entropy

MDPI

Article

A Hybrid Recommender System Based on Autoencoder and Latent Feature Analysis

Shangzhi Guo [1], Xiaofeng Liao [1], Gang Li [1], Kaiyi Xian [1], Yuhang Li [2] and Cheng Liang [3,*]

[1] College of Computer Science, Chongqing University, Chongqing 400044, China; 20211401018g@cqu.edu.cn (S.G.); xfliao@cqu.edu.cn (X.L.); 20211401019g@cqu.edu.cn (G.L.); 20211401021g@cqu.edu.cn (K.X.)
[2] College of Computer and Information Science, Southwest University, Chongqing 400715, China; limouhang@email.swu.edu.cn
[3] Institute of Artificial Intelligence and Blockchain, Guangzhou University, Guangzhou 510006, China
[*] Correspondence: c_liang@e.gzhu.edu.cn

Abstract: A recommender system (RS) is highly efficient in extracting valuable information from a deluge of big data. The key issue of implementing an RS lies in uncovering users' latent preferences on different items. Latent Feature Analysis (LFA) and deep neural networks (DNNs) are two of the most popular and successful approaches to addressing this issue. However, both the LFA-based and the DNNs-based models have their own distinct advantages and disadvantages. Consequently, relying solely on either the LFA or DNN-based models cannot ensure optimal recommendation performance across diverse real-world application scenarios. To address this issue, this paper proposes a novel hybrid recommendation model that combines Autoencoder and LFA techniques, termed AutoLFA. The main idea of AutoLFA is two-fold: (1) It leverages an Autoencoder and an LFA model separately to construct two distinct recommendation models, each residing in a unique metric representation space with its own set of strengths; and (2) it integrates the Autoencoder and LFA model using a customized self-adaptive weighting strategy, thereby capitalizing on the merits of both approaches. To evaluate the proposed AutoLFA model, extensive experiments on five real recommendation datasets are conducted. The results demonstrate that AutoLFA achieves significantly better recommendation performance than the seven related state-of-the-art models.

Keywords: data science; deep neural network; Latent Feature Analysis; multi-metric recommender system; matrix representation

check for updates

Citation: Guo, S.; Liao, X.; Li, G.; Xian, K.; Li, Y.; Liang, C. A Hybrid Recommender System Based on Autoencoder and Latent Feature Analysis. *Entropy* **2023**, *25*, 1062. https://doi.org/10.3390/e25071062

Academic Editor: Yanchun Liang

Received: 30 June 2023
Revised: 9 July 2023
Accepted: 10 July 2023
Published: 14 July 2023

1. Introduction

In the current era characterized by abundant information, individuals are confronted with a deluge of extensive data [1–4]. Notable examples include the colossal amount of data generated by Google, reaching the scale of petabytes, and Flickr, which produces terabytes of data on a daily basis [5,6]. The challenge at hand is to devise an intelligent system capable of extracting relevant information from these vast datasets [7–9]. One practical approach to tackle this challenge is the utilization of a recommender system (RS). RSs play crucial roles in enhancing online services, contributing to both business growth and improved user experiences [10]. Typically, a user-item rating matrix is employed to capture user preferences across various items such as news, short videos, music, movies, and commodities [11]. In this matrix, each row represents a specific user, each column corresponds to a specific item, and each entry signifies a user's preference for a particular item [3]. The key to implementing an RS lies in uncovering users' latent preferences for different items based on this user-item rating matrix [12,13].

Numerous approaches have been proposed for implementing an RS. Among them, the Latent Feature Analysis (LFA) model has gained significant popularity in industrial applications due to its efficiency and scalability [14]. When applied to a user-item rating matrix, the LFA model projects users and items onto a shared low-dimensional Latent Feature space [15]. By training two low-dimensional matrices using the observed entries only [16], the LFA model can estimate the missing entries by leveraging these trained matrices [17–20]. As a result, the LFA model offers advantages in terms of efficiency and scalability, particularly in industrial contexts. However, it should be noted that the LFA model is a linear model and may not effectively address complex non-linear relationships between users and items [21].

In recent times, the rapid advancement of deep learning has led to the widespread adoption of deep neural networks (DNNs) [22–24] in RSs [25,26]. DNNs have emerged as a promising approach for capturing complex non-linear relationships between users and items [27,28]. In the pursuit of implementing RSs, various DNN-based models have been proposed, with significant emphasis placed on devising sophisticated structures that can better accommodate user behavior data [29]. However, a notable difference between DNN-based models and the Latent Feature Analysis (LFA) model lies in their approaches to handling data [30–34]. While DNN-based models often operate on complete data, the observed entries within a user-item rating matrix, the reality is that RS-generated user-item rating matrices tend to exhibit low rating density [35–38]. This means that a significant portion of the matrix remains empty or contains missing ratings. Consequently, DNN-based models face challenges in effectively addressing the prevalent data sparsity issues in RSs [12,13,39,40].

Upon the aforementioned discussions, it becomes apparent that the LFA and DNN-based models have distinct advantages and disadvantages. Consequently, relying solely on either the LFA model or the DNN-based model cannot ensure optimal recommendation performance across diverse real-world application scenarios. To tackle this challenge, this study proposes a novel hybrid recommendation model called AutoLFA, which combines Autoencoder [41] and LFA techniques. The main concept behind AutoLFA is two-fold: (1) It leverages an Autoencoder and an LFA model separately to construct two distinct recommendation models, each residing in a unique metric representation space with its own set of strengths, and (2) it integrates the Autoencoder and LFA models using a customized self-adaptive weighting strategy, thereby capitalizing on the merits of both approaches. By incorporating elements from both the LFA model and DNN-based models, AutoLFA can deliver superior recommendation performance across various real-world application scenarios. This paper contributes to the field in the following ways:

1. It proposes an AutoLFA model that aggregates the merits of both the LFA model and the DNN-based model by a customized self-adaptive weighting strategy;
2. Theoretical analyses and model designs are provided for the proposed AutoLFA model;
3. Extensive experiments on five real recommendation datasets are conducted to evaluate the proposed AutoLFA model. The results demonstrate that AutoLFA achieves significantly better recommendation performance than the related state-of-the-art models.

2. Related Work

Collaborative Filtering (CF) stands as a popular and effective approach for implementing an RS [2]. Its fundamental principle involves utilizing historical user behavior data to uncover similarities between users and items, thereby predicting users' potential preferences for items. Matrix factorization serves as a prominent CF method, which typically maps the user-item rating matrix into two Latent matrices to explore the similarity between users and items [12]. Subsequently, the development of the LFA model introduced a notable distinction. Unlike matrix factorization, the LFA model exclusively trains the Latent Feature model using observed entries within the user-item rating matrix. As a result, LFA exhibits high efficiency and scalability, particularly in industrial applications [12,13]. Over time, several sophisticated LFA models have emerged, including those that consider data characteristics [42], incorporate non-negativity constraints [43], adopt generalized and fast-converging approaches [44], focus on smooth L_1-norm regularization [12], employ probabilistic methods [45], apply dual loss [13], utilize prediction sampling [46,47], prioritize confidence-driven techniques [48], incorporate posterior neighborhood regularization [49], employ ensemble approaches involving multiple spaces and norms [50], explore graph regularization [51], and embrace deep structured architectures [52]. However, it is essential to note that the LFA model is inherently shallow and linear in nature. Consequently, it faces challenges when attempting to capture the deep non-linear relationships between users and items embedded within complex user-item rating matrices [21].

