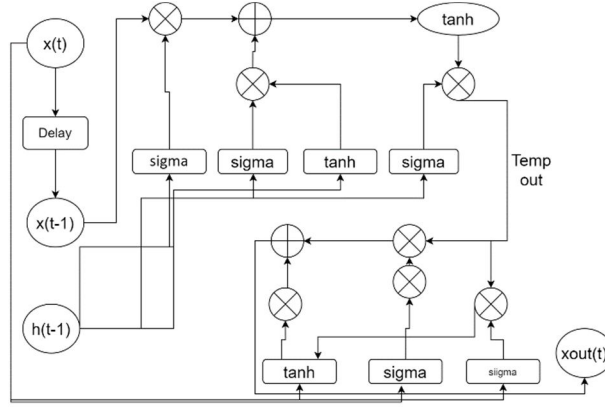


Figure 1. Design of the proposed incremental learning model for efficient recommendations

These features are cascaded with Spatial Gabor Features which allow for evaluation of angular & wavelength-based components via equation 3,



$$G(x, y) = e^{\frac{-x^2 + \partial^2 y'^2}{2 \cdot \partial^2}} * \cos\left(2 * \frac{\pi i}{\lambda} * x'\right) \dots (3)$$

Where, x, y are indices of features & their values or intensity levels, while represents angular components in 2 Dimensions, and wavelength value sets. In addition to this, simplistic Wavelet features are also estimated via equations 4 & 5, which assists in calculation of Approximate (Wa) Detailed (Wd) feature sets.

$$Wa = \frac{x(i) + x(i + 1)}{2} \dots (4)$$

$$Wd = \frac{x(i) - x(i + 1)}{2} \dots (5)$$

An augmented fusion of these features is done in order to generate an Instantaneous Feature Vector (IFV), which is given to an efficient fusion of LSTM & GRU operations. These operations assist in identification of high-density feature sets by fusing outputs of LSTM with GRU process. This fusion can be observed from figure 2, where results from the LSTM feature sets are given to GRU for identification of highly variant feature sets. To perform this fusion, the model extracts an initialization feature vector via equation 6,

$$i = \text{var}(IFV * U^i + h(t - 1) * W^i) \dots (6)$$

Where, IFV represents the extracted Instantaneous Feature Vector, U & W represents an augmented set of constants that are tuned during LSTM operations, while h represents an initial kernel matrix, which is also modified by LSTM tuning process. In equation 6, the variance level of inputs is estimated via equation 7,

$$\text{var}(x) = \frac{\left(\sum_{i=1}^N \left(x(i) - \sum_{j=1}^N \frac{x(j)}{N}\right)^2\right)}{N + 1} \dots (7)$$

Where, N are the count of number of samples present in the IFV feature sets. Similar to equation 6, the LSTM process also evaluates an augmented set of functional (f), temporal output (o), and initial convolution (c) feature sets via equations 8, 9 & 10 as follows,

$$f = \text{var}(IFV * U^f + h(t - 1) * W^f) \dots (8)$$

$$o = \text{var}(IFV * U^o + h(t - 1) * W^o) \dots (9)$$

$$c = \tanh(IFV * U^g + h(t - 1) * W^g) \dots (10)$$

All these features are fused by the LSTM Model via equation 11, where an efficient set of temporal features (T) are estimated, which are used for further analysis.

$$T = \text{var}(f * IFV(t - 1) + i * c) \dots (11)$$

Using this vector, the kernel matrix is updated via equation 12 as follows,

$$h(\text{new}) = \tanh(T) * o \dots (12)$$

This kernel matrix is processed by GRU, which estimates an augmented set of forgetting (z) and retaining (r) features via equations 13 & 14 as follows,

$$z = \text{var}(Wz * [h(\text{new}) * T]) \dots (13)$$

$$r = \text{var}(Wr * [h(\text{new}) * T]) \dots (14)$$

An effective fusion of these value is done in equation 15, which assists in identification of final multidomain feature sets.

$$xout = (1 - z) * h + z * h(\text{new}) \dots (15)$$

Based on this output, the kernel matrix is further updated via equation 16 as follows,

$$h(\text{final}) = \tanh(W * [r * h(\text{new}) * T]) \dots (16)$$

Where, $h(\text{final})$ is the final kernel matrix, which is updated continuously till equation 17 is satisfied over different iteration sets.

