Improving Consumer Experience with Pre-purify Temporal-decay Memory-based Collaborative Filtering Recommendation for Graduate School Application

Jinfeng Xu, *Student Member, IEEE,* Zheyu Chen, Zixiao Ma, Jiyi Liu, and Edith C. H. Ngai*

***Abstract*—The Internet is booming with information, and it has become especially difficult for consumers to sift through the information. Recommendation systems can effectively enhance the consumer experience. However, model-based recommendation systems require sufficient training data, so they perform poorly in small-scale recommendation scenarios such as graduate school recommendation. To this end, we focus on online recommendation in graduate school application scenarios. We propose a Pre-purify Temporal-decay Memory-based Collaborative Filtering model called PTMCF, which firstly improves the data quality based on the users' background information by pre-purifying the data to compensate for the poor performance caused by the small dataset. At the same time, considering that user preferences and the importance of information are constantly changing, we propose incorporating Newton's Law of Cooling when constructing the user-item scoring matrix to assign time-based weights. Experiments on a dataset collected from real-world questionnaires show that pre-purify and temporal-decay effectively improve recommendation quality and mitigate the impact of data sparsity on memory-based collaborative filtering.**

***Index Terms*—Intelligent Online Recommendation System, Collaborative Filtering, Preprocessing, k-Nearest Neighbors.**

I. INTRODUCTION

Consumers need personalized recommendations to find suitable information as the variety and volume of information explode. Personalized recommendations are ubiquitous and applied to many online services such as e-commerce [1]–[3], industrial [4], [5], and social media [6]. Although there are many general model-based recommendation systems, they require too much cost, which is unsuitable for all consumer scenarios. In today's fiercely competitive academic landscape, pursuing higher education has become increasingly intricate, and selecting a graduate school program is a pivotal decision that can influence one's academic future and professional trajectory [7]. With many choices available, consumers often find themselves overwhelmed by the vast array of programs, institutions, and disciplines. To address this challenge, an efficient and accurate recommendation algorithm about graduate schools effectively alleviates the information explosion that plagues students' choice of graduate schools [8].

The key to personalized recommendation systems is to model user preferences for items based on their past interactions (e.g., ratings and clicks), known as collaborative filtering [9], [10]. With the rapid emergence of deep learning, many recommendation algorithms based on deep learning achieve much better recommendation effects and accuracy than traditional collaborative filtering. These include the work of using neural networks instead of the dot product to learn higher-order feature combinations [11]–[13], the work of combining with reinforcement learning in order to respond to information changes on time [14]–[16], and the work of integrating Attention mechanism into user interest modeling to improve the degree of grasping the user's interests [17]–[19]. Recently, some notable work has been done on generalized knowledge-assisted recommendation using large language models [20]–[22].

Despite their remarkable effectiveness, the cost of model training and the required data are also rising significantly. Complex model-based recommendation systems are not suitable for small-scale recommendation scenarios. A small amount of training data is not enough to train the model well, and they are prone to over-fitting problems [23]. More specifically, in this scenario of graduate school recommendations, access to large amounts of data is difficult because the matter is inherently relatively private. For small amounts of data, model-based recommendation systems do not perform very satisfactorily, especially for this scenario where user features have an enormous impact, and often, model-based recommendation systems trained on small data will achieve unacceptable performance [24].

One of the essential difficulties facing graduate school recommendations is the dataset size. Collecting a sufficiently large dataset is too harsh, so one considerable method is to improve the data quality and thus mitigate the effects caused by the lack of data. Applying to graduate programs requires consideration of the student's interest and the program's feasibility. Therefore, pre-purifying the data based

Jinfeng Xu is with the Department of Electrical and Electronic Engineering, The University of Hong Kong, Hong Kong, China (e-mail: jinfeng@connect.hku.hk).

Zheyu Chen is with the Department of Electrical and Electronic Engineering, The Hong Kong Polytechnic University, Hong Kong, China (e-mail: zheyu.chen@connect.polyu.hk).

Zixiao Ma is with the Department of Electrical and Computer Engineering, Duke University, USA (e-mail: zixiao.ma@duke.edu).

Jiyi Liu is with the School of Software Engineering, Sun Yat-Sen University, Zhuhai, China (e-mail: liujy557@mail2.sysu.edu.cn).

Edith C. H. Ngai (*corresponding author) is with the Department of Electrical and Electronic Engineering, The University of Hong Kong, Hong Kong, China (e-mail: chngai@eee.hku.hk).

Manuscript received XXXX X, XXXX; revised XXXX XX, XXXX.

on the user's background can improve the quality of the data while giving more consideration to the feasibility of the program. Using memory-based collaborative filtering techniques to make recommendations based on filtered data and considering timeliness, more popular and feasible graduate school recommendations can be obtained. The graduate school recommendation process based on pre-purify is demonstrated in Fig 1.

Memory-based collaborative filtering is still famous for many small data volume scenarios. More recent work focuses on optimizing similarity computation methods [25]–[27]. When the data volume is not massive enough, relying on data quality improvement to optimize the recommendation effect is a good choice, especially for scenarios with prominent user characteristics. The small data volume allows the computational cost to be not too worried, so more consideration can be given to enhancing the personalization of the algorithm to achieve better recommendations. In this work, we filter out the worthless data by pre-purifying the data based on user characteristics to enhance the degree of personalization and alleviate the widespread problem of data sparsity, a recommendation algorithm, to a certain extent.

In addition, user preferences and the importance of information are constantly changing and time-sensitive. Many related works dynamize temporality, adding temporal weights when constructing user-item scoring matrix [28], or adding time weights when computing user similarity [29]. Our work draws inspiration from Newton's Law of Cooling [30], from which we extract the core concept and apply it in our methodology. Specifically, we integrate time-based weights based on Newton's cooling coefficient in the creation of the user-item scoring matrix. This augmentation significantly enhances the accuracy of recommendations. This law describes the physical property of heat transfer from an object with a higher temperature to its surrounding medium.

