


Article

Enhanced Collaborative Filtering: Combining Autoencoder and Opposite User Inference to Solve Sparsity and Gray Sheep Issues

Lamyae El Youbi El Idrissi ^{1,*}, Ismail Akharraz ², Aziza El Ouazizi ^{1,3} and Abdelaziz Ahaitouf ¹ 

¹ Engineering Sciences Laboratory, Polydisciplinary Faculty of Taza, Sidi Mohamed Ben Abdellah University, Taza 35000, Morocco; aziza.elouazizi@usmba.ac.ma (A.E.O.); abdelaziz.ahaitouf@usmba.ac.ma (A.A.)

² Mathematical and Informatics Engineering Laboratory, Faculty of Science Agadir, Ibnou Zohr University, Agadir 80000, Morocco; ismail.akharraz@usmba.ac.ma

³ Artificial Intelligence, Data Sciences and Emerging Systems Laboratory, National School of Applied Sciences, Sidi Mohamed Ben Abdellah University, Fez 30050, Morocco

* Correspondence: lamyae.elyoubielidrissi@usmba.ac.ma

Abstract: In recent years, the study of recommendation systems has become crucial, capturing the interest of scientists and academics worldwide. Music, books, movies, news, conferences, courses, and learning materials are some examples of using the recommender system. Among the various strategies employed, collaborative filtering stands out as one of the most common and effective approaches. This method identifies similar active users to make item recommendations. However, collaborative filtering has two major challenges: sparsity and gray sheep. Inspired by the remarkable success of deep learning across a multitude of application areas, we have integrated deep learning techniques into our proposed method to effectively address the aforementioned challenges. In this paper, we present a new method called Enriched_AE, focused on autoencoder, a well-regarded unsupervised deep learning technique renowned for its superior ability in data dimensionality reduction, feature extraction, and data reconstruction, with an augmented rating matrix. This matrix not only includes real users but also incorporates virtual users inferred from opposing ratings given by real users. By doing so, we aim to enhance the accuracy of predictions, thus enabling more effective recommendation generation. Through experimental analysis of the MovieLens 100K dataset, we observe that our method achieves notable reductions in both RMSE (Root Mean Squared Error) and MAE (Mean Absolute Error), underscoring its superiority over the state-of-the-art collaborative filtering models.

Keywords: collaborative filtering; deep learning; autoencoder; recommender system; sparsity; gray sheep



Citation: El Youbi El Idrissi, L.; Akharraz, I.; El Ouazizi, A.; Ahaitouf, A. Enhanced Collaborative Filtering: Combining Autoencoder and Opposite User Inference to Solve Sparsity and Gray Sheep Issues. *Computers* **2024**, *13*, 275. <https://doi.org/10.3390/computers13110275>

Academic Editor: Fernando Bobillo

Received: 14 September 2024

Revised: 12 October 2024

Accepted: 14 October 2024

Published: 23 October 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Internet technology relies on a vast network of interconnected devices, such as computers, smartphones, laptops, and Internet of Things (IoT) tools, which continuously generate and exchange massive amounts of data. Users can utilize these data for their research projects [1], as well as to find products and services. The volume and content of these data can be incredibly beneficial as they pave the way for the use of advanced analytical models, such as machine learning [2], which utilize large datasets to facilitate the decision-making process. However, this excessive volume of information can lead to a state of information overload [3], which has become one of the primary obstacles for users attempting to efficiently extract useful information from all the available data. To address this challenge and ensure efficient information processing, numerous expert systems are being developed, effectively alleviating the burden on humans. A recommender system is one of the main methods for tackling this issue [4]. Its software tools and methods offer recommendations for items that are likely to be of interest to a specific user [5].

Recommender systems have demonstrated their utility across a myriad of applications, and every year new approaches to developing the techniques are suggested [6]; the authors of [7] defined eleven distinct application domains for recommender systems: multimedia, e-group activity, document and image, networking, e-library, e-learning, e-government, e-tourism, e-business, e-commerce, and e-health grouped into three areas: entertainment, education, and service. Figure 1 illustrates these different application domains.