In recent times, Deep Neural Networks (DNN) have gained significant traction in the development of Collaborative Filtering (CF)-based RSs due to their powerful non-linear learning capabilities derived from deep learning structures [53]. DNN-based models aim to reduce the user-item rating matrix into a low-dimensional space to capture the similarities between users and items. A comprehensive review of DNN-based RSs was conducted by Zhang et al. [29]. Various sophisticated DNN-based models have emerged, including hybrid Autoencoder-based approaches [54], Autoencoder-based methods [41], multi-task learning-oriented techniques [11], graph neural network (GNN)-based models [55], neural factorization-based approaches [56], Autoencoders combined with radial basis function-based methods [57], attentional factorization-based models [58], hybrid deep models [28], biased Autoencoder-based techniques [21], and convolutional matrix factorization approaches [59]. However, it is worth noting that DNN-based models face challenges in addressing data sparsity problems since they are trained on complete data rather than solely relying on the observed entries within a user-item rating matrix [13]. Unfortunately, user-item rating matrices generated by RSs often exhibit very low rating densities.

Notably, although many LFA-based and DNN-based models have been built to achieve commendable recommendation performance, each approach has its own set of advantages and disadvantages. In comparison, the proposed AutoLFA is a hybrid recommendation model that combines the strengths of both Autoencoder and LFA techniques. This combination is controlled by a customized self-adaptive weighting strategy, ensuring that AutoLFA leverages the merits of both the LFA and DNN-based models, ultimately leading to superior recommendation performance across various real-world application scenarios.

3. Preliminaries

Definition 1 (**user behavior data**)**:** *Let M be a set of users, and N be a set of items. The matrix $X \in \mathbb{R}$ with $|M|$ rows and $|N|$ columns records the interactions between different users and items. Here, x_{mn} represents the specific interaction specification of user m on item n. The vector $\mathrm{x}^{m} = \{x_{m1}, \cdots x_{m|N|}\}$ denotes the behavioral data of user m across all items, while each item n can be represented as a vector $\mathrm{x}^{n} = \{x_{1n}, \cdots x_{|M|n}\}$. A binary matrix $B \in \mathbb{R}$ with $|M|$ rows and $|N|$ columns distinguish the observed and unobserved interactions of X:*

$$b_{mn} = \begin{cases} 1 & \text{if } x_{mn} \text{ observed} \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where b_{mn} denotes the specific entry on B.

Definition 2 (**problem**)**:** *In recommender systems, two primary tasks exist: rating prediction and ranking prediction. Our proposed model is more suited to rating prediction, which aims to learn a parametric model denoted as $f(\cdot)$ using observed ratings of X in order to predict the unobserved ones. The prediction process can be represented as follows:*

$$f(M, N; \theta) \rightarrow X. \tag{2}$$

Here, θ represents the parameters of f(·). The objective function of f(·) is to minimize the empirical risk, expressed as:

$$L(f) = \sum_{m \in M, n \in M} \epsilon(f(m,n;\theta), x_{mn}). \tag{3}$$

In this Equation, ϵ(·) denotes the error function that measures the distance between the predicted output $\hat{x}_{mn}$ from f(m, n; θ) and the true rating x_{mn}.

4. The Proposed AutoLFA

As mentioned above, traditional approaches, such as Latent Feature Analysis (LFA), offer efficiency and scalability but may not capture complex non-linear relationships. On the other hand, deep neural networks (DNNs) show potential in capturing non-linear relationships but face challenges in dealing with data sparsity issues. Inspired by this finding, we propose AutoLFA with the aim of addressing both the challenge of LFA's inability to capture complex non-linear relationships and the difficulty faced by DNN-based models in handling data sparsity issues. Figure 1 depicts the architecture of our proposed model, which can be separated into three steps: (1) Feed the user behavior data into the LFA-based and Autoencoder-based models separately; (2) obtain the predictions of the unobserved value from these two models; (3) aggregate the predictions of two models with a self-adaptive ensemble method to obtain final prediction $\hat{X}$. To illustrate the principle of Auto-LFA, we provide an example of predicting x_{22} in Figure 1. The predicted values from the two predictors differ by 3.5 in the LFA-based model and 2 in the Autoencoder-based model. These predictions are then weighted to derive the final prediction of 2.9. Next, we will provide a detailed description of AutoLFA.

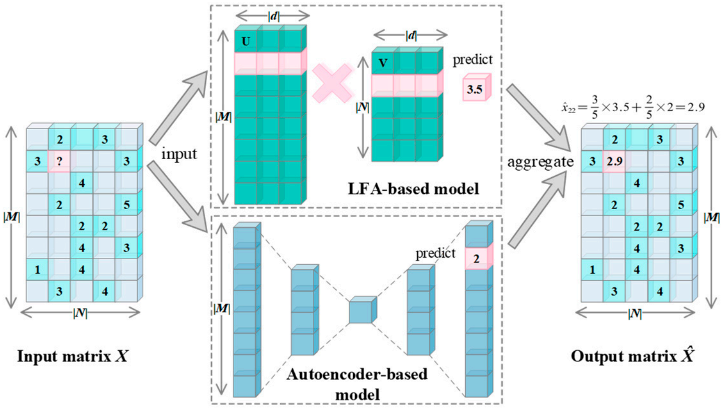


Figure 1. The architecture of the proposed AutoLFA model.

4.1. The Latent Feature Analysis-Based (LFA-Based) Model

Given a user behavior matrix X, an LFA-based predictor aims to train two Latent Feature matrices U of size $|M| \times d$ and V of size $d \times |N|$ to generate the rank-d approximation $\hat{X}$ of X is based on the known entry of X, in which d is much smaller than $min\{|M|, |N|\}$. In this context, the row vectors of U represent user characteristics, while the column vectors of V represent item characteristics in the Latent Feature space.

We utilize the inner product space with an L_2-norm $||\cdot||_{L_2}^2$ as the *Loss* function in the LFA-based model to measure the distance between X and $\hat{X}$, as demonstrated below:

$$L(U,V) = \frac{1}{2} \parallel B \odot (X - \hat{X})_{L_2}^2 = \frac{1}{2} \parallel B \odot (M - UV)_{L_2}^2 \tag{4}$$

where $\odot$ denotes the Hadamard product. According to [12,13], regularization is crucial in preventing overfitting. By incorporating Tikhonov regularization into Equation (4), we obtain:

$$L(U,V) = \frac{1}{2} \parallel B \odot (X - UV) \parallel_{L_2}^2 + \frac{\lambda_1}{2}(\|U\|_{L_2}^2 + \|V\|_{L_2}^2). \tag{5}$$

Here, λ_1 is a hyperparameter that controls the intensity of its regularization penalty. It is worth noting that since the user cannot fully access all items leading X to be sparse, it is necessary to expand Equation (5) into a density-oriented form to improve efficiency, as follows [12,13]:

$$L(U,V) = \frac{1}{2} \sum_{x_{mn} \in X_o} \left(x_{mn} - \sum_{k=1}^{d} u_{mk} v_{kn}\right)^2 + \frac{\lambda}{2} \sum_{x_{mn} \in X_o} \left(\sum_{k=1}^{d} u_{mk}^2 + \sum_{k=1}^{d} v_{kn}^2\right). \tag{6}$$

Here, u_{mk} represents the entry at the u-th row and k-th column of U, and v_{kn} represents the entry at the k-th row, n-th column of V, and X_o is the observed entries of X. We train the matrices U and V with the Adam optimizer [16] to obtain better prediction results.