To solve various problems in the graduate school recommendation scenario, we propose a novel improved system named the **P**re-purify **T**emporal-decay **M**emory-based **C**ollaborative **F**iltering Recommendation System (PTMCF). It pre-purifies the data based on user features and considers the temporal factor when computing the user-item scoring matrix. It can better capture more personalized user interests and ensure the feasibility of recommendations. Subsequent experiments will demonstrate PTMCF's effectiveness on real-world datasets collected through questionnaires. This work is an expanded version of our previous conference work [31]. The main contributions are:

- We propose a pre-purify method based on user features, which can effectively improve data quality.
- We propose a temporal-decay algorithm inspired by Newton's law of cooling that performs well in Memory-based Collaborative Filtering.
- We conduct extensive experiments on the real-world Graduate School dataset to demonstrate the validity of our PTMCF method.

The specific structure of this paper is: Section II discusses related works; Section III introduces the proposed PTMCF recommendation model; Section IV conducts experimental verification for the PTMCF recommended model; Section V summarizes the contents of this work.

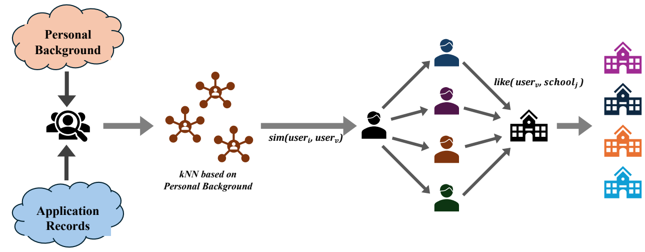


Fig. 1. The process of graduate school recommendation.

II. Related Work

A. Collaborative Filtering

Collaborative filtering (CF) is a widely employed technique in modern recommendation systems, as indicated by several seminal references [32], [33]. CF falls into two main categories: memory-based, offering greater personalization, and model-based, suitable for large datasets. Memory-based CF generates predictions by using the entire user-item interactions, which has a wide range of applications in scenarios with small amounts of data. Compared to model-based CF, it is more explanatory and easy to extend because it does not require training and generates predictions directly from historical interaction data. In recent years, Lima et al. [34] propose a new landmark space that alleviates the problem of similarity sensitivity in memory-based CF. SSCF [35] proposes a similarity computation method that allows negative values, which improves the recommendation accuracy while maintaining the strong interpretability of memory-based CF. The surge in available data has fueled interest in model-based CF. Typically, these models use user and item embeddings, iteratively refining them by reconstructing past interactions. Early models like Matrix Factorization (MF) project user or item IDs into embeddings, while newer ones like NCF [11] and LRML [36] combine embeddings with neural networks for improved interaction modeling. Some CF methods consider historical items as user features for better characterization, e.g., FISM [37] and SVD++ [38]. Recent research introduces attentional mechanisms to automatically weigh historical item contributions, as seen in ACF [39] and NAIS [40]. Additionally, incorporating the user-item graph structure in recommendation systems is popular. Methods like ItemRank [41] use label propagation to encourage similar preferences among connected nodes. Adaptations like NGCF [12], GC-MC [42], PinSage [33], and LightGCN [43] apply Graph Convolutional Networks (GCN) [44] to capture collaborative signals from high-order neighbors. However, the requirement of model-based collaborative filtering for training data limits its applicability on smaller datasets and may reduce personalization capability. In this case, memory-based CF is a more appropriate choice.

B. Data Preprocessing for Recommendation

Data preprocessing is a pivotal step in developing any recommendation system as it enhances the quality and reliability of the recommendations. Data processing includes but is not limited to, data cleaning and feature engineering. Data preprocessing is a paramount component in recommendation systems, meticulously refining data to yield precise, personalized recommendations across a myriad of domains. Early approaches, such as feature selection in Deep Crossing [45], employ a concise vector representation for individual features and leverage companion features to encapsulate campaign-related statistics in their recommendation systems succinctly. Other works, such as NCF [11], implement data purifying by transforming explicit feedback datasets into implicit preference data by marking entries as binary indicators of user interactions and meticulously filtering highly sparse datasets to retain only users with substantial interactions, thereby addressing sparsity and enabling more robust evaluations of collaborative filtering algorithms. Some data processing methods underscore the significance of temporal dynamics in user interactions. Notably, DIEN [18] engages with datasets by chronologically structuring user reviews and employing temporally defined windows in click logs, leveraging historical patterns to foresee future user behaviors. Recent advancements have honed in on optimizing training samples by integrating exposures, ranking candidates, and pre-ranking candidates. Building on this approach, ASMOL [46] elucidates the profound impact of training data compositions, specifically exposures and ranking candidates, on model efficacy. Their findings underscored that an over-reliance on exposures compromised recommendation quality, illuminating the paramount importance of a reasonable balance in training data for optimized performance metrics. These methods have been very successful, but they are too general. For particular scenarios, such as the graduate school recommendation system scenario, user features such as students' backgrounds play a significant role. However, model-based CF learns these features more as latent features, which makes it hard to learn insufficiently from small datasets. We propose a method that prioritizes pre-purify based on user features so that user features can be more fully learned.

III. METHODOLOGY

In this section, we will introduce PTMCF in detail. First, we will introduce the overall architecture. Secondly, we will introduce the pre-purify for k-Nearest Neighbors (kNN) [47] based on essential features, and then we will introduce the temporal decay algorithm based on Newton's cooling factor. Finally, we will introduce the idea of combining this algorithm in Model-based Collaborative filtering.

A. PTMCF Architecture

In pursuit of furnishing more pertinent and superior recommendations of graduate schools for college students, this paper introduces novel recommendation algorithms tailored for small-scale datasets characterized by varying feature significance. The initial phase entails data pre-purify through the employment of the kNN algorithm, wherein priority is accorded to important user features. Subsequently, in the second phase, the weighting of school ratings undergoes a temporal decay governed by Newton's cooling coefficient while constructing the user-school matrix within the framework of memory-based collaborative filtering. Concurrently, user similarity is determined by assessing the congruence in application information amongst users. The third and final phase comprises the computation, based on the user similarity ranking and the user-school rating matrix, followed by sorting to yield the ultimate recommendation list. The structure of PTMCF is demonstrated in Fig 2.

B. Pre-Purify

In recommendation systems, we often encounter two situations. Firstly, some users have only interacted with a few products, resulting in data sparsity. Secondly, new products lack user behavior data, leading to the cold start problem. Compared with model-based CF, traditional memory-based CF has more significant advantages for cold start, and for smaller datasets, memory-based CF can capture more critical information than model-based CF. However, for data sparsity, memory-based CF suffers more. For this reason, we pre-purify the data based on significant user features, which can effectively mitigate data sparsity and significantly improve accuracy.