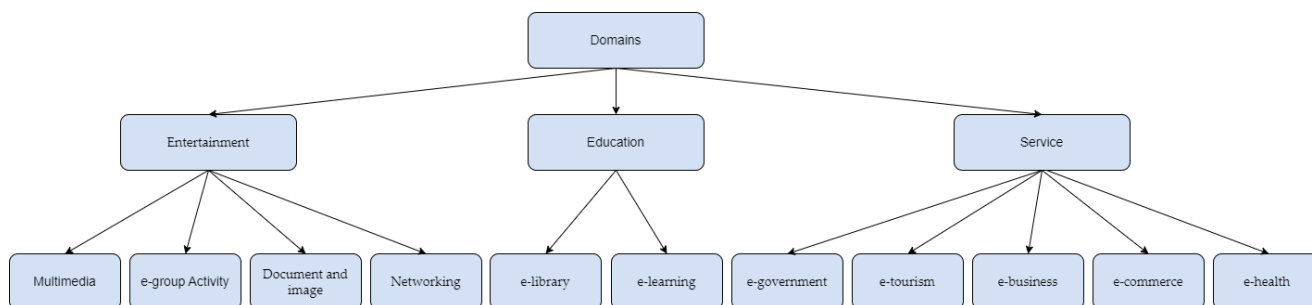


Figure 1. Classification of Recommender Systems in the Application Phase According to the Study of Taghavi et al. [7].

Recommender systems are typically categorized into three main categories [8–10]: content-based recommender system, collaborative filtering (CF) recommender system, and hybrid recommender system.

Content-based recommender system: This approach relies on comparing the features of items and users. It suggests items similar to those the user has interacted with previously, either implicitly or explicitly through ratings or clicks. Consequently, a user profile is made and utilized to find new, intriguing, and pertinent items for the user.

CF relies on leveraging either user community opinions or past user behavior to predict items that would capture a user’s interest. It gauges the similarity in preferences between two users by analyzing their respective rating histories.

CF techniques make use of a ratings matrix, exemplified in Table 1, with rows representing users and columns representing items as input. The output typically consists of predictive values indicating the extent to which a user likes or dislikes an item.

Table 1. Ratings Matrix.

	Item1	Item2	Item3	Item3	Item4	Item5
User1	2	4		2	1	
User2		3	3	2	5	
User3	5	1	3	5		4
User4	1	5			2	3

Hybrid recommendations involve combining two or more algorithms to develop an enhanced recommender system, leveraging the strengths and weaknesses of each. By combining various techniques, this approach yields improved and more accurate suggestions.

According to [11], the CF recommendations algorithm can be subdivided into two primary classes: memory-based CF and model-based CF. The memory-based algorithm is a heuristic algorithm that predicts ratings by comparing a user’s past ratings with those of others. It assesses rating similarities between users to make predictions. Conversely, model-based algorithms utilize ratings to learn a model, which is subsequently employed for making predictions [12].

Model-based CF encompasses various techniques, among which the latent factor model (LFM) stands out. It extracts the latent representations of the item and the user from high-dimensional data [13].

Matrix Factorization (MF) is a latent factor model (LFM) technique that decomposes the high-dimensional data into low-rank data and projects the latent factors into a shared

latent space [13,14]. Subsequently, by computing the inner product between the item latent and user latent vectors, it predicts the preferences of users for items in this space. Common algorithms for matrix factorization include “Singular Value Decomposition (SVD)”, “Principal Component Analysis (PCA)”, and “Non-negative Matrix Factorization (NMF)”. Despite its effectiveness, Matrix Factorization is inherently constrained by its linear framework. This limitation makes it challenging to uncover the complex and non-linear relationships that are crucial for accurately predicting user preferences [15].

Deep Learning (DL), a relatively new and rapidly advancing discipline, has made remarkable strides across various fields, particularly in sentiment analysis [16], natural language processing (NLP) [17], and pipeline leak detection [18–20]. Leveraging its powerful neural network architectures. More recently, recommendation systems have started to reap the benefits of DL, opening up exciting new possibilities for improving their effectiveness and efficiency.

One of the most significant advantages that DL brings to recommendation systems is its capability to manage and learn from non-linear data representations. Traditional recommendation techniques often find it challenging to capture the complex, non-linear relationships between users and items, especially in the presence of diverse and sparse interaction data. In contrast, DL models excel at uncovering these intricate patterns and interactions through their deep, multi-layered structures, enabling them to provide more accurate recommendations.

Furthermore, DL enhances the computational power and scalability of recommendation systems [13]. DL models are designed to process and analyze vast amounts of data efficiently, making them ideal for the large-scale datasets that are common in modern recommendation systems. This ability to handle big data allows for more detailed and comprehensive modeling of user behavior and preferences, thereby improving the relevance and quality of the recommendations provided. Autoencoders are a fundamental component in DL architectures [21]. Figure 2 summarizes the hierarchical flow to address the problem of recommender systems.

