electronics

MDPI

Article

Enhancing Recommendation Diversity and Novelty with Bi-LSTM and Mean Shift Clustering

Yuan Yuan, Yuying Zhou, Xuanyou Chen, Qi Xiong * and Hector Chimeremeze Okere

International College, Hunan University of Arts and Sciences, Changde 415000, China; yuanyuan@st.huas.edu.cn (Y.Y.); 202209010308zyy@st.huas.edu.cn (Y.Z.); chenxuanyou@st.huas.edu.cn (X.C.); okerejunior@gmail.com (H.C.O.)

* Correspondence: xiongqi@huas.edu.cn; Tel.: +86-18873688996

Abstract: In the digital age, personalized recommendation systems have become crucial for information dissemination and user experience. While traditional systems focus on accuracy, they often overlook diversity, novelty, and serendipity. This study introduces an innovative recommendation system model, Time-based Outlier Aware Recommender (TOAR), designed to address the challenges of content homogenization and information bubbles in personalized recommendations. TOAR integrates Neural Matrix Factorization (NeuMF), Bidirectional Long Short-Term Memory Networks (Bi-LSTM), and Mean Shift clustering to enhance recommendation accuracy, novelty, and diversity. The model analyzes temporal dynamics of user behavior and facilitates cross-domain knowledge exchange through feature sharing and transfer learning mechanisms. By incorporating an attention mechanism and unsupervised clustering, TOAR effectively captures important time-series information and ensures recommendation diversity. Experimental results on a news recommendation dataset demonstrate TOAR's superior performance across multiple metrics, including AUC, precision, NDCG, and novelty, compared to traditional and deep learning-based recommendation models. This research provides a foundation for developing more intelligent and personalized recommendation services that balance accuracy with content diversity.

Keywords: personalized recommendation systems; echo chambers; bidirectional long short-term memory; neural matrix factorization; mean shift

Citation: Yuan, Y.; Zhou, Y.; Chen, X.; Xiong, Q.; Okere, H.C. Enhancing Recommendation Diversity and Novelty with Bi-LSTM and Mean Shift Clustering. *Electronics* **2024**, *13*, 3841. https://doi.org/10.3390/electronics13193841

Academic Editor: George Angelos Papadopoulos

Received: 7 September 2024
Revised: 26 September 2024
Accepted: 27 September 2024
Published: 28 September 2024

1. Introduction

1.1. Research Background

Personalized recommendation systems have become a core component of daily life in the modern era, significantly altering methods of information retrieval [1,2]. These systems provide convenient and accurate content recommendations, greatly simplifying the process for users to find relevant information. However, with the evolution of recommendation systems, their objectives have gradually shifted towards enhancing accuracy and personalization to improve user retention. Nevertheless, this excessive emphasis on accuracy has resulted in a noticeable deficiency in the diversity and novelty of recommendations. Recommendation systems that lays excessive emphasis on accuracy can lead to the phenomenon known as the "Echo Chamber", confining users to narrow information domains. This prevents and obstructs users from exploring potentially new and interesting contents termed "dark information" [3,4]. In other words, this phenomenon limits users' informational horizons and their exposure to new knowledge and diverse viewpoints. To address this challenge of information bubbles, scholars have proposed methods to enhance recommendation systems' performance in diversity, serendipity, and novelty [5,6]. Diversity in recommendation systems is the variance among items on the recommendation list. A highly diverse recommendation list offers a wide range of items, encouraging users to explore new content and reducing content monotony. Studies have shown that enhancing

diversity meets users' broad interests and needs, particularly for those with wide ranging interests or undefined preferences [7]. Consequently, researchers have proposed models to increase recommendation systems' novelty. For instance, Serendipitous Personalized Ranking (SPR) extends traditional methods by de-emphasizing item popularity in AUC (Area Under the Curve) optimization [8].This method enhances recommendation serendipity—recommending unexpected yet interesting items, thus boosting user satisfaction and exploration. Auralist balances accuracy and novelty through topic modeling [9]. The Determinantal Point Process (DPP) is a mathematical model that diversifies recommendations via greedy maximum a posteriori (MAP) inference [10]. DPP considers item similarity to create a high-quality, diverse recommendation list, reducing redundancy. Panagiotis et al. proposed a model named HOM-LIN, which uses a hybrid utility function combining estimated scores and serendipity to provide recommendations [11]. This method aims to offer recommendations that match the preferences users have already expressed and possess a certain level of serendipity, increasing user satisfaction and the attractiveness of the recommendation system. All these models have successfully improved the serendipity metric in recommendations.

Despite their established positive outcomes, these techniques have their drawbacks. SPR may overemphasize niche items, overlooking the broad appeal of some popular items. While Auralist balances accuracy and novelty, insufficiently refined theme selection could decrease recommendation relevance. The DPP method requires accurate item similarity estimates; inaccuracies can decline recommendation list quality. Adjusting HOM-LIN can be challenging due to the need for a meticulous balance between scores and serendipity; an overemphasis on serendipity may compromise user experience.

Therefore, recommendation systems must comprehensively consider multiple factors and adopt refined strategies to balance long-term interests and novel exploration needs. Simultaneously, enhancing recommendation diversity and serendipity should not sacrifice user experience.

1.2. Paper Contributions

To address those challenges mentioned above, the study introduces a novel recommendation system model. The main innovations of the study are as follows:

- A composite recommendation system is proposed, integrating the Neural Matrix Factorization (NeuMF) structure, which is based on neural collaborative filtering, with a sequence model that incorporates an attention mechanism and Mean Shift clustering. This innovative combination is adept at accurately capturing the temporal dynamics of user behavior, while simultaneously infusing diversity and novelty into the recommendation process. Momentously, this integration not only focuses on users' current preferences but also the dynamic shifts and long-term trends in user interests, thereby offering a more comprehensive and enriched information experience.
- Transfer learning and feature sharing mechanisms are adopted: The model incorporates feature embedding technology to represent users and items, facilitating feature sharing and transfer across different models. This enhances the model's generalization ability and effectively prevents overfitting. This method improves recommendation accuracy, stability, and scalability.
- The Attentive-LSTM model is applied: Leveraging Bidirectional Long Short-Term Memory Networks (Bi-LSTM) with an integrated attention mechanism, the model more effectively captures important time-series information, enhancing prediction accuracy and stability. The introduction of an attention mechanism allows the model to weigh information at various time steps according to the sequence's importance, thereby enhancing prediction accuracy.
- Unsupervised clustering is used to enhance recommendation diversity and novelty: By applying unsupervised clustering to recently interacted items and calculating the weighted distance from each cluster center to the target item, model diversity and novelty are ensured. This approach not only increases the serendipity and diversity

of recommendations, but also offers users unexpected discoveries, enhancing the system's attractiveness.