The structure of pre-purify is demonstrated in Fig 3. This progress can highlight the role and influence of essential user characteristics, improving recommendation accuracy and effectiveness. On the other hand, data pre-purification can effectively solve the problem of data sparsity.

First, we focus on the user's own characteristics in the pre-purify phase and do not consider school-related information. This is because for applying to graduate schools, a student's basic background plays a huge role; an extremely talented student will not give much consideration to a relatively average school, and an average student will not try to apply to the top schools. Therefore, for users with widely different backgrounds, too much information interaction may be a negative optimization for accuracy. In this regard, we construct a distance function based on the user's background, and different important features of the user are given different weights for weighting calculation, and the calculation formula is:

$$Score(u_i) = \sum_{j=1}^{n} M_j \cdot Norm(F_{ij}), \qquad (1)$$

where F_{ij} is the j^{th} features of $user_i$, and M_j is the weight of user feature j.

In order to avoid individual erroneous data leading to disproportionate effects, we regularize F_{ij} through the tanh():

$$\hat{F}_i = \frac{1}{n}\sum_{j=1}^{n} F_{ij}, \quad Norm(F_{ij}) = tanh(F_{ij} - \hat{F}_i), \qquad (2)$$

where $\hat{F}_i$ is the average of all features of $user_i$.

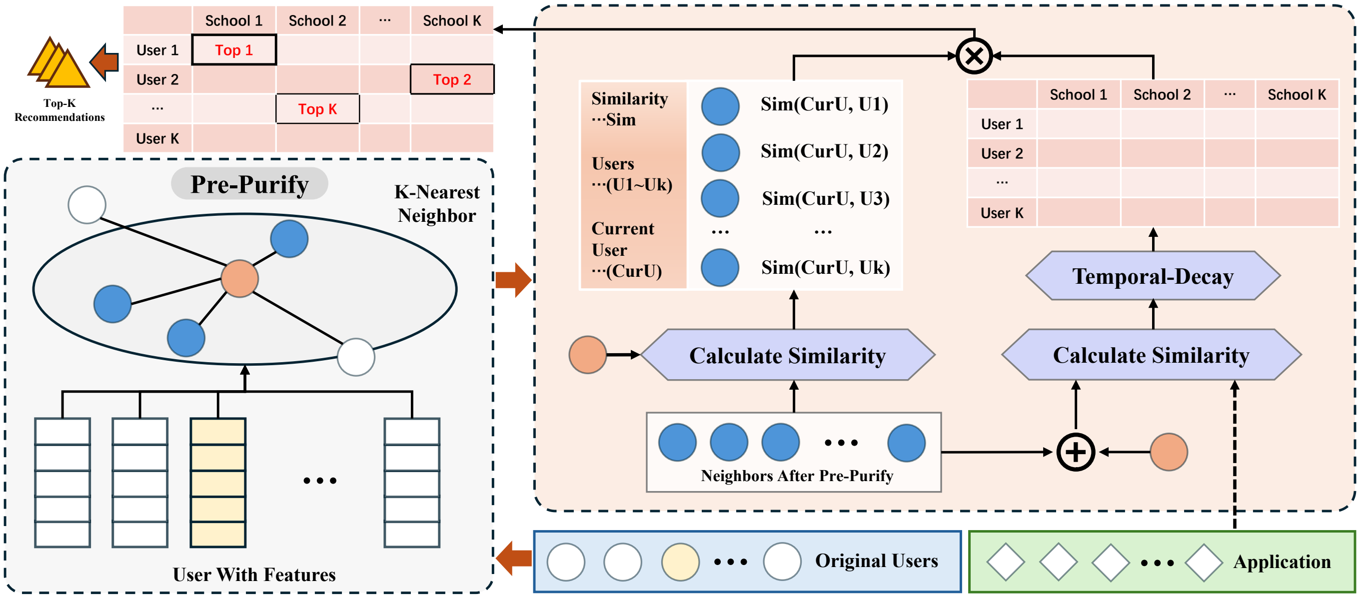


Fig. 2. The proposed PTMCF's architecture for intelligent recommendation of graduate schools. PTMCF initiates the pre-purify of user data by employing kNN techniques, wherein user data is systematically condensed, prioritizing key user features. Then, it performs user similarity computation based on users' applications and constructs a user-school scoring matrix based on users and their application information. During the matrix construction, PTMCF incorporates Newton's Law of Cooling, which serves as a mechanism for weight decay over time. The negative impact of data sparsity on memory-based collaborative filtering is effectively reduced. After that, PTMCF calculates the final recommendation list by sorting the results of multiplying the user similarity list by the user-school scoring matrix.

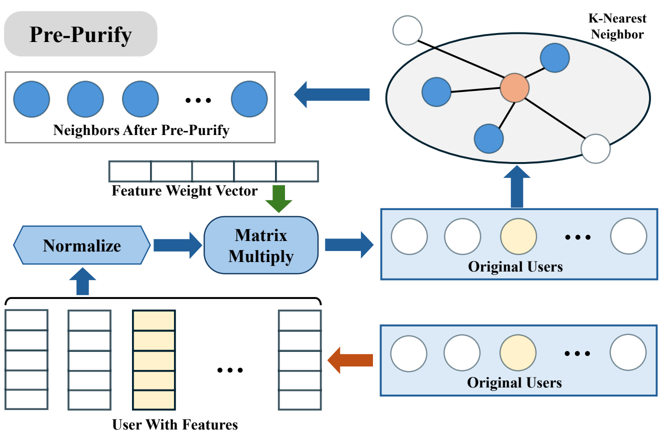


Fig. 3. Pre-purify structure diagram. It first normalizes user features and calculates the distance with different weights. Then it remains top K users through kNN.

C. Temporal-Decay

The timeliness of information should decrease as the freshness of information decreases, with newer information often meaning more valuable. The temporal decay mechanism can effectively reduce the attention to outdated information and increase the attention to fresh information. Therefore, this has been introduced into product recommendation scenarios, especially in time-dependent scenarios, where a reasonable temporal decay mechanism can significantly improve recommendation accuracy. In the past, there have also been many works on Memory-based CF that incorporate temporal decay. However, most of them consider temporal decay during the calculation of user similarity [48]–[50].

$$sim(u_a, u_b)^{TD} = sim(u_a, u_b) \cdot f_{TD}, \tag{3}$$

$$f_{TD} = func(|T_{u_a} - T_{u_b}|^{\omega}), \tag{4}$$

where $sim(u_a, u_b)^{TD}$ is the user similarity with temporal decay, f_{TD} is the temporal decay rate calculated according to different formulas, and ω is the temporal decay factor.