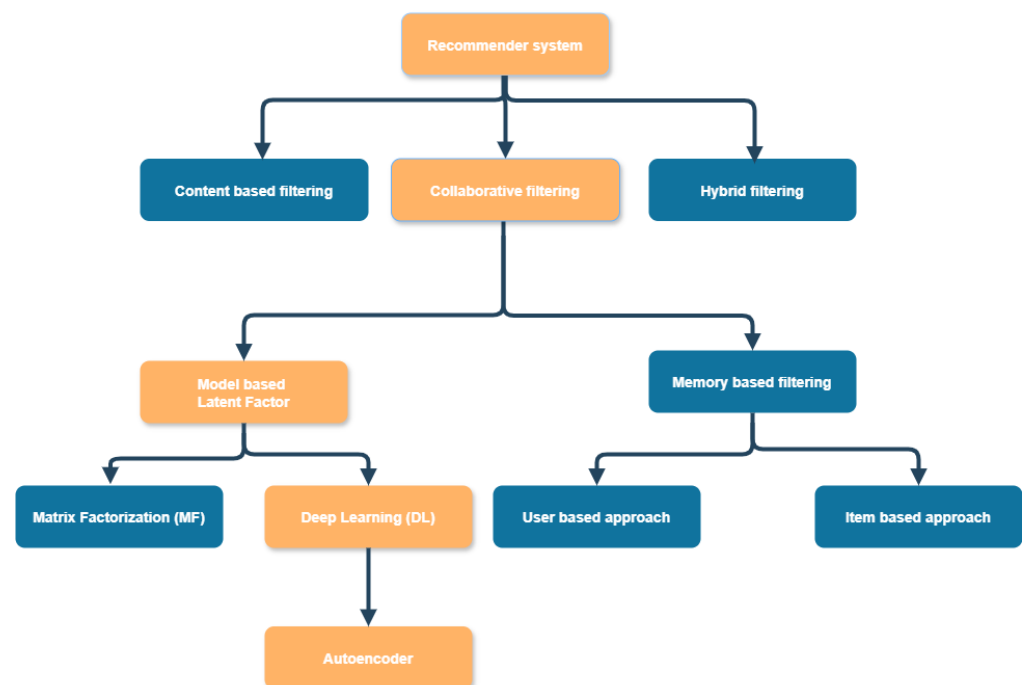


Figure 2. Hierarchical flow summary for tackling recommender system issues; the orange-colored boxes represent the hierarchical pathway of the research problem.

Despite advancements in CF methods, they continue to face significant challenges that hinder the quality of recommendations. Two notable issues are the sparsity problem and the

gray sheep problem. Sparsity arises when the dataset contains a vast amount of information but very few user ratings or interactions per item [22], making it challenging for the system to produce accurate recommendations. On the other hand, the gray sheep problem involves users whose preferences or behaviors are so unique and distinct from the majority that the system struggles to find commonalities with other users [23]. Consequently, gray sheep users might not receive effective recommendations and could even negatively influence the recommendations provided to other users [24].

In this paper, we propose a new method that combines the Autoencoder (AE) with an augmented ratings matrix in order to improve the quality of predictions and recommendations and also effectively address the challenges of sparsity and the gray sheep problem.

2. Related Works

In the field of recommendation systems, numerous solutions leveraging CF have been developed to help users discover items that align with their interests and preferences.

Fazziki et al. [25] introduced a novel CF approach specifically designed to address the gray sheep problem and enhance prediction accuracy in recommendation systems. This method involves new, “imaginary” users from existing real users in the dataset. These imaginary users are created by generating profiles with preferences that are the inverse of those of the actual users. This strategy achieves two main objectives. Firstly, it expands the pool of potential neighbors, which is particularly beneficial for users with unique or atypical behaviors (gray sheep). Secondly, it significantly increases the number of potential neighbors. The underlying principle of this approach is that if a real user X , does not share similarities with another user, Y , then an “imaginary” user with preferences opposite to X , denoted as $\neg X$, is likely to be similar to Y . This method ensures that we can find relevant neighbors and provide more accurate recommendations, even for users with uncommon preferences. To evaluate the effectiveness of this approach two databases, MovieLens and FilmTrust, are used. The experimental results demonstrated that this method significantly outperforms traditional user-based CF.