- The algorithm model mentioned in this paper, TOAR, known as the Time-Aware Outlier-aware Recommender, is distinguished for integrating two core features: time series analysis and outlier detection in the context of recommendation systems. By employing a Bidirectional Long Short-Term Memory (Bi-LSTM) network, TOAR captures the temporal dynamics of user behavior, enabling it to comprehend the evolution of user preferences and deliver timely recommendations. Additionally, the model utilizes the Mean Shift clustering algorithm to identify "outlier" items that differ from the majority of the user's interactions, thereby introducing a greater diversity and novelty in recommendations. This design, which incorporates both temporal sensitivity and outlier awareness, not only enhances the accuracy and diversity of the recommendation system, but also mitigates the "echo chamber" effect commonly observed in traditional recommendation approaches, offering users a richer content experience.

2. Related Works

All literature on recommendation systems from the past five years has been reviewed. The technologies discussed in this literature can generally be categorized into three groups: first, traditional algorithms which encompass methods that do not utilize deep learning; second, deep learning algorithms; and third, hybrid recommendation models that combine deep learning techniques with traditional methods to optimize performance and results.

2.1. Traditional Recommendation Model

Modern recommendation system research primarily focuses on two non-deep learning strategies: collaborative filtering and content-based recommendations. Collaborative Filtering holds an indispensable position as one of the most widely applied algorithms in recommendation systems [12]. It generates recommendations using a matrix of user–item associations based on historical user behavior data. Its notable advantages include a simple structure and efficient utilization of rich user behavior data. However, collaborative filtering faces challenges such as the cold start problem and the assumption that user interests rely solely on past behaviors, neglecting the dynamic nature of user interests and the current contextual environment [13].

Content-based recommendation algorithms analyze characteristics of previously liked items to recommend similar items to users. This method's advantages include intuitiveness and easy implementation, making it particularly suitable for initial recommendation system stages. It emphasizes the match between item features and user preferences, ensuring user independence and protection against manipulations such as inflating product rankings through multiple accounts [14]. In addition, content-based methods effectively recommend new items without relying on a user's interaction history. However, its limitations include the difficulty in item feature extraction, low recommendation diversity, and the cold start challenge for new users [15].

However, non-deep learning recommendation methods possess both advantages and limitations. To address these limitations and improve recommendation system performance, researchers are exploring new methods, including integrating various recommendation strategies with deep learning technologies [16]. The goal of these explorations is to achieve a more accurate, personalized, and dynamically adaptive recommendations that are responsive to changes in user needs.

2.2. Deep Learning Model

Recommendation systems have evolved from basic machine learning applications to the advanced realm of deep learning recommendation systems, undergoing several stages of development. Many algorithms developed during this evolution continue to serve as baselines and comparison points in recommendation algorithm research.

For instance, Deep Neural Networks (DNNs) and Convolutional Neural Networks (CNNs) have demonstrated exceptional capabilities in processing users' text and image information, offering new perspectives for content-based recommendations [17,18]. Moreover, Recurrent Neural Networks (RNNs), by learning serialized user behavior data, can better understand and predict the dynamic changes in user interests [19]. In recent years, numerous scholars have proposed novel architectures for neural networks. For instance, in the work titled "Dynamic Intent-aware Iterative Denoising Network for Session-based Recommendation", the authors introduce the DIDN (Dynamic Intent-aware Iterative Denoising Network) model. This model represents an innovative session-based recommendation algorithm designed to address two major challenges in session-based recommendation: the dynamic nature of user intent and the uncertainty in user behavior. Additionally, in "CL4CTR: A Contrastive Learning Framework for CTR Prediction", the authors incorporate self-supervised learning to directly generate high-quality feature representations. They proposed a model-agnostic contrastive learning framework for CTR prediction (CL4CTR), which utilizes three self-supervised learning signals—contrastive loss, feature alignment, and domain uniformity—to regularize feature representation learning.

Despite deep learning's powerful tools for enhancing recommendation system performance, challenges remain, including poor model interpretability, high training costs, and a substantial need for labeled data. Consequently, researchers are investigating ways to effectively integrate deep learning with traditional recommendation algorithms and address these challenges through technological innovation [20].

2.3. Clustering Algorithms

In recent years, clustering algorithms have demonstrated significant advantages and promising applications in personalized recommendation systems. By clustering users or items, recommendation systems can more effectively capture users' latent preferences, thereby improving the accuracy and diversity of recommendations. The application of clustering methods in recommendation systems has been extensively studied, encompassing traditional clustering algorithms such as K-means and hierarchical clustering, as well as more sophisticated approaches that have emerged in recent years, like multiview clustering and deep learning-based clustering methods.

According to the study by Pang et al. (2023) [21], a novel multiview clustering method is proposed, which enhances recommendation performance by integrating data from different perspectives. This approach leverages clustering to capture user features across multiple views, leading to more precise personalized recommendations. This indicates the significant value of multiview information fusion in complex recommendation scenarios.

Another study [22,23] introduced an ensemble clustering method that combines global and local structural information, demonstrating that fully utilizing both global and local information in complex multimodal data spaces can significantly enhance clustering performance and, consequently, improve the performance of recommendation systems. This method integrates global and local structural information into a unified learning framework through a self-representation model and a CA matrix self-enhancement model, using the Hadamard product to maximize their commonalities, resulting in outstanding performance in recommendation systems.

Moreover, recent studies have also explored the application of clustering methods, such as graph-based models, tensor projection methods, and the combination of self-supervised learning and clustering in recommendation systems. For instance, some research proposes improving clustering outcomes through graph embeddings and social network analysis, thereby providing more accurate user and item groupings for recommendation systems. Additionally, tensor-decomposition-based clustering methods have shown excellent performance in handling high-dimensional data, effectively uncovering hidden user interests in recommendation systems.

Overall, the application of clustering methods in recommendation systems has evolved from traditional simple clustering to the integration of multiview, multimodal, deep learn-

ing, and self-supervised learning techniques. This progression highlights the enormous potential of clustering methods in handling complex data and enhancing personalized recommendation performance. These studies provide rich theoretical and practical support for the design of future recommendation systems.

2.4. Hybrid Recommendation Model

Hybrid models are crucial in recommendation system research, combining deep learning methods and traditional algorithms to address their limitations and fully leverage their strengths. This approach integrates techniques from content-based recommendations, collaborative filtering, and deep learning, significantly improving recommendation accuracy and addressing cold start and data sparsity issues [24].

(1). Hybrid Neural Networks: The Neural Collaborative Filtering (NCF) algorithm, proposed by Rendle et al., is a classic example of a hybrid model [25]. It combines traditional matrix factorization techniques with Multilayer Perceptrons (MLP) to learn the non-linear relationships in user–item interactions, demonstrating how deep learning can enhance the performance of traditional algorithms.

(2). Sequential Recommendation Models: As user behavior is serialized with subsequent actions influenced by preceding ones, sequential recommendation models utilize Long Short-Term Memory networks (LSTM) or Transformers to analyze user behavior sequences. These models capture the temporal dependency of actions, yielding more dynamic and personalized recommendations [26].