However, this is unwise, and the user-item ratings are affected by decay, not the similarity between users. Therefore, considering temporal decay in constructing the user-item scoring matrix can incorporate the timeliness factor more effectively. Some related works consider timeliness in constructing the user-item scoring matrix. However, they are more likely to use linear or segmented decay [51], [52], in some occasions with high timeliness requirements, a gap of one month may mean that the information ranges from extremely useful to completely useless. Therefore, we propose a decay method based on Newton's cooling law in constructing a user-school fusion temporal decay mechanism, which effectively improves the role of timeliness in the recommendation and enhances the accuracy of the recommendation system.

Newton's law of cooling refers to the law that the temperature of an object is higher than that of the surrounding medium. When there is a temperature difference between the surface of the object and the ambient temperature, the amount of heat lost per unit area is proportional to the temperature difference.

$$\frac{dT(t)}{dt} = -\alpha(T(t) - H), \tag{5}$$

where T(t) represents the temperature of the object at time t, while H denotes the ambient room temperature, the parameter

α is a user-defined ratio coefficient that signifies the rate at which the object's temperature changes, typically with a value of $\alpha > 0$. To further understand Newton's law of cooling, we integrate the above differential equation:

$$\int \frac{dT(t)}{T(t) - H} = \int (-\alpha) dt, \tag{6}$$

$$T(t) - H = C \cdot e^{-\alpha t}. \tag{7}$$

Based on the initial conditions, we can solve the above equation.

$$T(t_0) = H + C \cdot e^{-\alpha t_0}, \tag{8}$$

$$T(t_0) = H + (T(t_0) - H) \cdot e^{-\alpha(t_0 - t)}, \tag{9}$$

$$T(t) = T(t_0) \cdot e^{-\alpha(t_0 - t)}, \tag{10}$$

$$Suppose : T(t_0) = 1, \tag{11}$$

$$T(t) = e^{-\alpha(t_0 - t)}, \tag{12}$$

where t_0 is the initial time.

Based on this formula, we constructed the user-school scoring matrix by multiplying each application message for each user by an additional weight.

We can get the total score between each user and school:

$$r_{u_i,s_j} = \sum_{k=1}^{n} r_{u_i,a_k} \cdot T(\hat{t_{u_i}}) \quad a_k \in u_i \cap s_j, \tag{13}$$

$$\hat{t_{u_i}} = \frac{\sum_{k=1}^{n} t_{u_i,a_k}}{n}, \tag{14}$$

$$r_{u_i,a_k} = \frac{\sum_{u_v \in N_u(k)} sim(u_i, u_v)^{TD} \times (r_{u_v,a_k} - \bar{r_{u_v}})}{\sum_{u_v \in N_u(k)} |sim(u_i, u_v)^{TD}|}, \tag{15}$$

where r_{u_i,s_j} is the total score between u_i and s_j, r_{u_i,a_k} is a score between u_i and a_k, and $\hat{t_{u_i}}$ means the average reporting time of all applications for u_i. $N_u(k)$ is the set of users similar to u_i who have rated a_k, $\bar{r_{u_v}}$ is the average ratings for u_v, and r_{u_v,a_k} is the ratings between u_v and a_k.

IV. Experiment

In this section, we conducted experiments to address the following research questions:

RQ1: What is the performance of Pre-purify in different purification ratios when utilized for recommendation tasks?

RQ2: How does the temporal-decay method based on Newton's cooling coefficient, as proposed in the paper, compare in performance to methods from other relevant temporal-decay methods for memory-based CF?

RQ3: Does the proposed PTMCF method demonstrate improvement under various similarity calculations, and which similarity method performs best?

RQ4: How does PTMCF perform on strong feature datasets compared to the baseline model?

Next, we will introduce the dataset and experimental environment and answer the four research questions in part C.

A. Dataset and Experimental Environment

To evaluate our proposed PTMCF in the top-N item recommendation for graduate school application scenario, we conduct extensive experiments on our Graduate School dataset. This dataset is collected through a questionnaire from students who applied in the years 2022 and 2023. The key statistics of this dataset are shown in Table I.

This study is conducted in a specific environment using Python 3.9 as the programming language and PyTorch 2.1.0 as the deep learning framework. The GPU used in the study is NVIDIA RTX 3090.

B. Baselines

This paper introduces a weight decay method based on the Newton cooling coefficient regarding the temporal decay methods.

The baselines used for RQ1-4: To demonstrate the effectiveness, we compare our proposed PTMCF with the following methods:

- **Normal**: Memory-based collaboration filtering without temporal decay. For this method, it replaces the $T(\hat{t_{u_i}})$ by constant number 1 in Eq 13.
- **Exponential** and **Power**: These methods apply weight decay based on temporal when calculating data similarity. For Exponential method, it replaces the $T(\hat{t_{u_i}})$ by $T(\hat{t_{u_i}}) = e^{\mu \hat{t_{u_i}}}$ in Eq 13, where μ = 0.2 is the optimal hyper-parameter. For Power method, it replaces the $T(\hat{t_{u_i}})$ by $T(\hat{t_{u_i}}) = \hat{t_{u_i}}^{\omega}$ in Eq 13, where ω = 0.6 is the optimal hyper-parameter.
- **Linear** and **Segment**: These methods apply weight decay based on temporal when building the data model. For Linear method, it replaces the $T(\hat{t_{u_i}})$ by $T(\hat{t_{u_i}}) = \alpha \hat{t_{u_i}}$ in Eq 13, where α = 0.25 is the optimal hyper-parameter. For Segment method, it replaces the $T(\hat{t_{u_i}})$ by $T(\hat{t_{u_i}}) = \alpha_i \hat{t_{u_i}}$ in Eq 13, where α_i is the hyper-parameter in set $\{0.1, 0.2, 0.4, 0.8\}$ according to the freshness of the application.
- **BPR-MF [53]:** a recommendation algorithm based on probabilistic modeling and matrix factorization, optimizing ranking problems using Bayesian personalized ranking.
- **NCF [11]:** a recommendation model that combines matrix factorization with neural networks. It combines user and item features through neural networks to predict user interests.
- **LightGCN [43]:** a collaborative filtering model based on graph convolutional networks. It constructs a graph structure using user-item interactions and uses GCN to learn embedding vectors for users and items for making recommendations.
- **SASRec [54]:** a sequential recommendation model that employs a self-attention mechanism inspired by the Transformer architecture to capture long-term dependencies and complex patterns in user interaction sequences. It enables personalized recommendations by modeling users' sequential behavior.