4.2. The Autoencoder-Based Model

We chose the representative I-AutoRec [41] as the Autoencoder-based model. Formally, when given a user behavior data matrix X, I-AutoRec aims to solve the same problem as defined in Equation (3). The objective is to minimize the following loss function:

$$L(f) = \sum_{\mathbf{x}^n \in M} \|(\mathbf{x}^n - f(\mathbf{x}^n;\theta)) \odot \mathbf{b}^n\|_{L_2}^2 + \frac{\lambda_2}{2} \cdot (\|w_1\|_{L_2}^2 + \cdots + \|w_K\|_{L_2}^2), \tag{7}$$

where $\lambda_2 > 0$ represents the regularization factor to prevent I-AutoRec from overfitting. The parameter set $\theta = \{w_1, \ldots, w_k, b_1, \ldots, b_k\}$ includes the weighted terms w_k and the intercept terms b_k of the hidden layers, where $k \in \{1, 2, \ldots, K\}$, $\mathbf{b}^n$ represents the n-th column of the index matrix B, and $\mathbf{x}^n$ corresponds to the item vector $\mathbf{x}^n = \{x_{1n}, \ldots, x_{|M|n}\}$.

4.3. Self-Adaptive Aggregation

Ensemble learning is a practical approach to combining multiple models. It is essential for the base models to exhibit diversity and accuracy [13]. To ensure diversity, we employ different types of models. Additionally, the representative LFA-based model and Autoencoder-based I-AutoRec ensure accuracy. As a result, the base models fulfil the two requirements for ensemble learning. To aggregate the models, we adopt a self-adaptive aggregation method based on their loss values on the validation set. The underlying principle is to increase the weight of the t-th base model if its loss decreases in the i-th training iteration or otherwise decreases. To comprehensively understand this idea, we will introduce relevant definitions to facilitate theoretical analysis.

Definition 3 (Fractional *Loss* of Base Models): *The fractional loss of the t-th base model at the i-th iteration, denoted as $Fl^t(i)$, is computed as follows:*

$$\begin{gathered} Fl^t(i) = \sqrt{\sum_{m \in M, n \in N, (M,N \in \Gamma)} ((x_{mn} - \hat{x}_{mn}^t) \times m_{mn})^2 / \|T\|_0} \\ \hat{x}_{mn}^t = \begin{cases} \sum_{k=1}^{d} u_{mk} v_{kn} & \text{if } t = 1 \\ f(m,n;\theta) & \text{if } t = 2 \end{cases} \end{gathered}, \tag{8}$$

where $||\cdot||_0$ represents the L_0-norm of a matrix which calculates the number of non-zero elements of it, and Γ is the validation subset of X.

Definition 4 (Cumulative *Loss* of Base Models): *We let $Cl^t(i)$ be the cumulative loss of Fl^t until the i-th training iteration and calculate as follows:*

$$Cl^t(i) = \sum_{j=1}^{i} Sl^t(j). \tag{9}$$

Definition 5 (Ensemble Weights): *The ensemble weight Ew^t for the t-th base model can be computed using the following formula:*

$$Ew^t(i) = \frac{e^{-\delta Cl^t(i)}}{\sum\limits_{l=1}^{2} e^{-\delta Cl^t(i)}}. \tag{10}$$

Here, δ represents the equilibrium factor that controls the ensemble weights of the aggregation during the training process. Considering Definitions 3 to 5, the final prediction of AutoLFA in the i-th training iteration can be denoted as:

$$\hat{x}_{mn} = \sum_{t=1}^{2} El^t(i) \cdot \hat{x}^t_{mn}. \tag{11}$$

4.4. Theoretical Analysis

The loss of the AutoLFA model at the i-th training iteration is represented as $Fl(i)$ and computed as follows:

$$Fl(n) = \sqrt{\sum_{m\in M, n\in N, (M,N\in\Gamma)} ((x_{mn} - \hat{x}_{mn}) \times b_{mn})^2 / \parallel \Gamma \parallel_0}, \tag{12}$$

where $\hat{x}_{mn}$ is calculated using align (11).

Definition 6 (Cumulative *Loss* of AutoLFA): *The cumulative loss of the AutoLFA model is represented as Cl(i) and can be expressed as:*

$$Cl(i) = \sum_{j=1}^{i} Fl(j). \tag{13}$$

Theorem 1. *For an AutoLFA model, assuming the $Cl^t(i)$ of the base models lies between [0, 1], and if $Ew^t(i)$ is set according to align (10) during training, the following alignment holds:*

$$Cl(I) \le \min\{Cl^t(I) \mid t = 1, 2\} + \frac{\ln 4}{\delta} + \frac{\delta I}{8}, \tag{14}$$

where I is the maximum iteration.

By setting $\delta = \sqrt{1/\ln I}$ in Theorem 1, the upper bound becomes:

$$Cl(I) \le \min\{Cl^t(I)|t = 1, 2\} + \ln 2\sqrt{\ln I} + \frac{I}{8\sqrt{\ln I}}, \tag{15}$$

where $\ln 2\sqrt{\ln I} + \frac{I}{8\sqrt{\ln I}}$ is bound by I linearly. This leads us to the following proposition.

Proposition 1. *With $\delta = \sqrt{1/\ln I}$, the inequality holds:*

$$Cl(I) \leq \min\{Cl^t(I) | t = 1, 2\} + const, \tag{16}$$

where the limit as I approaches infinity, const = 19.45.

Remark 1. *Proposition 1 indicates that Cl(I) is constrained by min{$Cl^t(I)$ | t = 1, 2} + const, with $\delta = \sqrt{1/\ln I}$. Remarkably, each base variant with a different foundation allows them to exist in separate metric spaces. The ensemble weight in align (10) ensures that the AutoLFA model's loss is always lower than the base models and benefits from the capabilities derived from the LFA and DNN-based models. Additionally, Proposition 1 is not intended to demonstrate the accuracy improvement of AutoLFA on the test set but rather to establish that the model possesses the advantages of the basic models. By showing that the proposed model achieves a smaller loss compared to each basic model used separately, it indicates that the model retains the respective strengths of the basic models without compromising its ability to fit the data.*

5. Experiments

In this section, we aim to address the following research questions (RQs) through subsequent experiments:

- RQ 1: Does the proposed AutoLFA model outperform state-of-the-art models in accurately predicting user behavior data?
- RQ 2: How does the AutoLFA model self-adaptively control the ensemble weights of its base models during the training process to ensure optimal performance?
- RQ 3: Are the base models of AutoLFA diversified in their ability to represent the same user behavior data matrix, thereby enhancing the performance of AutoLFA?
- RQ 4: What is the impact of the number of Latent Features and hidden units in the base models on the accuracy of AutoLFA?