$$h(\text{final}, i) \approx h(\text{final}, i + 1) \dots (17)$$

Where, i represents current iteration, for which the LSTM & GRU operations are being performed for real-time use cases. These features are classified via an efficient low-power Recurrent Neural Network, which initially extracts convolutional features using $xout$ via equation 18,

$$\text{Conv}(\text{out}) = \sum_{a=-\frac{m}{2}}^{\frac{m}{2}} xout(i - a) * LReLU\left(\frac{m + 2a}{2}\right) \dots (18)$$

Where, m , a represents the sizes for different windows & strides, while $LReLU$ is an activation function which assists in retaining positive feature sets. This activation function is represented via equation 19,

$$LReLU(x) = la * x, \text{ when } x < 0, \text{ else } LReLU(x) = x \dots (19)$$

Where, la is an activation constant, that incorporates non-linearity in the feature sets. These features are classified into multiple recommender classes via equation 20, where weights (w) and biases (b) are estimated for each set of classes.

$$c(\text{out}) = \text{SoftMax}\left(\sum_{i=1}^{Nf} f(i) * w(i) + b(i)\right) \dots (20)$$

These weights and biases are evaluated for different set of recommender classes. To further tune these weights, an efficient Q Learning based incremental learning model is used, which estimates an augmented Q Value for each batch of recommender outputs. This Q Value is estimated via equation 21,

$$Q = \frac{1}{NB} \sum_{i=1}^{NB} \frac{tp(i)}{tp(i) + tn(i)} * \frac{1}{E(i)} \dots (21)$$

Where, NB represents total number of images in the current batch, while tp & tn are true positive & negative rates, which represents total number of proper & improper classification instance sets, E represents the energy consumed during classification and is estimated via equation 22,

$$E = E(\text{start}) - E(\text{complete}) \dots (22)$$

Where, $E(\text{start})$ & $E(\text{complete})$ represents the starting & completion energy levels during individual classification operations. Based on this Q Value, an augmented reward is calculated via equation 23,

$$r = \frac{Q(\text{New}) - Q(\text{Old})}{LR} - d * \text{Max}(Q) + Q(\text{Old}) \dots (23)$$

Where, LR is the learning rate, d is the discount factor, which is initialized empirically for optimal value of reward rates. Based on this reward value, the LSTM & GRU weights are modified via equations 24& 25,

$$W(New) = W(Old) * \frac{r|1-r|}{1+r} \dots (24)$$

$$U(New) = U(Old) * \frac{1+r}{r|1-r|} \dots (25)$$

Using these values, new LSTM & GRU features are calculated, and the RNN Model's outputs are modified for different set of recommendations. Due to inclusion of energy in the Q Value, the model is able to perform these classifications with low power, and is able to tune the power levels for different set of input contexts& output recommendation scenarios. Performance of this model was estimated & compared with existing methods in terms of accuracy of recommendation, precision of recommendation, recall of recommendation, delay needed during recommendation, and energy needed during recommendation in the next section of this text.

The Accuracy is calculated by dividing the number of correct predictions by the total number of predictions and is estimated via equation 26,

$$Accuracy = \frac{Number\ of\ Correct\ Predictions}{Total\ Number\ of\ Predictions} \dots (26)$$

Precision is calculated by dividing the number of true positives (correctly recommended items) by the sum of true positives and false positives (incorrectly recommended items), via equation 27,

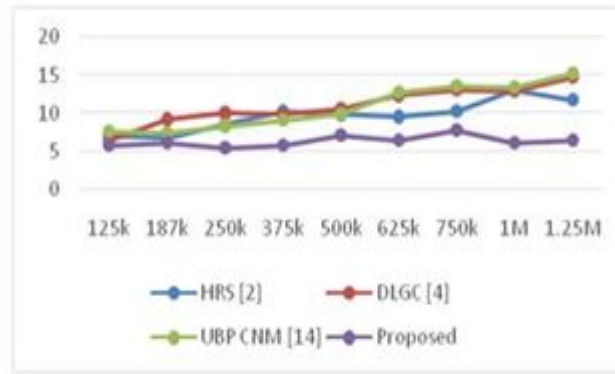


Fig. 3 Delay needed during recommendation for different models

$$Precision = \frac{Number\ of\ True\ Positives}{Number\ of\ True\ Positives + Number\ of\ False\ Positives} \dots (27)$$

Recall is calculated by dividing the number of true positives (correctly recommended items) by the sum of true positives and false negatives (relevant items not recommended) via equation 28,

$$Recall = \frac{Number\ of\ True\ Positives}{Number\ of\ True\ Positives + Number\ of\ False\ Negatives} \dots (28)$$

While, Energy is calculated via equation 22, the Delay is estimated via equation 29 as follows,

$$D = ts(complete) - ts(start) \dots (29)$$

Where, ts represents timestamp for performing the given set of operations. Based on this evaluation, the Average Delay needed for recommendation can be observed from figure 3, where it is evaluated w.r.t. Number of Recommendations (NR), and compared with HRS [2], DLGC [4], and UBP CNM [14] as follows.