- **User-based CF:** a traditional collaborative filtering approach recommends items to a target user based on user similarity without temporal decay. It identifies users similar to the target user and recommends items liked by those similar users.

TABLE I
STATISTICS OF THE EXPERIMENTED DATA.

Dataset	User #	Item #	Interaction #	Density
Graduate School	317	459	5345	0.0367

During the experiment, the researchers evaluated their algorithm on this real dataset. They iterated by selecting each user as the test user and calculated the corresponding evaluation metrics.

C. Experimental Metrics and Parameters Selection

In terms of evaluation metrics, the experiments utilized the following indicators:

- *Recall@10*: This metric represents the recall rate of the experiment at the top 10 results. The TP denotes True Positive and FN denotes False Negative. Formally:

$$Recall@10 = \frac{TP@10}{TP@10 + FN@10}. \quad (16)$$

- *MAP@10*: This measures the average precision of the experiment at the top 10 results. AP@N evaluates the performance of the algorithm for a single user, and MAP@N is the average AP@N of the algorithm for all users $|U|$. Formally:

$$AP@10 = \frac{1}{min(n,10)} \sum_{k=1}^{min(n,10)} P(k) \cdot rel(k), \quad (17)$$

$$MAP@10 = \frac{1}{|U|} \sum_{u=1}^{|U|} AP@10, \quad (18)$$

where n represents the amount of data. If it is less than ten, all data will be used for calculation. P(k) represents the precision of the first k items in the list, rel(k) represents whether the k-th item is relevant, the correlation is 1, and otherwise 0.

- *NDCG@10*: This assesses the precision of the query order at the top 10 results. Discounted cumulative gain (DCG) is a cumulative gain that takes into account the order of items. normalized discounted cumulative gain (NDCG) normalizes the DCG results to be between [0,1], and the closer it is to 1, the better the effect of the method. The normalization coefficient of NDCG is the ideal discounted cumulative gain (IDCG), which is the ideal perfect DCG. Formally:

$$DCG@10 = \sum_{i=1}^{10} \frac{rel_i}{log_2(i+1)}, \quad (19)$$

$$IDCG@10 = \sum_{i=1}^{|REL|} \frac{rel_i}{log_2(i+1)}, \quad (20)$$

$$NDCG@10 = \frac{DCG@10}{IDCG@10}, \quad (21)$$

where $\mathbf{rel}_i$ represents the true relevance score of the ith result. $|\mathbf{REL}|$ represents the number of results in the set composed of the top k results, sorted in descending order of actual relevance scores.

These metrics provide insights into the experiment's performance in terms of recall, average precision, and query order accuracy.

For RQ1-RQ3: Each experimental group employs distinct similarity calculation methods, different data purification rates, and varying temporal decay methods.

Because for memory-based models without pre-training, even small configuration changes can significantly impact the results, the differences in similarity calculation methods are relatively significant. Therefore, this experiment uses three different similarity calculation methods: Spearman Correlation [55], LogLikelihood [56], and Tanimoto Coefficient [57].

The Spearman Correlation measures the strength of association between ranked variables, and it can be formulated as:

$$\begin{aligned}\beta &= \frac{\sum_{i=1}^{n}(r(p_i)-\overline{r(p)})(r(q_i)-\overline{r(q)})}{\sqrt{\sum_{i=1}^{n}(r(p_i)-\overline{r(p)})^2 \cdot \sum_{i=1}^{n}(r(q_i)-\overline{r(q)})^2}} \\ &= 1 - \frac{6\sum_{i=1}^{n}(r(p_i)-r(q_i))^2}{n(n^2-1)},\end{aligned} \quad (22)$$

where the $r(p_i)$ and $r(q_i)$ denote the rank of p_i and q_i. The $\overline{r(p_i)}$ and $\overline{r(q_i)}$ denote mean rank of p_i and q_i. n is the number of pairs.

The LogLikelihood similarity method quantifies the fit of a model to data by assessing the likelihood of observing the data given the model, which can be formulated as:

$$\begin{aligned} E(row) &= e(k_{11},k_{12}) + e(k_{21},k_{22}), \\ E(column) &= e(k_{11},k_{21}) + e(k_{12},k_{22}), \\ E(matrix) &= e(k_{11},k_{12},k_{21},k_{22}), \\ sim(x,y) &= 2 \cdot (E(matrix) - E(row) - E(column)), \end{aligned} \quad (23)$$

where the symbols E() and e() denote the entropy of elements.

The Tanimoto Coefficient evaluates the similarity between sets by comparing their intersection to their union. Formally:

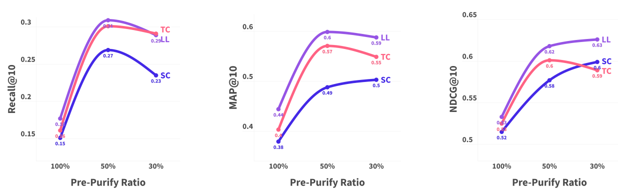
$$sim(x,y) = \frac{||x \cap y||}{||x|| + ||y|| - ||x \cap y||}. \quad (24)$$

Data purification ratio is a term proposed in this paper, which refers to the proportion of training data remaining after reduction by pre-purify technology. The experiment conducted data measurements at three typical values: 30%, 50%, and 100% (meaning no purification).

This article has been accepted for publication in IEEE Transactions on Consumer Electronics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCE.2024.3411875

TABLE II
PERFORMANCE COMPARISON OF DIFFERENT TEMPORAL DECAY METHODS.