5.1. General Settings

Datasets: For our experiments, we utilize five commonly used user-item datasets, as summarized in Table 1 These datasets include MovieLens_1M, MovieLens_100k, and MovieLens_HetRec from the MovieLens website, the Yahoo dataset from the Yahoo website, and the Douban dataset obtained from an open-access code. Table 1 summarizes the details of these datasets. The datasets are divided into train–validate–test sets using a ratio of 70%–10%–20%.

Table 1. Properties of all the datasets.

No.	Name	∣M∣	∣N∣	∣HO∣	Density *
D1	MovieLens_1M	6040	3952	1,000,209	4.19%
D2	MovieLens_100k	943	1682	100,000	6.30%
D3	MovieLens_HetRec	2113	10,109	855,598	4.01%
D4	Yahoo	15,400	1000	365,704	2.37%
D5	Douban	3000	3000	136,891	1.52%

* Density denotes the percentage of observed entries in the user-item matrix.

Evaluation Metrics: The primary objective of representing the user-item matrix is to predict missing ratings accurately. To assess the prediction accuracy of the tested models, we employ two evaluation metrics: root mean square error (RMSE) and mean absolute error (MAE), which are calculated according to [52].

Baselines: Our proposed MMA model is compared against seven state-of-the-art models: AutoRec (an original model), MF, and FML (Latent Feature Analysis-based models), and NRR, SparseFC, IGMC, and GLocal-K (deep-learning models). A brief description of these competing models is provided in Table 2.

Table 2. Descriptions of all the contrasting models.

Model	Description
MF [10]	A representative LFA-based model for factorizing user-item matrix data in recommender systems. *Computer 2009.*
AutoRec [41]	A notable DNNs-based model for representing user-item data in recommender systems. *WWW 2015.*
NRR [11]	A DNNs-based multi-task learning framework for rating prediction in recommender systems. *SIGIR 2017.*
SparseFC [27]	A DNNs-based model that reparametrizes weight matrices into low-dimensional vectors to capture important features. *ICML 2018.*
IGMC [55]	A GNNs-based model for inductive matrix completion without using side information. *ICLR 2019.*
FML [9]	An LFA-based model that combines metric learning (distance space) and collaborative filtering. *IEEE TII 2020.*
GLocal-K [57]	A DNNs-based model for generalizing and representing user-item data in a low-dimensional space with important features. *CIKM 2021.*

Implementation Details: For all datasets, we set the learning rate to 0.001 for two models. We set the number of hidden units for the Autoencoder to 500 and the number of latent factors for the LFA model 30 to achieve better performance. The final testing results are obtained from the best-performing model, which exhibits the lowest prediction error on the validation set during training. The training process terminates when the preset threshold for training iterations is reached. All experiments are conducted on a GPU server with two 2.4 GHz Xeon Gold 6240 R CPUs, 376.40 GB RAM, and 4 Tesla V100 GPUs.

5.2. Performance Comparison (RQ. 1)

5.2.1. Comparison of Prediction Accuracy

Table 3 presents the prediction accuracies of all models from D1 to D5. Statistical tests, including loss/tie/win analysis, the Wilcoxon signed-ranks test [60], and the Friedman test [21], are performed to analyze these results. The loss/tie/win analysis identifies cases where AutoLFA's RMSE/MAE is higher/same/lower than other competitors. The Wilcoxon signed-ranks test is a non-parametric pairwise comparison method that determines if AutoLFA's prediction accuracy is significantly higher than each comparison model based on p-values. The Friedman test compares the performance of multiple models across multiple datasets using F-rank values, with lower values indicating higher prediction accuracy. The comparative experiment results are normalized for better interpretation before conducting the Wilcoxon signed-ranks test and the Friedman test. The statistical analysis results of loss/tie/win, the Wilcoxon signed-ranks test, and the Friedman test are presented in the third-to-last, second-to-last, and last rows of Table 3. Key observations from Table 3 are as follows:

- AutoLFA achieves the lowest RMSE/MAE in most cases, with only ten cases of loss and one case of a tie in comparison. The total count of loss/tie/win cases is 7/1/62.
- All p-values are below the significance level of 0.1, indicating that AutoLFA outperforms all competitors in terms of prediction accuracy.
- AutoLFA obtains the lowest F-rank among all participants, confirming its highest accuracy across all datasets.

These observations highlight that AutoLFA achieves the highest prediction accuracy for predicting missing user data compared to other models.

Table 3. Performance comparison of AutoLFA and its competitors.

Dataset	Metric	MF	AutoRec	NRR	SparseFC	IGMC	FML	Glocal-K	AutoLFA
D1	RMSE	0.857 ●	0.847 ●	0.881 ●	0.839 ○	0.867 ●	0.849 ●	0.839 ○	0.842
	MAE	0.673 ●	0.667 ●	0.691 ●	0.656 ○	0.681 ●	0.667 ●	0.655 ○	0.664
D2	RMSE	0.913 ●	0.897 ●	0.923 ●	0.899 ●	0.915 ●	0.904 ●	0.892 ●	0.887
	MAE	0.719 ●	0.706 ●	0.725 ●	0.706 ●	0.722 ●	0.718 ●	0.697	0.699
D3	RMSE	0.757 ●	0.752 ●	0.774 ●	0.749 ●	0.769 ●	0.754 ●	0.756 ●	0.744
	MAE	0.572 ●	0.569 ●	0.583 ●	0.567 ●	0.582 ●	0.573 ●	0.573 ●	0.562
D4	RMSE	1.206 ●	1.172 ●	1.227 ●	1.203 ●	1.133 ○	1.176 ●	1.204 ●	1.167
	MAE	0.937 ●	0.900 ●	0.949 ●	0.915 ●	0.848 ○	0.937 ●	0.905 ●	0.895
D5	RMSE	0.738 ●	0.744 ●	0.726 ●	0.745 ●	0.751 ●	0.762 ●	0.737	0.737
	MAE	0.588 ●	0.588 ●	0.573 ●	0.587 ●	0.594 ●	0.598 ●	0.580 ○	0.584
Statistic	loss/tie/win	0/0/10	0/0/10	0/0/10	2/0/8	2/0/8	0/0/10	3/1/6	**7/1/62** *
	p-value	0.0039	0.0039	0.0039	0.039	0.0195	0.039	0.0977	-
	F-rank	5.7	3.75	6.6	3.5	5.9	5.45	3.05	**2.05**

* The total loss/tie/win cases of AutoLFA. ● The cases in which AutoLFA wins the other models in comparison. ○ The cases in which AutoLFA loses the comparison.

5.2.2. Comparison of Computational Efficiency

Figure 2 depicts the total time required for all participating models to reach the optimal RMSE on the validation dataset during training. The following observations can be made:

- LFA-based models generally exhibit higher computational efficiency compared to DNN-based models, as they are trained on observed user behavior data, unlike DNN-based models.
- Due to their complex data form and architecture, GNN-based models consume significant computational resources and time. From Figure 2, it is evident that IGMC surpasses 3000 s in time costs.
- Except for slightly longer time consumption on dataset D4, AutoLFA's time consumption falls between LFA-based and GNN-based models. It is slightly higher than the original Autoencoder-based model but faster than other DNN-based models in most cases.