(3). Cross-Feature Learning Models: Cross-feature learning models employ deep learning networks to understand interactions between user and item features, adept at tackling sparse data challenges. The Wide and Deep model exemplifies this by merging a wide linear model (for memorizing feature interactions) with deep neural networks (for examining non-linear feature relationships), thereby enriching recommendation breadth and depth [27].

Adopting this comprehensive strategy, hybrid models enhance the accuracy and efficiency of recommendation systems, while also improving their flexibility and robustness for adaptation to diverse domains and complex scenarios. This study adopts hybrid model strategies, improving both accuracy and serendipity by combining deep learning with traditional algorithms. The success of these models demonstrates their immense potential by integrating various recommendation technologies, offering new directions for future recommendation system development.

3. Materials and Methods

The proposed model, as illustrated in Figure 1, is composed of two main components: the NeuMF part on the left and the combination of the Attentive-LSTM and Mean-Shift components on the right. The model's inputs include three primary elements: item features, user features, and the user's historical behavior sequence. Initially, these inputs are encoded into vector representations. Subsequently, the model selects the most recent two behavior vectors from the user's historical behavior sequence. These vectors are processed through the Attentive-LSTM module, which aims to capture the temporal characteristics and key actions within the user's behavior.

On the right side, the Mean-Shift algorithm is employed to cluster historical behavior vectors and calculate the weighted distance between the target item and the cluster centers to assess the item's serendipity and novelty. The processed features from the Mean-Shift module are concatenated with the item features rather than simply aggregated. These combined features are further integrated and learned through a multilayer network. Specifically, the NeuMF component on the left is responsible for training the basic item and user features, which are then transmitted via a transition matrix to the more complex network on the right, where they are combined with the outputs from the Mean-Shift and Attentive-LSTM modules.

In the complex network on the right, the processed historical behavior and group preference features are integrated and subjected to non-linear transformation via a multilayer perceptron (MLP). Ultimately, these pieces of information converge, and the model outputs a predicted user preference score for a specific item. This output represents a synthesis of individual preferences, group influences, and historical behavior features, with the goal of enhancing the overall performance of the recommendation system, particularly in terms of accuracy, novelty, and serendipity.

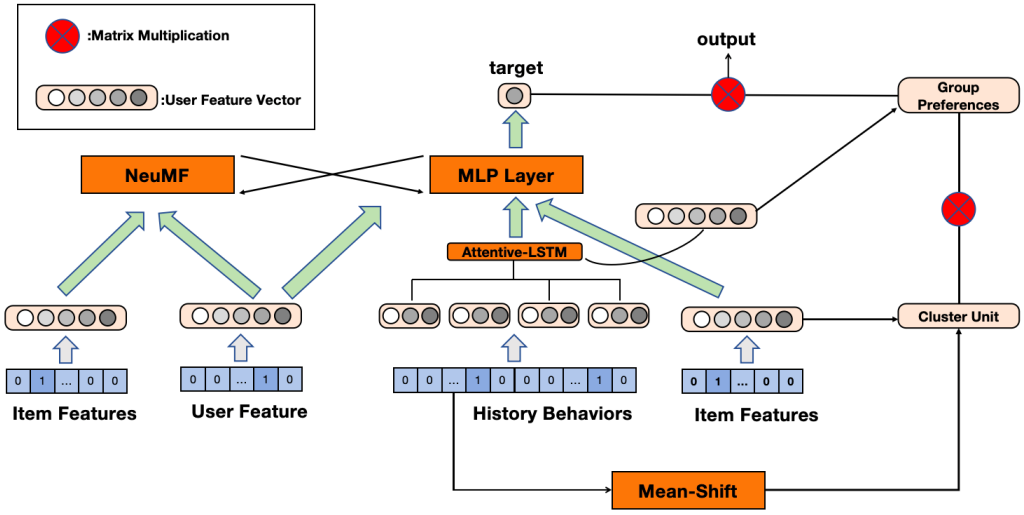

Figure 1. System overall structure.

In the proposed model, the relationships between the inputs and outputs of each component are as follows. First, the input item features, user features, and user historical behavior sequences are encoded as vectors, which are then fed into the respective modules of the model. For the Attentive-LSTM component, the input consists of the two most recent behavior vectors from the user's historical behavior sequence, and the output is a weighted representation that captures temporal characteristics and key behaviors. These outputs are passed to the MLP layer. Separately, the historical behavior vectors are also processed by the Mean-Shift module, which computes the weighted distance between the target item and the cluster centers. The output from the Mean-Shift module is then concatenated with the item feature vectors, creating a combined representation that evaluates the item's novelty and serendipity. In the NeuMF component on the left, the input comprises the basic item and user features, which, after training, produce features that are transmitted via a transition matrix to the complex network on the right. In the final MLP layer, all outputs from the preceding modules, including the processed historical behavior and group preference features, are integrated and undergo non-linear transformation, resulting in the output of a predicted user preference score for the specific item.

3.1. Attentive-LSTM

The initial step in the model's implementation involves extracting user–item features as well as features from users' historical behavior sessions. Users u and items i are encoded using one-hot encoding, translating categorical variables into vectors encoded vectors $I_u \in \{0,1\}^m$ and $I_i \in \{0,1\}^{n_T}$, where each one-hot encoded vector is 1 at the ID index position and 0 at all other positions. Subsequently, this document transforms these one-hot vectors into embedded representations of users and items using the user embedding matrix $X \in R^{m \times d}$ and the item embedding matrix $Y \in R^{n_T \times d}$, respectively, where d represents the dimensionality post-embedding. These are then passed through different parallel neural network layers as input features, denoted as $\tilde{I_{u,i}} = [XI_u, YI_i] \in R^{1 \times 2d}$. The encoding process of the modeling features can be depicted as $(u,i) -> I_u, I_i -> I_{u,i}$. Concurrently, Long Short-Term Memory (LSTM) is employed to extract features from users' historical behavior

sessions: representing the users' historical item clicks as a sequence $S_u = [i_1, i_2, \ldots i_k]$, the preference prediction model retrieves a fixed number K of representation vectors for the users' historical item clicks, aggregating all representation vectors onto the user behavior sequence S_u, and then obtaining sequence embeddings using the LSTM neural network. Utilizing Long Short-Term Memory neural networks to simulate user interests is crucial, as they can capture temporal information and the order of user purchases, enabling recent behaviors to exert a more significant influence compared to earlier behaviors. Moreover, due to the long-term memory capability of LSTM, it can somewhat address issues such as gradient explosion and gradient vanishing in lengthy historical sequences.

It is important to recognize that in processing users' historical interactions, each item from past interactions can exert a unique influence on current recommendations. Therefore, to effectively discern the diversity among items in historical behaviors, this study utilizes a Self-Attentive MLP in its modeling process . This method adeptly assigns distinct weights to each historical item, thereby enhancing the precision of predictions for new content that may interest users.