Metric	Recall@10			MAP@10			NDCG@10		
Similarity	*SC*	*LL*	*TC*	*SC*	*LL*	*TC*	*SC*	*LL*	*TC*
Normal	0.269	0.309	0.301	0.488	0.599	0.571	0.577	0.618	0.601
Exponential	0.278	0.343	0.327	0.491	0.608	0.588	0.634	0.701	0.662
Power	0.278	0.343	0.327	0.496	0.609	0.584	0.633	0.702	0.662
Linear	<u>0.282</u>	<u>0.347</u>	<u>0.331</u>	0.505	<u>0.629</u>	<u>0.591</u>	<u>0.626</u>	<u>0.724</u>	<u>0.669</u>
Segment	<u>0.282</u>	<u>0.347</u>	<u>0.331</u>	<u>0.506</u>	0.609	<u>0.591</u>	<u>0.626</u>	0.702	<u>0.669</u>
Newton	**0.314**	**0.364**	**0.333**	**0.546**	**0.638**	**0.603**	**0.680**	**0.755**	**0.680**



(a) Recall@10 (b) MAP@10 (c) NDCG@10

Fig. 4. Metrics based on Normal.

TABLE III
PERFORMANCE COMPARISON OF DIFFERENT SIMILARITY CALCULATION METHODS.

Pre-Purify	Similarity	Metric	Normal	Newton
30%	*SC*	Recall@10	0.235	0.266
		MAP@10	0.503	0.519
		NDCG@10	0.599	0.641
	LL	Recall@10	0.289	**0.339**
		MAP@10	**0.588**	**0.612**
		NDCG@10	**0.626**	**0.716**
	TC	Recall@10	**0.291**	0.329
		MAP@10	0.549	0.593
		NDCG@10	0.589	0.696
50%	*SC*	Recall@10	0.269	0.314
		MAP@10	0.488	0.546
		NDCG@10	0.577	0.680
	LL	Recall@10	**0.309**	**0.364**
		MAP@10	**0.599**	**0.638**
		NDCG@10	**0.618**	**0.755**
	TC	Recall@10	0.301	0.333
		MAP@10	0.571	0.603
		NDCG@10	0.601	0.680
100%	*SC*	Recall@10	0.151	0.172
		MAP@10	0.379	0.455
		NDCG@10	0.515	0.572
	LL	Recall@10	**0.177**	**0.222**
		MAP@10	**0.444**	0.481
		NDCG@10	**0.533**	**0.593**
	TC	Recall@10	0.161	0.201
		MAP@10	0.403	**0.498**
		NDCG@10	0.525	0.589

D. Comparison, Analysis, and Interpretation

1) Temporal-Decay with Newton Cooling Algorithm (RQ1): To keep the comparison fair, we fixed the purification ratio at 50%. The results in Table II show that after considering temporal-decay, the recommendation performance achieves a significant improvement. **the data in bold** is the optimal result by comparing all temporal-decay methods with the same similarity calculation method, and <u>the data in underline</u> is a suboptimal result. The abbreviations in the table are explained as follows: SC: SpearmanCorrelation; LL: LogLikelihood; TC: TanimotoCoefficients. Specifically, decaying during the construction of the data model yielded better results than decaying during similarity calculation. Subsequently, experimental results show that our proposed Newton's cooling method achieves the best results in most cases.

2) Pre-Purify (RQ2): Fixed the similarity calculation method constant, experimental results indicated that a purification ratio of 50% yielded the best outcomes Fig 4. In the case of 100% (no purification), the performance of recommendation is suboptimal due to the data with high sparsity of data. In the case of 30%, we observe a deterioration in performance, suggesting that the dataset is too small, resulting in valuable data not being adequately learned. Furthermore, using the Tanimoto Coefficient similarity as a similarity algorithm performs consistently in different low purification rates. This result shows that the Tanimoto Coefficient similarity uses a more lenient standard for similarity calculation than other similarity calculation methods. Therefore, more data is retained for learning during the similarity calculation stage, to some extent compensating for data loss during pre-purification.

3) Best Similarity Method (RQ3): The results in Table III show that LogLikelihood similarity outperforms other similarity calculation methods under all purification ratios in most cases. **The data in bold** is the optimal result in each purification ratio.

Unlike Cosine similarity or Tanimoto similarity, the LogLikelihood similarity does not directly depend on how active a user is, which helps provide fair recommendations between active and less active users. In addition, because it calculates similarity from a probabilistic perspective, it can better capture the complex patterns of user interaction with items without being severely skewed by the most popular items or active users. It is more suitable for small-scale projects with high sparsity datasets.

4) Performance Comparison with Model-based CF Baselines (RQ4): To validate the superiority of our proposed method on the dataset in our graduate school recommendation scenario, we compared it with the modal-based CF models. We select the optimal hyper-parameters for our model, with a purification ratio of 50%, Newton cooling algorithm as the temporal decay method, and LogLikelihood as the similarity method. The experimental results are shown in Table IV, we also **blod** the optimal result, and <u>underline</u> the suboptimal result.

TABLE IV
PERFORMANCE COMPARISON OF MODAL-BASED CF METHODS.

Method	Recall@10	MAP@10	NDCG@10
BPR-MF	0.211	0.541	0.541
NCF	0.251	0.599	0.631
LightGCN	0.293	0.621	0.676
SASRec	0.193	0.483	0.537
PTMCF(Our)	**0.364**	**0.638**	**0.755**
%Improv.	24.23%	2.73%	11.69%
User-based CF	0.177	0.444	0.533
Normal	0.309	0.599	0.618
Linear	0.347	0.629	0.724
PTMCF(Our)	**0.364**	**0.638**	**0.755**
%Improv.	4.90%	1.43%	4.28%

Our PTMCF outperforms the seven baselines across all three evaluation metrics. PTMCF consistently demonstrates the best performance on our dataset. When compared with model-based CF, PTMCF outperforms the strongest baseline, LightGCN, with improvements of 24.23% in Recall@10, 2.73% in MAP@10, and 11.69% in NDCG@10. This is attributed to the computational demands of model-based collaborative filtering, which limit its applicability on small datasets and may reduce its level of personalization. Furthermore, when compared with memory-based methods, PTMCF also shows improvements with increases of 4.90%, 1.43%, and 4.28% in the respective metrics than the suboptimal method. We also find that there is a significant improvement compared with User-based CF, which is without pre-purify and temporal-decay. This enhancement could be attributed to the pre-purify process, which alleviates a significant portion of perturbation in the data, resulting in higher data quality. The advantage of our PTMCF lies in its ability to achieve good results on small datasets, as it focuses more on crucial features rather than relying on a large amount of interaction data. As a result, PTMCF achieves outstanding performance on the Graduate School dataset.