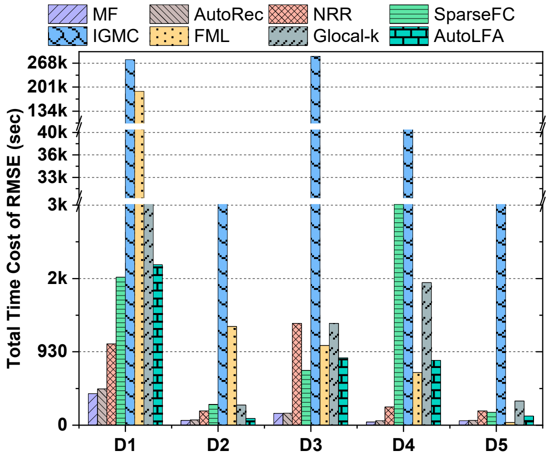


Figure 2. The histogram graph of the total time cost to reach the optimal accuracy of all the participating models.

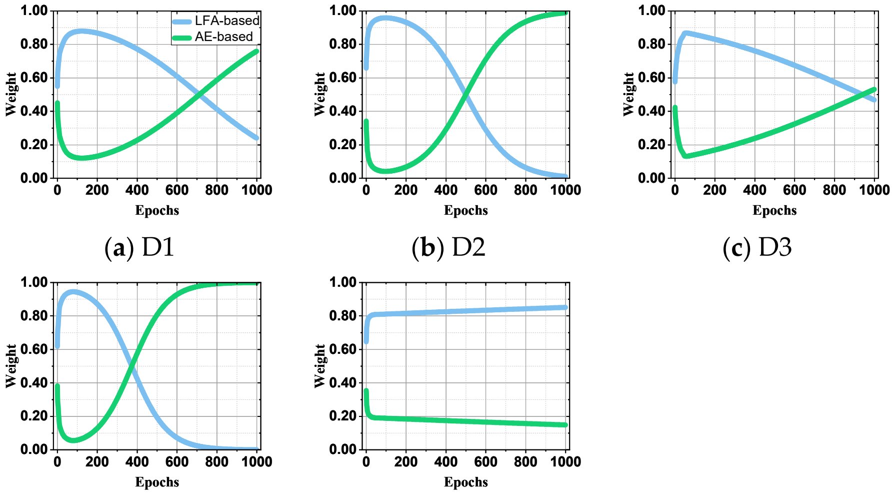
Based on these observations, we can conclude that the relatively simple structure of the base models in AutoLFA allows for acceptable total time costs after ensembling two base models.

5.3. The Self-Ensembling of MMA (RQ. 2)

To discuss the self-adaptive control of AutoLFA in ensembling different variant models and ensuring its performance, we monitor the variations of ensemble weights between its base models.

Monitoring ensemble weight variations: Figure 3 illustrates the changes in ensemble weights from D1 to D5, yielding the following observations:

- In most cases (e.g., Figure 3a–d), the ensemble weights of the Autoencoder-based model gradually increase and surpass the LFA-based model as the training progresses until the base models are fitted.
- In some instances, the LFA-based model's weight may exceed that of the Autoencoder-based model. For example, in Figure 3e, the ensemble weight of the LFA-based model is greater than those of the Autoencoder-based model due to their faster convergence.



(**a**) D1 (**b**) D2 (**c**) D3

(**d**) D4 (**e**) D5

Figure 3. The changes in ensemble weights during the training process.

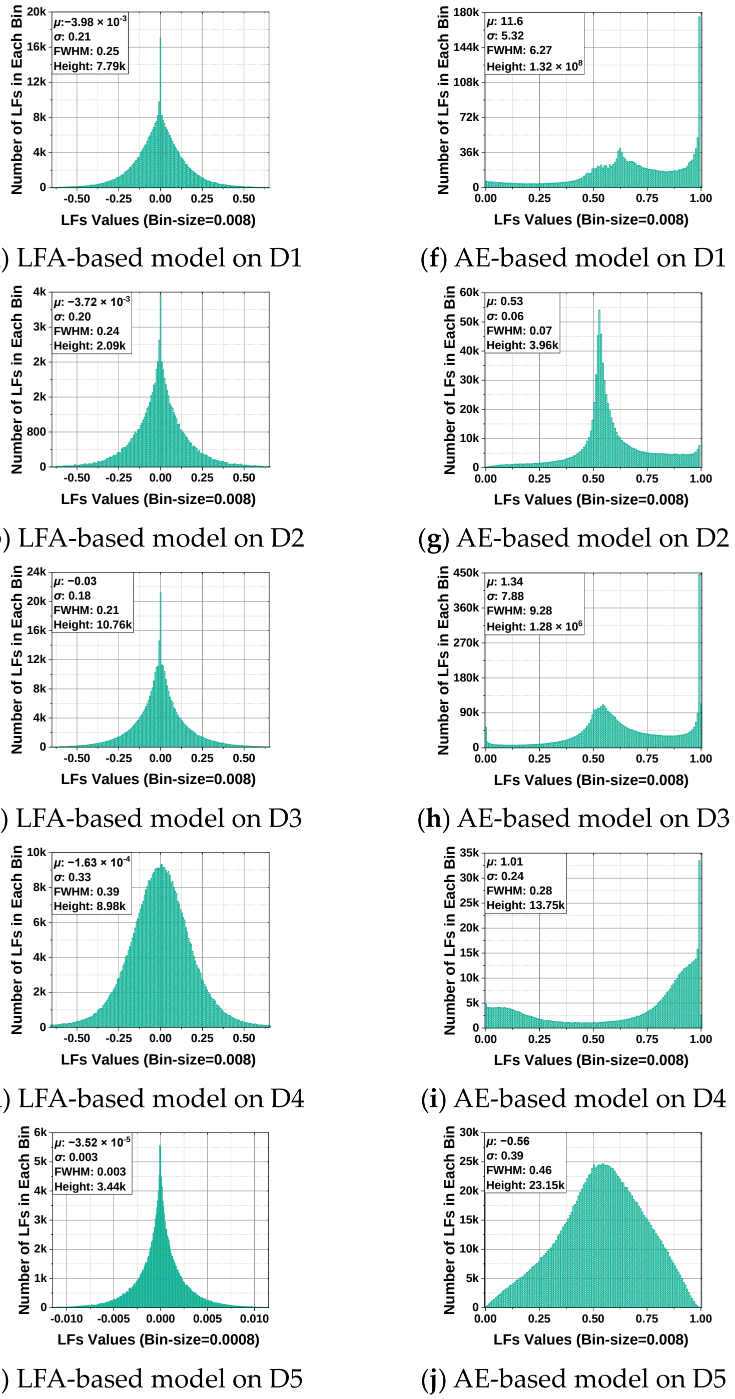
In conclusion, based on the experimental results and observations above, we can infer that AutoLFA effectively leverages different types of models. By aggregating these models in the ensemble stage, AutoLFA surpasses other state-of-the-art models in predicting missing ratings with only minor sacrifices in computational resources.