The attention mechanism operates in the following manner: First, a compatibility function computes the similarity score e_{ti} between each input vector x_i in the sequence and the target behavior embedding x_t at timestep t . These similarity scores reflect how relevant each historical behavior is to the current context. The scores are then normalized using the softmax function to produce the attention weights a_{ti} , which quantify the relative importance of each behavior in contributing to the final output. Finally, these attention weights are applied to the input vectors, resulting in a weighted sum O_t , which serves as the output of the attention mechanism, capturing the key behaviors and their temporal characteristics effectively. The specific formulas are as follows:

$$a_{ti} = \frac{\exp e_{ti}}{\sum_{i-1}^{n} \exp e_{ti}} \tag{1}$$

$$O_t = \sum_{i=1}^{k} a_{ti}\left(x_i W^t\right) \tag{2}$$

3.2. Mean Shift

The Mean Shift clustering algorithm is an iterative process that locates the maximum value points of a density function using given discrete data. Incorporating Mean Shift clustering in the model aims to enhance the novelty of the model. The greatest advantage of this algorithm is that it is an unsupervised clustering algorithm, and it has a strong analytical capability for applications in complex multimodal feature spaces. When using the Mean shift algorithm, it is necessary to define the kernel function $K(i_i - i)$, where i_i is the initial estimate and i is the observation. This kernel function is used to determine the weight of the neighborhood points on the mean re-estimation, which is usually modeled with a Gaussian distribution.

The weighted mean of the density within the window defined by the kernel function K can be calculated as follows:

$$m(i) = \frac{\sum_{i \in N(i)} i_i K(i_i - i)}{\sum_{i \in N(i)} K(i_i - i)} \tag{3}$$

where N_i represents the neighborhood of i. The Mean Shift algorithm updates i to m_i and repeats this process until m_i converges.

For each user u, their historical interaction sequence $\{i_1, i_2 \ldots i_n\}$is extracted alongside the corresponding latent space embeddings $\{w_1, w_2 \ldots w_n\}$ via sequence modeling. Subsequently, the Mean Shift algorithm is employed to partition these embeddings into clusters reflective of user interests $\{C_1, C_2 \ldots C_n\}$. For each new item recommendation $i*$

for user u, we model the serendipity as the weighted average distance between the new item embedding $w*$ and each cluster Ck:

$$\text{unexpi} *, \text{u} = \frac{\sum_{\text{k}=1}^{\text{N}} \text{d}(\text{w}*, \text{Ck}) \times |\text{Ck}|}{\sum_{\text{k}=1}^{\text{N}} (\text{Ck})}, \tag{4}$$

where $\text{d}(\text{w}*, \text{Ck})$ denotes the distance between the embedding of the new item $w*$ and cluster Ck, while $|Ck|$ signifies the quantity of points within cluster Ck. Concisely, should the newly recommended item diverge substantially from the user's historical interests, the recommendation is deemed to possess elevated serendipity. N is the number of clusters.

3.3. Feature Sharing

As shown in Figure 2, feature sharing technology enables the dissemination of information across various models or tasks, thereby augmenting learning efficacy and performance. This process often entails the distribution of weights or feature representations among disparate models or segments within a single model. Such an approach permits the assimilation of general information, diminishing the necessity for redundant learning of identical features. Crucial in domains like multitask learning, transfer learning, and recommendation systems, feature sharing bolsters models' capacity to generalize to novel tasks or datasets, fortifying their resilience and operational efficiency.

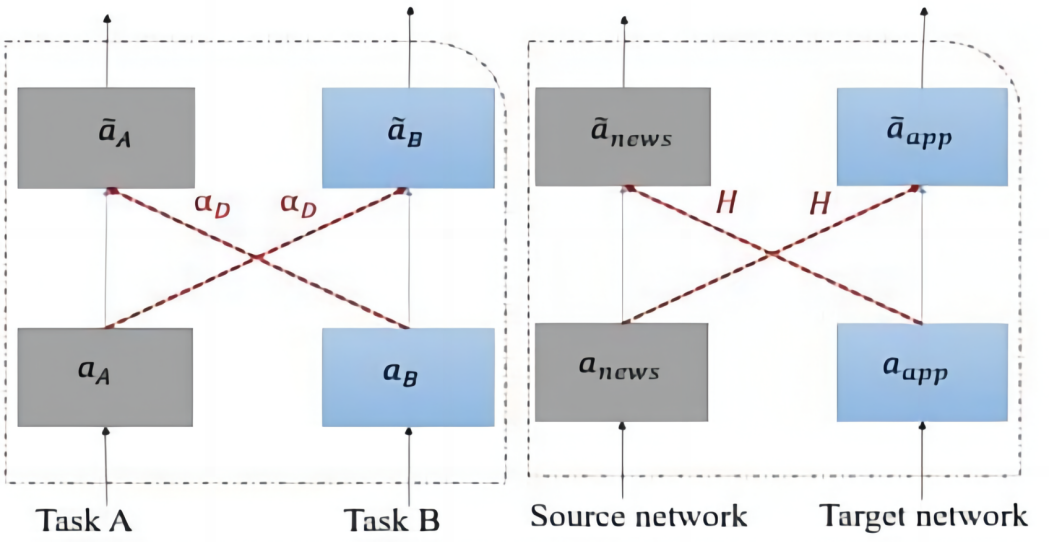

Figure 2. The knowledge transfer process.

$$I_T^{l+1\prime} = W_{TTT}^l + b_T^l \tag{5}$$

$$I^{l+1}{}_T'' = M^l I^l{}_S \tag{6}$$

$$I^{l+1\prime}{}_S = W^l{}_S I_{SS} + b_S^l \tag{7}$$

$$I_S^{l+1\prime\prime} = M^l I_T^l \tag{8}$$

where $W^l{}_T$ and $W^l{}_S$ are the weight matrices of the I to $I+1$ hidden layer in the target and source domains, $I^l{}_T$ and $I^l{}_S$ are the input representations of the Ith hidden layer in the target and source domains, respectively, $b^l{}_T$ and $b^l{}_S$ are the bias entries of the Ith layer, and matrix M^l is the knowledge migration matrix shared by the two models in the Ith layer, which corresponds to the linear projection of the cross-connections controlling the input information from the source domain to the target domain and vice versa.

The TOAR model integrates the left-side knowledge transfer score with the right-side item unexpectedness score (i.e., unexp), which is calculated after clustering, using a comprehensive utility function. This approach enhances the recommendation accuracy through

knowledge transfer from the source domain while also increasing the unexpectedness of the recommendations. As a result, the model generates recommendations that are both relevant and surprising to the user. The objective function is as follows:

$$\text{Loss}\left(\theta \mid R^{+} \cup R^{-}\right) = -\sum_{(u,i)\in R^{+}\cup R^{-}} r_{u,i} \ln \hat{r}_{u,i} + (1 - r_{u,i}) \ln(1 - \hat{r}_{u,i}) \tag{9}$$

where R^{+},R^{-}represent the positive and negative samples in the user–item interaction matrix, respectively, $r_{u,i}$ denotes the true labels of the samples, and $\hat{r}_{u,i}$ represents the predicted scores by the model.