V. CONCLUSION

In this work, we propose a memory-based collaborative filtering algorithm, named PTMCF, which considers temporal decay and uses user high-impact features for data pre-purify to personalize the most suitable graduate school recommendation for a user based on their background and interactions with other users. In PTMCF, we perform data pre-purify based on user background before constructing the user-item scoring matrix, significantly improving the recommendation quality. The algorithm also introduces a temporal decay algorithm inspired by Newton's Law of Cooling, which allows the timeliness of the information to be fully taken into account, resulting in a significant improvement in the recommendation quality. PTMCF achieves promising results on a real-world Graduate School dataset collected through questionnaires. It is worth noting that PTMCF provides a high-performance recommendation for consumers, and requires fewer costs than most general recommendation systems.

REFERENCES

[1] K. Yoon, J. Lee, and M.-U. Kim, "Music recommendation system using emotion triggering low-level features," *IEEE Transactions on Consumer Electronics*, vol. 58, no. 2, pp. 612–618, 2012.

[2] N. K. Singh, D. S. Tomar, M. Shabaz, I. Keshta, M. Soni, D. R. Sahu, M. Bhende, A. K. Nandanwar, and G. Vishwakarma, "Self-attention mechanism based federated learning model for cross context recommendation system," *IEEE Transactions on Consumer Electronics*, vol. 70, no. 1, pp. 2687–2695, 2024.

[3] L. Li, L. Jia, and S. Al Otaibi, "Intelligent recommendation algorithm of consumer electronics products with graph embedding and multi-head self-attention in ioe," *IEEE Transactions on Consumer Electronics*, vol. 70, no. 1, pp. 1278–1285, 2024.

[4] X. Yu, Q. Peng, H. Lv, D. Zhan, Q. Hu, J. Du, D. Gong, and D. Sun, "A privacy-preserving cross-domain recommendation algorithm for industrial iot devices," *IEEE Transactions on Consumer Electronics*, vol. 70, no. 1, pp. 227–237, 2024.

[5] G. G. Devarajan, S. M. Nagarajan, A. Daniel, T. Vignesh, and R. Kaluri, "Consumer product recommendation system using adapted pso with federated learning method," *IEEE Transactions on Consumer Electronics*, vol. 70, no. 1, pp. 2708–2715, 2024.

[6] R. L. Rosa, D. Z. Rodriguez, and G. Bressan, "Music recommendation system based on user's sentiments extracted from social networks," *IEEE Transactions on Consumer Electronics*, vol. 61, no. 3, pp. 359–367, 2015.

[7] P. G. Altbach, L. Reisberg, and L. E. Rumbley, *Trends in global higher education: Tracking an academic revolution*. Brill, 2019.

[8] H. Lu, S. Xu, X. Hu, E. Ngai, Y. Guo, W. Wang, and B. Hu, "Postgraduate student depression assessment by multimedia gait analysis," *IEEE MultiMedia*, vol. 29, no. 2, pp. 56–65, 2022.

[9] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proceedings of the 10th international conference on World Wide Web*, 2001, pp. 285–295.

[10] H. Zhang, F. Shen, W. Liu, X. He, H. Luan, and T.-S. Chua, "Discrete collaborative filtering," in *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, 2016, pp. 325–334.

[11] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *Proceedings of the 26th international conference on world wide web*, 2017, pp. 173–182.

[12] X. Wang, X. He, M. Wang, F. Feng, and T.-S. Chua, "Neural graph collaborative filtering," in *Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval*, 2019, pp. 165–174.

[13] Y. Qu, H. Cai, K. Ren, W. Zhang, Y. Yu, Y. Wen, and J. Wang, "Product-based neural networks for user response prediction," in *2016 IEEE 16th international conference on data mining (ICDM)*. IEEE, 2016, pp. 1149–1154.

[14] F. Yu, V. Koltun, and T. Funkhouser, "Dilated residual networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 472–480.

[15] M. Chen, B. Chang, C. Xu, and E. H. Chi, "User response models to improve a reinforce recommender system," in *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, 2021, pp. 121–129.

[16] C. Pei, X. Yang, Q. Cui, X. Lin, F. Sun, P. Jiang, W. Ou, and Y. Zhang, "Value-aware recommendation based on reinforcement profit maximization," in *The World Wide Web Conference*, 2019, pp. 3123–3129.

[17] G. Zhou, N. Mou, Y. Fan, Q. Pi, W. Bian, C. Zhou, X. Zhu, and K. Gai, "Deep interest evolution network for click-through rate prediction," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, no. 01, 2019, pp. 5941–5948.

[18] G. Zhou, X. Zhu, C. Song, Y. Fan, H. Zhu, X. Ma, Y. Yan, J. Jin, H. Li, and K. Gai, "Deep interest network for click-through rate prediction," in *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 2018, pp. 1059–1068.

[19] Y. Feng, F. Lv, W. Shen, M. Wang, F. Sun, Y. Zhu, and K. Yang, "Deep session interest network for click-through rate prediction," in *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, 2019, pp. 2301–2307.

[20] J. Xu, Z. Chen, S. Yang, J. Li, H. Wang, and E. C. H. Ngai, "Mentor: Multi-level self-supervised learning for multimodal recommendation," *arXiv preprint arXiv:2402.19407*, 2024.

[21] S. Dai, N. Shao, H. Zhao, W. Yu, Z. Si, C. Xu, Z. Sun, X. Zhang, and J. Xu, "Uncovering chatgpt's capabilities in recommender systems," in *Proceedings of the 17th ACM Conference on Recommender Systems*, 2023, pp. 1126–1132.

[22] J. Zhang, R. Xie, Y. Hou, W. X. Zhao, L. Lin, and J.-R. Wen, "Recommendation as instruction following: A large language model empowered recommendation approach," *arXiv preprint arXiv:2305.07001*, 2023.

[23] Y. Koren, S. Rendle, and R. Bell, "Advances in collaborative filtering," *Recommender systems handbook*, pp. 91–142, 2021.

[24] G. B. Martins, J. P. Papa, and H. Adeli, "Deep learning techniques for recommender systems based on collaborative filtering," *Expert Systems*, vol. 37, no. 6, p. e12647, 2020.

[25] G. Jain, T. Mahara, and S. C. Sharma, "Effective time context based collaborative filtering recommender system inspired by gower's coefficient," *International Journal of System Assurance Engineering and Management*, vol. 14, no. 1, pp. 429–447, 2023.