5.4. Distribution of Latent Features of Base Models (RQ. 3)

In order to investigate the diversity of the base models of AutoLFA and their abilities to predict the user behavior data matrix, we visually analyze the encoder output of the Autoencoder-based model, which represents the Latent Features of an Autoencoder model, and the Latent Features of the LFA-based model. The distribution of these Latent Features for the base models across all datasets is depicted in Figure 4. To analyze the distribution, we employ a Gaussian function and examine factors such as expectation (μ) and standard deviation (σ). The measurements of the full width at half maximum (FWHM) and the height of the Gaussian curve are also presented. From Figure 4, the following observations emerge:

- The distribution of Latent Features in the Autoencoder-based model tends to have more values concentrated at the extremes (i.e., 0 or 1), as shown in Figure 4f,h,i, while in the LFA-based model, the distribution tends to follow a normal distribution.
- After encoding, the Autoencoder-based model's Latent Features are more likely to exhibit unusually high values within specific ranges. In contrast, in the LFA-based model, there are no extreme values, as depicted in Figure 4a,c,d.

- In some cases, the distribution of Latent Features in the Autoencoder-based model appears to be slightly more uniform compared to the LFA-based model, as illustrated in Figure 4e,j.



(**a**) LFA-based model on D1 (**f**) AE-based model on D1

(**b**) LFA-based model on D2 (**g**) AE-based model on D2

(**c**) LFA-based model on D3 (**h**) AE-based model on D3

(**d**) LFA-based model on D4 (**i**) AE-based model on D4

(**e**) LFA-based model on D5 (**j**) AE-based model on D5

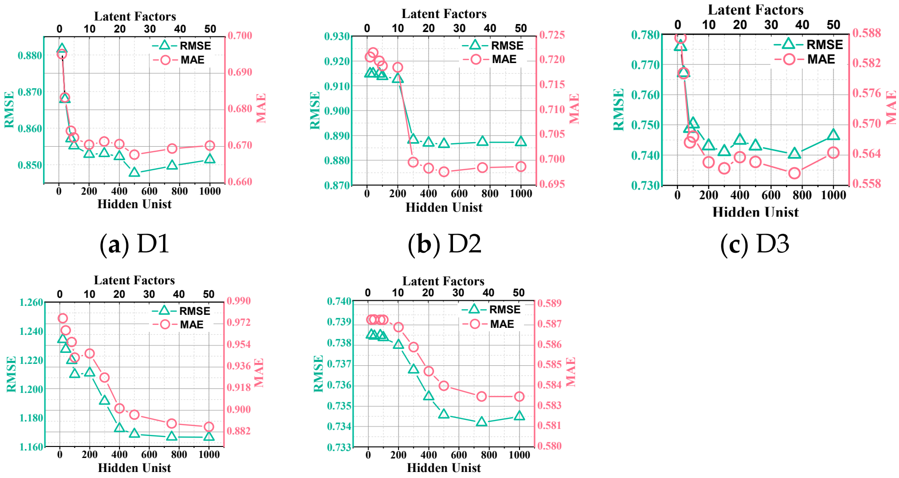
Figure 4. The distribution histogram of LFs of LFA-based and Autoencoder-based models from D1 to D5.

The observed information above indicates that the Autoencoder-based and LFA-based models have distinct representation characteristics, allowing AutoLFA to benefit from their different representation abilities. Consequently, AutoLFA ensures accurate prediction of missing ratings.

5.5. Influence of Numbers of Latent Features and Hidden Units to Base Models (RQ. 4)

We further investigate the impact of the number of Latent Features and hidden units in the base models on AutoLFA. Figure 5 illustrates the RMSE and MAE of AutoLFA as the number of Latent Features and hidden units varies simultaneously across D1 to D5. The following observations can be made from Figure 5:

- Increasing the number of Latent Features/hidden units from 2/20 to 20/300 results in a rapid improvement in the accuracy of AutoLFA. During this range, AutoLFA substantially increases accuracy without incurring significant computational costs.
- Once the number of Latent Features/hidden units reaches 25/400, the rate of accuracy improvement becomes less prominent in Figure 5b–d.



(**a**) D1 (**b**) D2 (**c**) D3

(**d**) D4 (**e**) D5

Figure 5. The line graphs of RMSE and MAE of AutoLFA from D1 to D5 as the number of Hidden Units and Latent Factors vary.

These observations suggest that setting the number of Latent Features/hidden units as 30/500 allows AutoLFA to achieve optimal accuracy in most cases without imposing significant computational resource demands. Although this setting may not yield the highest accuracy in certain cases, it remains relatively close to the optimal value.

6. Conclusions

This paper proposes a novel hybrid recommendation model by combining Autoencoder and LFA models, termed AutoLFA. Its main idea is two-fold: (1) It leverages an Autoencoder and a Latent Feature Analysis (LFA) model separately to construct two distinct recommendation models, each residing in a unique metric representation space with its own set of strengths, and (2) it integrates the Autoencoder and LFA models using a customized self-adaptive weighting strategy. As such, the merits of the LFA and DNN-based models are combined into the AutoLFA model, making it achieve superior recommendation performance under various real-world applications. The experiments investigate four research questions on five real recommendation datasets. The results verify that the proposed AutoLFA outperforms several state-of-the-art models. In the future, we plan to aggregate more variants of LFA-based and deep neural networks (DNNs)-based models to achieve better recommendation performance.