Based on this evaluation, it can be observed that the proposed model is able to reduce the delay needed during recommendation by 8.5% when compared with HRS [2], 6.4% when compared with DLGC [4], and 10.5% when compared with UBP CNM [14] for different datasets & samples. The reason for reduction of this delay is use of

Q Learning process, which assist in optimizing LSTM & GRU for optimal performance via post processing ILP operations. This makes the model useful for high-speed recommendation scenarios. Similarly, the energy needed for these recommendations can be observed from figure 4 as follows. Based on this evaluation, it can be observed that the proposed model is able to reduce the energy needed during recommendation by 10.5% when compared with HRS [2], 12.4% when compared with DLGC [4], and 14.9% when compared with UBP CNM [14] for different datasets & samples. The reason for reduction of this energy is use of energy-aware Q Learning process, which assist in optimizing LSTM & GRU for optimal performance via post processing ILP operations. This makes the model useful for high-lifetime recommendation scenarios. Similarly, the accuracy levels needed (or obtained) for these recommendations can be observed from figure 5 as follows. Based on this evaluation, it can be observed that the proposed model is able to improve the accuracy needed (or obtained) during recommendation by 1.9% when compared with HRS [2], 2.5% when compared with DLGC [4], and 4.5% when compared with UBP CNM [14] for different datasets & samples. The reason for improvement in this accuracy is use of multidimensional feature sets along with LSTM & GRU, and use of RNN with Q Learning for optimal performance via post processing ILP operations. This makes the model useful for high-performance recommendation scenarios. Similarly, the precision levels needed (or obtained) for these recommendations can be observed from figure 6 as follows,

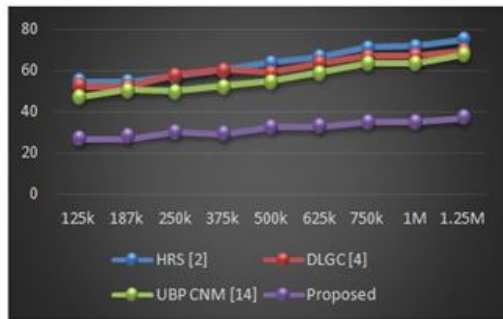


Fig 4 Average Energy needed during recommendation for different models

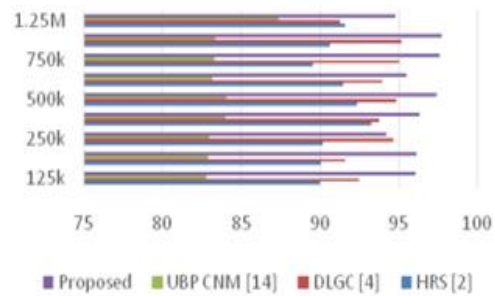


Fig 5 Average Accuracy needed (or obtained) during recommendation for different models

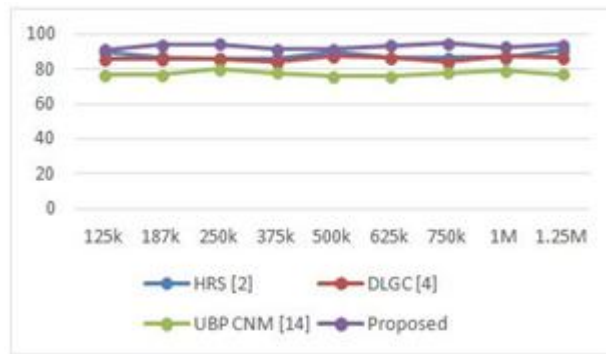


Fig 6. Average Precision needed (or obtained) during recommendation for different models