The predicted score $\hat{r}_{u,i}$ in our model is calculated as a synthesis of various features processed through different components of the model. Specifically, it is derived from the integration of individual preferences, group influences, and historical behavior features. Mathematically, the predicted score can be expressed as:

$$\hat{r}_{u,i} = \sigma\left(\text{MLP}\left(h_{u,i}^{\text{NeuMF}} \parallel h_{u}^{\text{AttLSTM}} \parallel h_{u,i}^{\text{MS}}\right)\right) \tag{10}$$

where $\text{MLP}(\cdot)$ denotes a multilayer perceptron that learns non-linear interactions among the input features, $\parallel$ represents the concatenation operation, $h_{u,i}^{\text{NeuMF}}$ is the embedding representation from the Neural Matrix Factorization component, capturing collaborative filtering information between user u and item i, $\mathbf{h}_{u}^{\text{AttLSTM}}$ is the output from the Attentive-LSTM module, which models the temporal dynamics of the user's historical behavior with an attention mechanism to highlight significant actions, and $h_{u,i}^{\text{MS}}$ is the output from the Mean-Shift clustering component, representing the novelty and diversity by assessing the distance between the target item and the centers of clusters formed from recent user interactions.

4. Experiments

4.1. Datasets

The experimental data are sourced from the Globo news recommendation dataset[28], delineated in Table 1, chosen for its ability to robustly assess recommendation system performance, attributable to its richness and complexity. The dataset encompasses a wide range of user interactions, including clicks and impressions, which are critical for understanding user engagement with the content. It contains detailed records such as user IDs, article IDs, timestamps of interactions, and user demographics, offering a granular view of user behavior. Additionally, the dataset logs the number of impressions each article received, which is the count of times an article was displayed to a user, regardless of whether it was clicked or not. This inclusion of impressions, alongside clicks, provides a more nuanced understanding of user interest and allows for the analysis of implicit feedback in recommendation algorithms. Regarding this dataset, a random selection of 20% is allocated as the test set, with the balance of 80% designated as the training set, enabling a thorough evaluation of the recommendation models on both explicit and implicit user feedback.

Table 1. Dataset's main characteristics.

Category	Clicks	Impressions	CTR (Click-Through-Rate)	Density	Avg User Interactions
News	1931	607	281,600	0.2403	146

4.2. Experiment Environment

The experiments were conducted on a computing system equipped with an AMD Ryzen 5 3550H processor, Radeon Vega Mobile Graphics, 16 GB RAM, and an NVIDIA GeForce GTX 1650 GPU, sourced from manufacturers based in Santa Clara, CA, USA. The software environment utilized for these experiments consisted of Ubuntu 20.04 LTS as

the operating system, with Python 3.8 for programming. For deep learning and numerical computing, TensorFlow 2.4.0 and NumPy 1.19.4 were used, respectively, alongside Pandas 1.1.5 for data manipulation.

4.3. Experiment's Configuration

In this experiment, the model is constructed by setting several key parameters. The hidden layer size is set to 64, and the long-term and short-term memory windows are 10 and 3, respectively, for capturing temporal features of user behavior. The batch size and the number of source domain items are dynamically passed in to adapt to different data sizes. The MLP is configured as a five-layer model with the number of neurons [64, 32, 16, 8] in order from the input layer to the output layer. The crossover layer is set to 1, the weights are initialized with a standard deviation of 0.01, the classification size is 1, and ReLU is used as the activation function.

4.4. Evaluation Standards

To comprehensively evaluate our proposed time-series recommender system, the study uses the following standardized and accepted evaluation metrics:

(1). AUC (Area Under the Curve)

AUC is a metric used to measure the performance of a model in a binary classification problem. In recommender systems, AUC can be used to evaluate the model's ability to assess whether a user will interact with an item or not:

$$\mathrm{AUC} = \int_0^1 \mathrm{TPR}\left(\mathrm{FPR}^{-1}(\mathrm{t})\right)\mathrm{dt} \tag{11}$$

(2). Precision (PC)

Precision measures how many of the recommended items are actually of interest to the user. It is concerned with the accuracy of the recommendations. Furthermore, Recall measures the proportion of items that users are interested in that the recommendation system successfully recommends:

$$\mathrm{Precision} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}} \tag{12}$$

$$\mathrm{Recall} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} \tag{13}$$

$$\mathrm{PC} = \frac{\mathrm{Precision} * \mathrm{Recall}}{\mathrm{Precision+Recall}} \tag{14}$$

(3). Normalized Discounted Cumulative Gain (NDCG)

NDCG is a metric used to measure the ranking quality of a recommender system. It takes into account the relevance of the recommended items and their position in the recommendation list:

$$\mathrm{DCG} = \sum_{\mathrm{i}=1}^{\mathrm{K}} \frac{2^{\mathrm{rel_i}-1}}{\log_2(\mathrm{i}+1)} \tag{15}$$

$$\mathrm{IDCG} = \sum_{\mathrm{i}=1}^{\mathrm{K}} \frac{1}{\log_2(\mathrm{i}+1)} \tag{16}$$

$$\mathrm{NDCG} = \frac{\mathrm{DCG}}{\mathrm{IDCG}} \tag{17}$$

where rel_i is the relevance of the ith item and K is the first K items evaluated.

(4). Mean Reciprocal Rank (MRR)

MRR measures the average rank position of a user's preferred recommended item in the recommended list. This focuses on the speed of the user's first interaction:

$$\mathrm{MRR} = \frac{1}{\mathrm{rank}} \tag{18}$$

In the formula, rank is the ranking of the first correct item.

(5). Unexpectedness(UNCP)

Unexpectedness focuses on the novelty and unexpectedness of recommended content:

$$\mathrm{UNCP} = \sum_{\mathrm{i}=1}^{\mathrm{K}} \mathrm{unceppness(i)} \tag{19}$$

where $unceppness(i)$ is the unexpectedness score of the i_{th} item.

(6). Novelty (NOV)

This metric assesses the novelty of the recommended content, i.e., recommendations that the user has not been exposed to before:

$$\mathrm{NOV} = -\sum_{\mathrm{i}} \frac{\mathrm{C(i)}}{\mathrm{N}} \log_2\left(\frac{\mathrm{C(i)}}{\mathrm{N}}\right) \tag{20}$$

(7). Maximun Mean Discrepancy Loss (MMD loss)

MMD Loss is a measure of the difference between two different data distributions. In recommender systems, especially where time series data are involved, MMD Loss can be used to assess the sensitivity and adaptability of a model to dynamic changes in user behavior. Specifically, it measures the difference between the distribution predicted by the model and the actual user behavior distribution. By minimizing this discrepancy, MMD Loss helps ensure that the model more accurately captures and reflects the latest preferences and behavioral patterns of users.