Author Contributions: Formal analysis, X.L.; investigation, Y.L.; methodology, S.G.; software, K.X.; validation, G.L.; writing—original draft, S.G.; writing—review and editing, C.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: All data supporting the reported results in our study are publicly available and can be accessed through the following link: https://grouplens.org/datasets/movielens/, https://webscope.sandbox.yahoo.com/catalog.php?datatype=r, https://github.com/fmonti/mgcnn.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Deng, S.; Zhai, Y.; Wu, D.; Yue, D.; Fu, X.; He, Y. A Lightweight Dynamic Storage Algorithm With Adaptive Encoding for Energy Internet. *IEEE Trans. Serv. Comput.* **2023**, 1–14. [CrossRef]
2. He, Y.; Wu, B.; Wu, D.; Beyazit, E.; Chen, S.; Wu, X. Toward Mining Capricious Data Streams: A Generative Approach. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *32*, 1228–1240. [CrossRef] [PubMed]
3. Wang, D.; Liang, Y.; Xu, D.; Feng, X.; Guan, R. A content-based recommender system for computer science publications. *Knowl.-Based Syst.* **2018**, *157*, 1–9. [CrossRef]
4. Bai, X.; Huang, M.; Xu, M.; Liu, J. Reconfiguration Optimization of Relative Motion Between Elliptical Orbits Using Lyapunov-Floquet Transformation. *IEEE Trans. Aerosp. Electron. Syst.* **2022**, *59*, 923–936. [CrossRef]
5. You, D.; Niu, S.; Dong, S.; Yan, H.; Chen, Z.; Wu, D.; Shen, L.; Wu, X. Counterfactual explanation generation with minimal feature boundary. *Inf. Sci.* **2023**, *625*, 342–366. [CrossRef]
6. You, D.; Xiao, J.; Wang, Y.; Yan, H.; Wu, D.; Chen, Z.; Shen, L.; Wu, X. Online Learning From Incomplete and Imbalanced Data Streams. *IEEE Trans. Knowl. Data Eng.* **2023**, 1–14. [CrossRef]
7. Chen, J.; Wang, Q.; Peng, W.; Xu, H.; Li, X.; Xu, W. Disparity-based multiscale fusion network for transportation detection. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 18855–18863. [CrossRef]
8. Cao, B.; Zhao, J.; Lv, Z.; Yang, P. Diversified personalized recommendation optimization based on mobile data. *IEEE Trans. Intell. Transp. Syst.* **2020**, *22*, 2133–2139. [CrossRef]
9. Zhang, S.; Yao, L.; Wu, B.; Xu, X.; Zhang, X.; Zhu, L. Unraveling metric vector spaces with factorization for recommendation. *IEEE Trans. Ind. Inform.* **2019**, *16*, 732–742. [CrossRef]
10. Koren, Y.; Bell, R.; Volinsky, C. Matrix Factorization Techniques for Recommender Systems. *Computer* **2009**, *42*, 30–37.
11. Li, P.; Wang, Z.; Ren, Z.; Bing, L.; Lam, W. Neural rating regression with abstractive tips generation for recommendation. In Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval, Shinjuku, Tokyo, Japan, 7–11 August 2017; pp. 345–354.
12. Wu, D.; Luo, X. Robust Latent Factor Analysis for Precise Representation of High-Dimensional and Sparse Data. *IEEE/CAA J. Autom. Sin.* **2021**, *8*, 796–805. [CrossRef]
13. Wu, D.; Shang, M.; Luo, X.; Wang, Z. An L1-and-L2-Norm-Oriented Latent Factor Model for Recommender Systems. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, *33*, 5775–5788. [CrossRef]
14. Luo, X.; Wu, H.; Li, Z. Neulft: A Novel Approach to Nonlinear Canonical Polyadic Decomposition on High-Dimensional Incomplete Tensors. *IEEE Trans. Knowl. Data Eng.* **2023**, *35*, 6148–6166. [CrossRef]
15. Wu, D.; He, Y.; Luo, X.; Zhou, M. A Latent Factor Analysis-Based Approach to Online Sparse Streaming Feature Selection. *IEEE Trans. Syst. Man Cybern. Syst.* **2022**, *52*, 6744–6758. [CrossRef]
16. Wu, D. *Robust Latent Feature Learning for Incomplete Big Data*; Springer Nature: Berlin, Germany, 2022.
17. Liu, H.; Yuan, H.; Hou, J.; Hamzaoui, R.; Gao, W. Pufa-gan: A frequency-aware generative adversarial network for 3d point cloud upsampling. *IEEE Trans. Image Process.* **2022**, *31*, 7389–7402. [CrossRef] [PubMed]
18. Liu, Q.; Yuan, H.; Hamzaoui, R.; Su, H.; Hou, J.; Yang, H. Reduced reference perceptual quality model with application to rate control for video-based point cloud compression. *IEEE Trans. Image Process.* **2021**, *30*, 6623–6636. [CrossRef]
19. Liu, X.; He, J.; Liu, M.; Yin, Z.; Yin, L.; Zheng, W. A Scenario-Generic Neural Machine Translation Data Augmentation Method. *Electronics* **2023**, *12*, 2320. [CrossRef]
20. Liu, X.; Zhao, J.; Li, J.; Cao, B.; Lv, Z. Federated neural architecture search for medical data security. *IEEE Trans. Ind. Inform.* **2022**, *18*, 5628–5636. [CrossRef]
21. Huang, T.; Liang, C.; Wu, D.; He, Y. A Debiasing Autoencoder for Recommender System. *IEEE Trans. Consum. Electron.* **2023**. [CrossRef]
22. Zenggang, X.; Mingyang, Z.; Xuemin, Z.; Sanyuan, Z.; Fang, X.; Xiaochao, Z.; Yunyun, W.; Xiang, L. Social similarity routing algorithm based on socially aware networks in the big data environment. *J. Signal Process. Syst.* **2022**, *94*, 1253–1267. [CrossRef]
23. Xiong, Z.; Li, X.; Zhang, X.; Deng, M.; Xu, F.; Zhou, B.; Zeng, M. A Comprehensive Confirmation-based Selfish Node Detection Algorithm for Socially Aware Networks. *J. Signal Process. Syst.* **2023**, 1–19. [CrossRef]
24. Xie, L.; Zhu, Y.; Yin, M.; Wang, Z.; Ou, D.; Zheng, H.; Liu, H.; Yin, G. Self-feature-based point cloud registration method with a novel convolutional Siamese point net for optical measurement of blade profile. *Mech. Syst. Signal Process.* **2022**, *178*, 109243. [CrossRef]
25. Wu, D.; Sun, B.; Shang, M. Hyperparameter Learning for Deep Learning-based Recommender Systems. *IEEE Trans. Serv. Comput.* **2023**, 1–13. [CrossRef]
26. Chen, F.; Yin, G.; Dong, Y.; Li, G.; Zhang, W. KHGCN: Knowledge-Enhanced Recommendation with Hierarchical Graph Capsule Network. *Entropy* **2023**, *25*, 697. [CrossRef] [PubMed]