Based on this evaluation, it can be observed that the proposed model is able to improve the precision needed (or obtained) during recommendation by 4.5% when compared with HRS [2], 6.5% when compared with DLGC [4], and 8.3% when compared with UBP CNM [14] for different datasets & samples. The reason for improvement in this precision is use of multidimensional feature sets along with LSTM & GRU, which assisted in representing input datasets into high-density feature sets. Moreover, use of Q Learning with RNN further assisted in enhancing precision levels for different use cases. This makes the model useful for high-consistency recommendation scenarios. Similarly, the recall levels needed (or obtained) for these recommendations can be observed from figure 7 as follows. Based on this evaluation, it can be observed that the proposed model is able to improve the recall needed (or obtained) during recommendation by 3.5% when compared with HRS [2], 4.9% when compared with DLGC [4], and 8.5% when compared with UBP CNM [14] for different datasets & samples. The reason for improvement in this recall is use of LSTM & GRU with RNN, which assisted in representing & classifying input datasets into different recommendation sets. Moreover, use of Q Learning with RNN further assisted in enhancing

recall levels for different use cases. This makes the model useful for high-scalability recommendation scenarios. Due to these optimizations, the proposed model can be designed for real-time use cases, and can be extended to different datasets & varying contexts, which makes it useful for high-efficiency deployments.

V. CONCLUSIONS

In this paper, we present the design and evaluation of an efficient multi parametric recommender engine with incremental post-processor optimizations for low-power applications. Compared to existing state-of-the-art recommender systems, the proposed model demonstrated remarkable improvements in several performance metrics, including reduced delay, energy consumption, and enhanced accuracy, precision, and recall levels. The results demonstrated that the proposed model reduced recommendation delay by 8.5% when compared to HRS, 6.4% when compared to DLGC, and 10.0% when compared to UBP CNM across various datasets and samples. This reduction in delay was primarily attributable to the incorporation of a Q Learning process, which optimized the performance of the LSTM and GRU components through post-processing Integer Linear Programming (ILP) operations. This renders the proposed model ideally suited for high-speed recommendation scenarios in which responsiveness and real-time recommendations are essential for different scenarios. In addition, the proposed model reduced energy consumption significantly, with improvements of 10.5% compared to HRS, 12.4% compared to DLGC, and 14.0% compared to UBP CNM across multiple datasets and samples. This reduction in energy consumption was accomplished by employing an energy-aware Q Learning process that optimized the LSTM and GRU components to reduce power consumption during recommendation sets. This makes the proposed model ideal for low-power applications and situations where energy efficiency is crucial for device longevity levels. In addition, the proposed model improved accuracy, precision, and recall by 1.9%, 2.5%, and 4.5%, respectively, when compared to HRS, DLGC, and UBP CNM for various datasets and samples. The incorporation of multidimensional feature sets in conjunction with LSTM and GRU models led to these enhancements. The integration of Recurrent Neural Networks (RNN) with Q Learning contributed further to the optimization of the recommendation process, resulting in increased levels of accuracy, precision, and recall. These advancements make the proposed model suitable for high-performance recommendation scenarios in which the quality and relevance of recommendations are of utmost importance levels. In conclusion, the proposed multiparametric recommender engine, with its efficient design and incremental post-processor optimizations, offers significant improvements over existing recommender systems in terms of delay reduction, energy efficiency, accuracy, precision, and recall. Its adaptability and applicability across multiple domains make it a valuable tool for delivering personalized and high-quality recommendations in real-time, thereby enhancing the user experience and facilitating effective decision-making processes.