4.5. Results and Analysis

In Figure 3, the research model performs well in several metrics. In particular, the model demonstrated its efficiency and accuracy in processing news recommendations in terms of AUC and Precision–Recall. The model is noted for its significant performance improvement in the initial phase, with the performance growth stabilizing as training progresses.

As shown in Figure 3, the recommendation system demonstrates proficiency with a high training precision of over 70%, signifying its ability to accurately identify relevant items. The AUC value peaks at 62%, indicating a decent discriminative power. Test performance is consistent, with precision and NDCG both steady at around 35%, suggesting the system maintains its relevance with unseen data. MRR on the training data are near perfect, while test MRR suggests relevant recommendations are typically positioned high in the ranking. Novelty scores in the test set are over 20%, indicating the system's strength in suggesting new items to users. The loss has reduced by 29%, from approximately 0.48 to below 0.34, indicating enhanced model accuracy over time. Overall, the system showcases solid learning and generalization capabilities, with room for improvement in boosting test precision and NDCG, and in maintaining novelty in recommendations.

4.6. Comparative Experiment

Table 2 presents a comprehensive comparison of our TOAR model against traditional recommendation models (Pop, ItemKNN, BPR), deep learning models (NeuMF, DeepFM), and hybrid models (WideDeep, DiffRec, EulerNet, FEARec, LDiffRec). The comparison is based on six key metrics: AUC, Precision (Pc), NDCG, MRR, UNCP, and NOV. These metrics collectively evaluate the accuracy, ranking quality, diversity, and novelty of the recommendations.

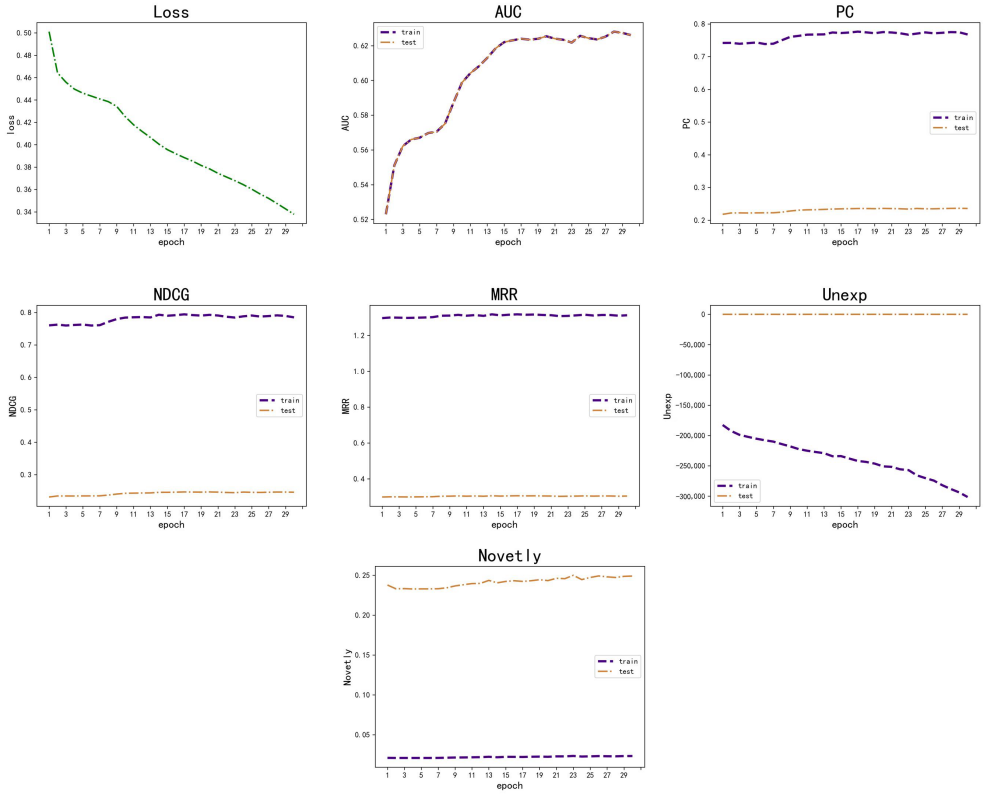

Figure 3. Experiment results.

Table 2. Model comparison.

Metrics/Models	Method	AUC	Pc	NDCG	MRR	UNCP	NOV
Traditional Recommendation Models	Pop [29]	0.4689	0.1591	0.1712	0.3038	0.1585	0.2391
	Item KNN [30]	0.4767	0.1435	0.1603	0.2753	0.1311	0.2216
	BPR [31]	0.5026	0.1576	0.1709	0.2792	0.1653	0.1674
deep learning models	NeuMF	0.4823	0.1673	0.1748	0.2984	0.1839	0.1794
	DeepFM [32]	0.5294	0.2294	0.2175	0.3011	0.1698	0.1786
hybrid models	WideDeep [33]	0.5152	0.2301	0.2314	0.3072	0.1726	0.1891
	DiffRec [34]	0.5122	0.2251	0.2312	0.2975	0.2596	0.2311
	EulerNet [35]	0.5231	0.2105	0.2278	0.3021	0.2556	0.2212
	FEARec [36]	**0.5358**	**0.2376**	0.2321	0.3011	0.2591	0.2377
	LDiffRec [37]	0.4826	0.1574	0.1741	0.2992	0.1819	0.1782
	Proposed Model (TOAR)	0.5327 ↓	0.2268 ↓	**0.2371** ↑	**0.3121** ↑	**0.2631** ↑	**0.2393** ↑

The superiority of the TOAR model in experimental results can be analyzed from the following perspectives:

(1) Overall Performance: The TOAR model demonstrates superior performance across most metrics, particularly excelling in NDCG (0.2371), MRR (0.3121), UNCP (0.2631), and NOV (0.2393). This indicates that our model not only provides accurate recommendations but also enhances the diversity and novelty of the results.

(2) Accuracy Metrics (AUC and Precision): TOAR achieves the highest AUC (0.5327) among all models, slightly outperforming FEARec (0.5358). In terms of precision, TOAR (0.2268) is competitive, second only to FEARec (0.2376). This demonstrates the model's strong ability to distinguish between relevant and irrelevant items.

(3) Ranking Quality (NDCG and MRR): TOAR outperforms all other models in both NDCG and MRR, suggesting that it not only recommends relevant items but

also ranks them more effectively. This is crucial for user satisfaction in real-world recommendation scenarios.