[26] H. I. Abdalla, A. A. Amer, Y. A. Amer, L. Nguyen, and B. Al-Maqaleh, "Boosting the item-based collaborative filtering model with novel similarity measures," *International Journal of Computational Intelligence Systems*, vol. 16, no. 1, p. 123, 2023.

[27] L. V. Nguyen, Q.-T. Vo, and T.-H. Nguyen, "Adaptive knn-based extended collaborative filtering recommendation services," *Big Data and Cognitive Computing*, vol. 7, no. 2, p. 106, 2023.

[28] G. Behera and N. Nain, "Collaborative filtering with temporal features for movie recommendation system," *Procedia Computer Science*, vol. 218, pp. 1366–1373, 2023.

[29] B. Karahodža, D. Donko, and H. Šupić, "Using temporal user profiles in collaborative filtering recommender system," in *2023 XXIX International Conference on Information, Communication and Automation Technologies (ICAT)*. IEEE, 2023, pp. 1–6.

[30] I. Newton, *Philosophiae naturalis principia mathematica*. G. Brookman, 1833, vol. 1.

[31] J. Xu, J. Liu, Z. Ma, Y. Wang, W. Wang, and E. C.-H. Ngai, "Knn-based collaborative filtering for fine-grained intelligent grad-school recommendation system," in *Quality, Reliability, Security and Robustness in Heterogeneous Systems: 19th EAI International Conference, QShine 2023, Shenzhen, China, October 8-9, 2023, Proceedings*. Springer, 2023.

[32] P. Covington, J. Adams, and E. Sargin, "Deep neural networks for youtube recommendations," in *Proceedings of the 10th ACM conference on recommender systems*, 2016, pp. 191–198.

[33] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec, "Graph convolutional neural networks for web-scale recommender systems," in *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 2018, pp. 974–983.

[34] G. R. Lima, C. E. Mello, A. Lyra, and G. Zimbrao, "Applying landmarks to enhance memory-based collaborative filtering," *Information Sciences*, vol. 513, pp. 412–428, 2020.

[35] G. Albora, L. R. Mori, and A. Zaccaria, "Sapling similarity: A performing and interpretable memory-based tool for recommendation," *Knowledge-Based Systems*, vol. 275, p. 110659, 2023.

[36] Y. Tay, L. Anh Tuan, and S. C. Hui, "Latent relational metric learning via memory-based attention for collaborative ranking," in *Proceedings of the 2018 world wide web conference*, 2018, pp. 729–739.

[37] S. Kabbur, X. Ning, and G. Karypis, "Fism: factored item similarity models for top-n recommender systems," in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2013, pp. 659–667.

[38] Y. Koren, "Factorization meets the neighborhood: a multifaceted collaborative filtering model," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2008, pp. 426–434.

[39] J. Chen, H. Zhang, X. He, L. Nie, W. Liu, and T.-S. Chua, "Attentive collaborative filtering: Multimedia recommendation with item-and component-level attention," in *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*, 2017, pp. 335–344.

[40] X. He, Z. He, J. Song, Z. Liu, Y.-G. Jiang, and T.-S. Chua, "Nais: Neural attentive item similarity model for recommendation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 12, pp. 2354–2366, 2018.

[41] M. Gori, A. Pucci, V. Roma, and I. Siena, "Itemrank: A random-walk based scoring algorithm for recommender engines." in *IJCAI*, vol. 7, 2007, pp. 2766–2771.

[42] R. van den Berg, T. N. Kipf, and M. Welling, "Graph convolutional matrix completion," in *KDD Workshop on Deep Learning Day*, 2018.

[43] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang, "Lightgcn: Simplifying and powering graph convolution network for recommendation," in *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, 2020, pp. 639–648.

[44] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *International Conference on Learning Representations (ICLR)*, 2017.

[45] Y. Shan, T. R. Hoens, J. Jiao, H. Wang, D. Yu, and J. Mao, "Deep crossing: Web-scale modeling without manually crafted combinatorial features," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 2016, pp. 255–262.

[46] Z. Zhang, Y. Huang, D. Ou, S. Li, L. Li, Q. Liu, and X. Zeng, "Rethinking the role of pre-ranking in large-scale e-commerce searching system," *arXiv preprint arXiv:2305.13647*, 2023.

[47] L. E. Peterson, "K-nearest neighbor," *Scholarpedia*, vol. 4, no. 2, p. 1883, 2009.

[48] X. Han, Z. Wang, and H. J. Xu, "Time-weighted collaborative filtering algorithm based on improved mini batch k-means clustering," *Advances in science and technology*, vol. 105, pp. 309–317, 2021.

[49] D. Li, P. Cao, Y. Guo, and M. Lei, "Time weight update model based on the memory principle in collaborative filtering." *J. Comput.*, vol. 8, no. 11, pp. 2763–2767, 2013.

[50] G. Jain, T. Mahara, and S. Sharma, "Performance evaluation of time-based recommendation system in collaborative filtering technique," *Procedia Computer Science*, vol. 218, pp. 1834–1844, 2023.

[51] Y.-C. Chen, L. Hui, and T. Thaipisutikul, "A collaborative filtering recommendation system with dynamic time decay," *The Journal of Supercomputing*, vol. 77, pp. 244–262, 2021.

[52] D. Sánchez-Moreno, Y. Zheng, and M. N. Moreno-García, "Time-aware music recommender systems: Modeling the evolution of implicit user preferences and user listening habits in a collaborative filtering approach," *Applied Sciences*, vol. 10, no. 15, p. 5324, 2020.

[53] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "Bpr: Bayesian personalized ranking from implicit feedback," in *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, 2009, pp. 452–461.

[54] W.-C. Kang and J. McAuley, "Self-attentive sequential recommendation," in *2018 IEEE international conference on data mining (ICDM)*. IEEE, 2018, pp. 197–206.

[55] C. Wissler, "The spearman correlation formula," *Science*, vol. 22, no. 558, pp. 309–311, 1905.

[56] J. C. Pinheiro and D. M. Bates, "Approximations to the log-likelihood function in the nonlinear mixed-effects model," *Journal of computational and Graphical Statistics*, vol. 4, no. 1, pp. 12–35, 1995.

[57] N. C. Chung, B. Miasojedow, M. Startek, and A. Gambin, "Jaccard/tanimoto similarity test and estimation methods for biological presence-absence data," *BMC bioinformatics*, vol. 20, no. 15, pp. 1–11, 2019.