27. Muller, L.; Martel, J.; Indiveri, G. Kernelized synaptic weight matrices. In *International Conference on Machine Learning*; PMLR: London, UK, 2018; pp. 3654–3663.
28. Han, H.; Liang, Y.; Bella, G.; Giunchiglia, F.; Li, D. LFDNN: A Novel Hybrid Recommendation Model Based on DeepFM and LightGBM. *Entropy* **2023**, *25*, 638. [CrossRef]
29. Zhang, S.; Yao, L.; Sun, A.; Tay, Y. Deep learning based recommender system: A survey and new perspectives. *ACM Comput. Surv.* **2019**, *52*, 1–38. [CrossRef]
30. Lu, S.; Liu, M.; Yin, L.; Yin, Z.; Liu, X.; Zheng, W. The multi-modal fusion in visual question answering: A review of attention mechanisms. *PeerJ Comput. Sci.* **2023**, *9*, e1400. [CrossRef]
31. Shen, X.; Jiang, H.; Liu, D.; Yang, K.; Deng, F.; Lui, J.C.; Liu, J.; Dustdar, S.; Luo, J. PupilRec: Leveraging Pupil Morphology for Recommending on Smartphones. *IEEE Internet Things J.* **2022**, *9*, 15538–15553. [CrossRef]
32. Shen, Y.; Ding, N.; Zheng, H.-T.; Li, Y.; Yang, M. Modeling relation paths for knowledge graph completion. *IEEE Trans. Knowl. Data Eng.* **2020**, *33*, 3607–3617. [CrossRef]
33. Wang, H.; Wang, B.; Luo, P.; Ma, F.; Zhou, Y.; Mohamed, M.A. State evaluation based on feature identification of measurement data: For resilient power system. *CSEE J. Power Energy Syst.* **2021**, *8*, 983–992.
34. Wang, Y.; Xu, N.; Liu, A.-A.; Li, W.; Zhang, Y. High-order interaction learning for image captioning. *IEEE Trans. Circuits Syst. Video Technol.* **2021**, *32*, 4417–4430. [CrossRef]
35. Luo, X.; Zhou, Y.; Liu, Z.; Zhou, M. Fast and Accurate Non-Negative Latent Factor Analysis of High-Dimensional and Sparse Matrices in Recommender Systems. *IEEE Trans. Knowl. Data Eng.* **2023**, *35*, 3897–3911. [CrossRef]
36. Chen, J.; Xu, M.; Xu, W.; Li, D.; Peng, W.; Xu, H. A Flow Feedback Traffic Prediction Based on Visual Quantified Features. *IEEE Trans. Intell. Transp. Syst.* **2023**, 1–9. [CrossRef]
37. Deng, X.; Liu, E.; Li, S.; Duan, Y.; Xu, M. Interpretable Multi-modal Image Registration Network Based on Disentangled Convolutional Sparse Coding. *IEEE Trans. Image Process.* **2023**, *32*, 1078–1091. [CrossRef]
38. Fan, W.; Yang, L.; Bouguila, N. Unsupervised grouped axial data modeling via hierarchical Bayesian nonparametric models with Watson distributions. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *44*, 9654–9668. [CrossRef] [PubMed]
39. Guo, C.; Hu, J. Fixed-Time Stabilization of High-Order Uncertain Nonlinear Systems: Output Feedback Control Design and Settling Time Analysis. *J. Syst. Sci. Complex.* **2023**, 1–22. [CrossRef]
40. Huang, H.; Xue, C.; Zhang, W.; Guo, M. Torsion design of CFRP-CFST columns using a data-driven optimization approach. *Eng. Struct.* **2022**, *251*, 113479. [CrossRef]
41. Sedhain, S.; Menon, A.K.; Sanner, S.; Xie, L. Autorec: Autoencoders meet collaborative filtering. In Proceedings of the 24th International Conference on World Wide Web, Florence, Italy, 18–22 May 2015; pp. 111–112.
42. Wu, D.; Luo, X.; Shang, M.; He, Y.; Wang, G.; Wu, X. A Data-Characteristic-Aware Latent Factor Model for Web Services QoS Prediction. *IEEE Trans. Knowl. Data Eng.* **2022**, *34*, 2525–2538. [CrossRef]
43. Luo, X.; Zhou, M.; Li, S.; Wu, D.; Liu, Z.; Shang, M. Algorithms of Unconstrained Non-Negative Latent Factor Analysis for Recommender Systems. *IEEE Trans. Big Data* **2021**, *7*, 227–240. [CrossRef]
44. Yuan, Y.; Luo, X.; Shang, M.; Wu, D. A generalized and fast-converging non-negative latent factor model for predicting user preferences in recommender systems. In Proceedings of the Web Conference 2020, Taipei, Taiwan, 20–24 April 2020; pp. 498–507.
45. Ren, X.; Song, M.; Haihong, E.; Song, J. Context-aware probabilistic matrix factorization modeling for point-of-interest recommendation. *Neurocomputing* **2017**, *241*, 38–55. [CrossRef]
46. Wu, D.; Jin, L.; Luo, X. PMLF: Prediction-Sampling-Based Multilayer-Structured Latent Factor Analysis. In Proceedings of the 2020 IEEE International Conference on Data Mining (ICDM), Sorrento, Italy, 17–20 November 2020; pp. 671–680.
47. Wu, D.; Luo, X.; He, Y.; Zhou, M. A Prediction-Sampling-Based Multilayer-Structured Latent Factor Model for Accurate Representation to High-Dimensional and Sparse Data. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, 1–14. [CrossRef]
48. Wang, C.; Liu, Q.; Wu, R.; Chen, E.; Liu, C.; Huang, X.; Huang, Z. Confidence-aware matrix factorization for recommender systems. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; pp. 434–442.
49. Wu, D.; He, Q.; Luo, X.; Shang, M.; He, Y.; Wang, G. A Posterior-Neighborhood-Regularized Latent Factor Model for Highly Accurate Web Service QoS Prediction. *IEEE Trans. Serv. Comput.* **2022**, *15*, 793–805. [CrossRef]
50. Wu, D.; Zhang, P.; He, Y.; Luo, X. A Double-Space and Double-Norm Ensembled Latent Factor Model for Highly Accurate Web Service QoS Prediction. *IEEE Trans. Serv. Comput.* **2023**, *16*, 802–814. [CrossRef]
51. Leng, C.; Zhang, H.; Cai, G.; Cheng, I.; Basu, A. Graph regularized Lp smooth non-negative matrix factorization for data representation. *IEEE/CAA J. Autom. Sin.* **2019**, *6*, 584–595. [CrossRef]
52. Wu, D.; Luo, X.; Shang, M.; He, Y.; Wang, G.; Zhou, M. A Deep Latent Factor Model for High-Dimensional and Sparse Matrices in Recommender Systems. *IEEE Trans. Syst. Man Cybern. Syst.* **2021**, *51*, 4285–4296. [CrossRef]
53. Sun, B.; Wu, D.; Shang, M.; He, Y. Toward auto-learning hyperparameters for deep learning-based recommender systems. In Proceedings of the International Conference on Database Systems for Advanced Applications: 27th International Conference, DASFAA 2022, Virtual Event, 11–14 April 2022; Proceedings, Part II. Springer: Cham, Switzerland, 2022; pp. 323–331.
54. Wang, Q.; Peng, B.; Shi, X.; Shang, T.; Shang, M. DCCR: Deep collaborative conjunctive recommender for rating prediction. *IEEE Access* **2019**, *7*, 60186–60198. [CrossRef]

55. Zhang, M.; Chen, Y. Inductive matrix completion based on graph neural networks. In Proceedings of the International Conference on Learning Representations, Formerly Addis Ababa, Ethiopia, 26 April–1 May 2020.
56. He, X.; Chua, T.-S. Neural factorization machines for sparse predictive analytics. In Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Tokyo, Japan, 7–11 August 2017; pp. 355–364.
57. Han, S.C.; Lim, T.; Long, S.; Burgstaller, B.; Poon, J. GLocal-K: Global and Local Kernels for Recommender Systems. In Proceedings of the 30th ACM International Conference on Information & Knowledge Management, Virtual Event, Australia, 1–5 November 2021; pp. 3063–3067.
58. Xiao, J.; Ye, H.; He, X.; Zhang, H.; Wu, F.; Chua, T.-S. Attentional factorization machines: Learning the weight of feature interactions via attention networks. In Proceedings of the 26th International Joint Conference on Artificial Intelligence, IJCAI, Melbourne, Australia, 19–25 August 2017; pp. 3119–3125.
59. Kim, D.H.; Park, C.; Oh, J.; Lee, S.; Yu, H. Convolutional matrix factorization for document context-aware recommendation. In Proceedings of the 10th ACM Conference on Recommender Systems, Boston, MA, USA, 15–19 September 2016; pp. 233–240.
60. Wu, D.; Luo, X.; Wang, G.; Shang, M.; Yuan, Y.; Yan, H. A Highly Accurate Framework for Self-Labeled Semisupervised Classification in Industrial Applications. *IEEE Trans. Ind. Inform.* **2018**, *14*, 909–920. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.