(4) Diversity and Novelty (UNCP and NOV): TOAR significantly outperforms other models in these metrics, with UNCP at 0.2631 and NOV at 0.2393. This highlights the model's ability to provide diverse and novel recommendations, addressing the "echo chamber" effect often seen in traditional recommendation systems.

(5) Comparison with Traditional Models: TOAR shows substantial improvements over traditional models like Pop, ItemKNN, and BPR across all metrics. For instance, compared to Pop, TOAR improves AUC by 13.6% and NDCG by 38.5%.

(6) Comparison with Deep Learning Models: While deep learning models like DeepFM show improvements in accuracy metrics, TOAR outperforms them in diversity and novelty. For example, TOAR's UNCP is 54.9% higher than DeepFM's.

(7) Comparison with Hybrid Models: TOAR demonstrates balanced performance across all metrics compared to other hybrid models. It particularly excels in diversity and novelty metrics while maintaining competitive accuracy.

In conclusion, Table 2 demonstrates that our TOAR model successfully addresses the trade-off between accuracy and diversity in recommendation systems, outperforming both traditional and state-of-the-art models across various metrics. This comprehensive performance underscores the effectiveness of our approach in enhancing user experience through more accurate, diverse, and novel recommendations.

As shown in Table 3, in the ablation experiments, the NOV metric significantly increased to 0.2897 after the removal of the knowledge clustering module, indicating that knowledge clustering plays a crucial role in maintaining the diversity of recommendation results. The knowledge clustering module likely enhances recommendation diversity by better capturing the preference characteristics of different user groups through clustering analysis of user rating behaviors. On the other hand, removing the knowledge transfer module led to an improvement in UNCP, suggesting an expansion in the coverage of recommendation results, but there was a decline in overall performance metrics such as AUC and Pre. This reflects the critical role of knowledge transfer in enhancing recommendation accuracy, possibly because knowledge transfer enables the model to learn valuable information from related domains or similar tasks, thereby exhibiting greater robustness in the face of data sparsity or cold-start problems.

Table 3. Ablation study.

Algorithm	AUC	Pc	NDCG	MRR	UNCP	NOV
proposed Model (without knowledge clustering)	0.5211	0.2227	0.2289	**0.3201**	0.2305	0.2297
proposed Model (without knowledge transfer)	0.5123	0.2155	0.2297	0.3012	0.2621	0.2311
Proposed Model (TOAR)	**0.5327** ↑	**0.2268**↑	**0.2371** ↑	0.3121 ↓	**0.2631** ↑	**0.2393** ↑

In summary, the superiority of the TOAR model can be primarily attributed to its ingenious design of the rating knowledge transfer mechanism and the dedicated diversity enhancement module. The rating knowledge transfer enables the model to better capture users' latent preference information, while the diversity enhancement module effectively prevents the homogenization of recommendation results, allowing TOAR to achieve a better balance between diversity and accuracy. These advantages enable TOAR to excel across various metrics, particularly demonstrating significant strengths in recommendation diversity and ranking accuracy.

5. Conclusions

This study explores the integration of Bidirectional Long Short-Term Memory Networks (Bi-LSTM), the Mean-Shift clustering algorithm, and the feature-sharing mechanism within recommender systems. It examines the synergistic effect of these technologies in augmenting system performance and predictive accuracy. The findings suggest that the

synerg of Bi-LSTM with the Mean-Shift algorithm proficiently captures and forecasts user behavior while preserving interest diversity. Additionally, to curb overfitting, the NeuMF model is deployed to simplify the model architecture. This hybrid methodology not only bolsters the precision and operational efficacy of the recommender system but also ushers in avenues for instantaneous, dynamic, and individualized recommendations. Our empirical evidence attests to its superiority across various application contexts. The TOAR model, as demonstrated in our experiments, adeptly processes news recommendations, concurrently sustaining accuracy and notably enriching recommendation diversity and novelty. Nonetheless, a performance downturn was observed in some marginal scenarios, potentially attributable to unique dataset characteristics or intrinsic model presumptions.

Although the experimental outcomes are promising, they are not without limitations. Presently, the scope of experimentation is limited to a news recommendation dataset. Future research endeavors could extend to a broader array of datasets to ascertain the model's generalizability. Subsequent experiments are slated to incorporate cutting-edge deep-learning methodologies to enhance model efficacy further.

Author Contributions: Conceptualization, Y.Y. and X.C.; methodology, Y.Y. and X.C.; software, Y.Y. and X.C.; validation, Y.Y., Y.Z., and H.C.O.; formal analysis, Y.Y.; investigation, Y.Y.; resources, X.C. and Y.Z.; data curation, Y.Y.; writing—original draft preparation, Y.Y.; writing—review and editing, Q.X. and H.C.O.; visualization, Y.Z.; supervision, Q.X.; project administration, Q.X.; funding acquisition, Q.X. and Y.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Innovation Training Program for College Students, grant number 202210549014.

Data Availability Statement: The data presented in this study are available at https://www.kaggle.com/datasets/everydaycodings/global-news-dataset (accessed on 27 September 2024).

Acknowledgments: The authors would like to thank the anonymous reviewers for their constructive comments and insightful suggestions.

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

References

1. Lee, Y.L.; Zhou, T.; Yang, K.; Du, Y.; Pan, L. Personalized recommender systems based on social relationships and historical behaviors. *Appl. Math. Comput.* **2023**, *437*, 127549. [CrossRef]
2. Geng, S.; He, X.; Liang, G.; Niu, B.; Liu, S.; He, Y. Accuracy-diversity optimization in personalized recommender system via trajectory reinforcement based bacterial colony optimization. *Inf. Process. Manag.* **2023**, *60*, 103205. [CrossRef]
3. Cinelli, M.; De Francisci Morales, G.; Galeazzi, A.; Quattrociocchi, W.; Starnini, M. The echo chamber effect on social media. *Proc. Natl. Acad. Sci. USA* **2021**, *118*, e2023301118. [CrossRef] [PubMed]
4. Guess, A.; Nyhan, B.; Lyons, B.; Reifler, J. Avoiding the echo chamber about echo chambers. *Kn. Found.* **2018**, *2*, 1–25.
5. Chen, L.; Yang, Y.; Wang, N.; Yang, K.; Yuan, Q. How serendipity improves user satisfaction with recommendations? a large-scale user evaluation. In Proceedings of the World Wide Web Conference, San Francisco, CA, USA, 13–17 May 2019; pp. 240–250.
6. Kotkov, D.; Wang, S.; Veijalainen, J. A survey of serendipity in recommender systems. *Knowl.-Based Syst.* **2016**, *111*, 180–192. [CrossRef]
7. Kunaver, M.; Požrl, T. Diversity in recommender systems—A survey. *Knowl.-Based Syst.* **2017**, *123*, 154–162. [CrossRef]
8. Lu, Q.; Chen, T.; Zhang, W.; Yang, D.; Yu, Y. Serendipitous personalized ranking for top-n recommendation. In Proceedings of the 2012 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology, Macau, China, 4–7 December 2012; Volume 1, pp. 258–265.
9. Zhang, Y.C.; Séaghdha, D.Ó.; Quercia, D.; Jambor, T. Auralist: Introducing serendipity into music recommendation. In Proceedings of the Fifth ACM International Conference on Web Search and Data Mining, Seattle, WA, USA, 8–12 February 2012; pp. 13–22.
10. Gartrell, M.; Paquet, U.; Koenigstein, N. Low-rank factorization of determinantal point processes. In Proceedings of the AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017; Volume 31.
11. Adamopoulos, P.; Tuzhilin, A. On unexpectedness in recommender systems: Or how to better expect the unexpected. *ACM Trans. Intell. Syst. Technol.* **2014**, *5*, 1–32. [CrossRef]
12. Schafer, J.B.; Frankowski, D.; Herlocker, J.; Sen, S. Collaborative filtering recommender systems. In *The Adaptive Web: Methods and Strategies of Web Personalization*; Springer: Cham, Switzerland, 2007; pp. 291–324.