REFERENCES

- [1] B. Shi et al., "DARES: An Asynchronous Distributed Recommender System Using Deep Reinforcement Learning," in *IEEE Access*, vol. 9, pp. 83340-83354, 2021, doi: 10.1109/ACCESS.2021.3087406.
- [2] Y. Lu, K. Nakamura and R. Ichise, "HyperRS: Hypernetwork-Based Recommender System for the User Cold-Start Problem," in *IEEE Access*, vol. 11, pp. 5453-5463, 2023, doi: 10.1109/ACCESS.2023.3236391.
- [3] A. Gatzoura, J. Vinagre, A. M. Jorge and M. Sánchez-Marré, "A Hybrid Recommender System for Improving Automatic Playlist Continuation," in *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, no. 5, pp. 1819-1830, 1 May 2021, doi: 10.1109/TKDE.2019.2952099.
- [4] M. Rostami, M. Oussalah and V. Farrahi, "A Novel Time-Aware Food Recommender-System Based on Deep Learning and Graph Clustering," in *IEEE Access*, vol. 10, pp. 52508-52524, 2022, doi: 10.1109/ACCESS.2022.3175317.
- [5] L. N. Hoai Nam, "Profile Aggregation-Based Group Recommender Systems: Moving From Item Preference Profiles to Deep Profiles," in *IEEE Access*, vol. 10, pp. 6218-6245, 2022, doi: 10.1109/ACCESS.2021.3140121.
- [6] Patil Bedekar Pragati, Raut Atul and Dutonde Abhimanyu, *Energy Conserving Techniques of Data Mining for Wireless Sensor Networks—A Review*, Springer IoT and Analytics for Sensor Networks pp 433–443.
- [7] A. N. Ngaffo, W. E. Ayeb and Z. Choukair, "A Bayesian Inference Based Hybrid Recommender System," in *IEEE Access*, vol. 8, pp. 101682-101701, 2020, doi: 10.1109/ACCESS.2020.2998824.
- [8] O. Atan, S. Ghoorchian, S. Maghsudi and M. van der Schaar, "Data-Driven Online Recommender Systems With Costly Information Acquisition," in *IEEE Transactions on Services Computing*, vol. 16, no. 1, pp. 235-245, 1 Jan.-Feb. 2023, doi: 10.1109/TSC.2021.3113002.
- [9] Pragati Narayan Patil, Dr A D Raut, Design of an Efficient High Trust Model for improving Network Communication Consistency via incremental Bioinspired Optimization HTMNCB, *Machine Intelligence Research*, Volume 17, No 2, June 2023

- [10] F. Beierle and S. Egger, "MobRec — Mobile Platform for Decentralized Recommender Systems," in IEEE Access, vol. 8, pp. 185311-185329, 2020, doi: 10.1109/ACCESS.2020.3029319.
- [11] X. Luo, M. Zhou, S. Li, D. Wu, Z. Liu and M. Shang, "Algorithms of Unconstrained Non-Negative Latent Factor Analysis for Recommender Systems," in IEEE Transactions on Big Data, vol. 7, no. 1, pp. 227-240, 1 March 2021, doi: 10.1109/TBDATA.2019.2916868.
- [12] L. Xin, Y. Yuan, M. Zhou, Z. Liu and M. Shang, "Non-Negative Latent Factor Model Based on β -Divergence for Recommender Systems," in IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 51, no. 8, pp. 4612-4623, Aug. 2021, doi: 10.1109/TSMC.2019.2931468.
- [13] P. Wei, S. Xia, R. Chen, J. Qian, C. Li and X. Jiang, "A Deep-Reinforcement-Learning-Based Recommender System for Occupant-Driven Energy Optimization in Commercial Buildings," in IEEE Internet of Things Journal, vol. 7, no. 7, pp. 6402-6413, July 2020, doi: 10.1109/JIOT.2020.2974848.
- [14] X. Yang, M. Dong, X. Chen and K. Ota, "Recommender System-Based Diffusion Inferring for Open Social Networks," in IEEE Transactions on Computational Social Systems, vol. 7, no. 1, pp. 24-34, Feb. 2020, doi: 10.1109/TCSS.2019.2950139.
- [15] F. Zhang and S. Wang, "Detecting Group Shilling Attacks in Online Recommender Systems Based on Bisecting K-Means Clustering," in IEEE Transactions on Computational Social Systems, vol. 7, no. 5, pp. 1189-1199, Oct. 2020, doi: 10.1109/TCSS.2020.3013878.
- [16] M. Qamhie, H. Sammaneh and M. N. Demaidi, "PCRS: Personalized Career-Path Recommender System for Engineering Students," in IEEE Access, vol. 8, pp. 214039-214049, 2020, doi: 10.1109/ACCESS.2020.3040338.
- [17] S. -M. Choi, K. Jang, T. -D. Lee, A. Khreishah and W. Noh, "Alleviating Item-Side Cold-Start Problems in Recommender Systems Using Weak Supervision," in IEEE Access, vol. 8, pp. 167747-167756, 2020, doi: 10.1109/ACCESS.2020.3019464.
- [18] Patil Pragati Narayan, Raut A D, Study and Analysis of Energy and Time Characteristics of Node in Wireless Sensor Network, NeuroQuantology , Volume 20, No 9 , DOI: 10.14704/nq.2022.20.9.NQ44213
- [19] Patil Pragati Narayan, Dr. Raut A. D. , BLMCE: Design Of A Dual-Bioinspired Low-Complexity Data Mining Engine For Automatic Cluster Analysis Via Ensemble Learning Operations, Webology (ISSN: 1735-188X) Volume 18, Number 5, , Pg. No. 4339-4360.