13. Wei, Y.; Wang, X.; Li, Q.; Nie, L.; Li, Y.; Li, X.; Chua, T.S. Contrastive learning for cold-start recommendation. In Proceedings of the 29th ACM International Conference on Multimedia, Virtual, 20–24 October 2021; pp. 5382–5390.
14. Javed, U.; Shaukat, K.; Hameed, I.A.; Iqbal, F.; Alam, T.M.; Luo, S. A review of content-based and context-based recommendation systems. *Int. J. Emerg. Technol. Learn.* **2021**, *16*, 274–306. [CrossRef]
15. Du, Y.; Ranwez, S.; Sutton-Charani, N.; Ranwez, V. Is diversity optimization always suitable? Toward a better understanding of diversity within recommendation approaches. *Inf. Process. Manag.* **2021**, *58*, 102721. [CrossRef]
16. Patil, T.; Pandey, S.; Visrani, K. A review on basic deep learning technologies and applications. In *Data Science and Intelligent Applications*; Springer: Singapore, 2020; pp. 565–573.
17. Zhang, J.; Zheng, Y.; Qi, D.; Li, R.; Yi, X. DNN-based prediction model for spatio-temporal data. In Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, Burlingame, CA, USA, 31 October–3 November 2016; pp. 1–4.
18. Xie, L.; Yuille, A. Genetic CNN. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 1379–1388.
19. Yin, W.; Kann, K.; Yu, M.; Schütze, H. Comparative study of CNN and RNN for natural language processing. *arXiv* **2017**, arXiv:1702.01923.
20. Martins, G.B.; Papa, J.P.; Adeli, H. Deep learning techniques for recommender systems based on collaborative filtering. *Expert Syst.* **2020**, *37*, e12647. [CrossRef]
21. Pang, X.; Wang, Y.; Yang, S.; Liu, W.; Yu, Y. A bi-objective low-carbon economic scheduling method for cogeneration system considering carbon capture and demand response. *Expert Syst. Appl.* **2024**, *243*, 122875. [CrossRef]
22. Hao, Z.; Lu, Z.; Li, G.; Nie, F.; Wang, R.; Li, X. Ensemble clustering with attentional representation. *IEEE Trans. Knowl. Data Eng.* **2023**, *36*, 581–593. [CrossRef]
23. Xu, J.; Li, T.; Zhang, D.; Wu, J. Ensemble clustering via fusing global and local structure information. *Expert Syst. Appl.* **2024**, *237*, 121557. [CrossRef]
24. Gupta, P.; Gasse, M.; Khalil, E.; Mudigonda, P.; Lodi, A.; Bengio, Y. Hybrid models for learning to branch. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 18087–18097.
25. Rendle, S.; Krichene, W.; Zhang, L.; Anderson, J. Neural collaborative filtering vs. matrix factorization revisited. In Proceedings of the 14th ACM Conference on Recommender Systems, Virtual, 22–26 September 2020; pp. 240–248.
26. Kang, W.C.; McAuley, J. Self-attentive sequential recommendation. In Proceedings of the 2018 IEEE International Conference on Data Mining (ICDM), Singapore, 17–20 November 2018; pp. 197–206.
27. Li, F.; Yan, B.; Long, Q.; Wang, P.; Lin, W.; Xu, J.; Zheng, B. Explicit semantic cross feature learning via pre-trained graph neural networks for CTR prediction. In Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual, 11–15 July 2021; pp. 2161–2165.
28. Global News Dataset. Available online: https://www.kaggle.com/datasets/everydaycodings/global-news-dataset (accessed on 27 September 2024).
29. Qin, Y.; Zhang, S.; Zhu, X.; Zhang, J.; Zhang, C. POP algorithm: Kernel-based imputation to treat missing values in knowledge discovery from databases. *Expert Syst. Appl.* **2009**, *36*, 2794–2804. [CrossRef]
30. Vairale, V.S.; Shukla, S. Recommendation of food items for thyroid patients using content-based knn method. In *Data Science and Security 2020*; Springer: Singapore, 2021; pp. 71–77.
31. Rendle, S.; Freudenthaler, C.; Gantner, Z.; Schmidt-Thieme, L. BPR: Bayesian personalized ranking from implicit feedback. *arXiv* **2012**, arXiv:1205.2618.
32. Chen, L.; Xu, J.; Li, S.C. DeepMF: Deciphering the latent patterns in omics profiles with a deep learning method. *BMC Bioinform.* **2019**, *20*, 648. [CrossRef] [PubMed]
33. Singh, N.B.; Singh, M.M.; Sarkar, A.; Mandal, J.K. A novel wide & deep transfer learning stacked GRU framework for network intrusion detection. *J. Inf. Secur. Appl.* **2021**, *61*, 102899.
34. Li, W.; Dong, Q.; Fu, Y. List-Wise Diffusion-Based Recommender Algorithm from Implicit Feedback. In Proceedings of the 2015 2nd International Conference on Information Science and Security (ICISS), Seoul, Republic of Korea, 14–16 December 2015; pp. 1–4.
35. Tian, Z.; Bai, T.; Zhao, W.X.; Wen, J.R.; Cao, Z. EulerNet: Adaptive Feature Interaction Learning via Euler's Formula for CTR Prediction. In Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, Taipei, Taiwan, 23–27 July 2023; pp. 1376–1385.
36. Du, X.; Yuan, H.; Zhao, P.; Qu, J.; Zhuang, F.; Liu, G.; Liu, Y.; Sheng, V.S. Frequency enhanced hybrid attention network for sequential recommendation. In Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, Taipei, Taiwan, 23–27 July 2023; pp. 78–88.
37. Wang, W.; Xu, Y.; Feng, F.; Lin, X.; He, X.; Chua, T.S. Diffusion recommender model. In Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, Taipei, Taiwan, 23–27 July 2023; pp. 832–841.