Aziz et al. [26] suggested a solution approach to the sparsity problem using various imputation methods. Specifically, it employs the mean, “min”, “max”, and “K-Nearest Neighbors Imputation (KNNI)” techniques. The process involves imputing the missing data, then calculating similarities using the cosine similarity method and evaluating the results with the “(RMSE)”. The experimental results demonstrate that the mean imputation method outperforms others, achieving a remarkable average similarity value of 0.99 and a minimal RMSE value of 0.98. These findings indicate that the “mean” imputation method effectively addresses the “sparsity problem” and enhances the quality of recommendations.

Huang et al. [27] introduced an enhanced CF personalized recommendation “(ICF)” algorithm designed to effectively address the issue of “data sparsity” by minimizing the item space. By employing the “k-means clustering” method to extract similarity information secondarily, the “ICF” algorithm more accurately acquires user similarity information, thereby enhancing the precision of recommender systems. Experiments conducted with the MovieLens and Netflix datasets demonstrate that the ICF algorithm significantly improves the accuracy and quality of recommendations.

Bathla et al. [28] suggested the “AutoTrustRec” model, which integrates “autoencoders” and “social trust” to enhance recommendation accuracy for large-scale, complex, and heterogeneous datasets. A shared correlated layer in AE is employed, integrating “user–item ratings” and user–user trust values. Comprehensive experimental analysis of three public databases (Epinions, FilmTrust, and Ciao) demonstrates that this method significantly improves both RMSE and MAE.

Pan et al. [29] introduced a model for recommendation focused on deep autoencoders for learning social representations. This model was engineered to learn features at both low and high levels from “social” data using “matrix factorization” and a “multilayer neural network”. Specifically, a “Sparse Stacked Denoising Autoencoder” model was proposed to address the challenges of imbalanced and sparse data in social networks.

Additionally, a unified recommender algorithm has been developed by integrating matrix factorization with deep representations. The authors' experiments with the Epinions and Ciao datasets demonstrate that their approach significantly improves the performance of recommendation systems, especially for users with sparse data.

Rajput et al. [30] revealed a new method for "multi-criteria" recommender systems by employing DL algorithms to address the problem of "data sparsity". They used deep autoencoders to identify non-linear, complex, and latent relationships in users' multi-criteria preferences. This was subsequently followed by a "matrix factorization (MF)" method to enhance recommendation precision. This model is tested on the "multi-criteria" database from "Yahoo!Movies". The results show that this approach outperforms current "state-of-the-art" recommendation models by providing more personalized and accurate recommendations, and it decreases "data sparsity" in the multi-criteria database by up to 11%.

Hiriyannaiah et al. [31] addressed "data sparsity" by employing a novel neural CF-based "DeepLSGR" model. This bidirectional model utilizes user-submitted "text reviews" to predict ratings and subsequently generate recommendations. The model features hidden layers with "Long Short-Term Memory (LSTM)", and "Gated Recurrent Units (GRU)". In evaluations using the "Amazon Fine Food Reviews" and OpinRank datasets, DeepLSGR reached a recall of 61%, an accuracy of 97%, and an RMSE of 0.87. Compared to prior methods, DeepLSGR provides superior recommendations.

Liu and Zhao [32] created a recommender system utilizing "sentiment analysis" and "matrix factorization" ("SAMF") to tackle "data sparsity" and "credibility" issues. This system integrates topic modeling, matrix factorization, and DL techniques to improve the accuracy of recommendations. First, "Latent Dirichlet Allocation (LDA)" is employed to create a user topic distribution from both user and item reviews. The user and item feature matrices are then constructed based on the topic probabilities. The "user-item preference matrix" is created by combining the user feature matrix with the item feature matrix. Third, the "user-item preference matrix" is integrated with the "original rating matrix" to produce the "user-item rating matrix". Fourth, "Bidirectional Encoder Representations from Transformers (BERT)" is utilized to determine the sentiment information present in the reviews. This sentiment information is then combined with the "user-item rating matrix", enabling modifications and updates to the matrix. The updated "user-item rating matrix" is employed to predict ratings and generate Top N recommendations. Experiments using Amazon data source show that the suggested "SAMF" outperforms other classical algorithms in recommendation performance.

Choi et al. [33] presented a method to enhance "data sparsity" and improve prediction accuracy by utilizing user responses alongside item features. Building on the principles of content-based filtering, the authors extracted category rates from the "user-item matrix" based on user preferences and organized these rates into vectors. They presented a method for filtering the "user-item matrix" using these vectors and regenerating the input matrix for CF. The results of the prediction of this method are compared with those of traditional CF using MAE and RMSE metrics. The results of the suggested method show that using the "regenerated matrix" as input for CF enables the construction of a denser matrix with superior predictive accuracy compared to conventional methods.

Muhammad et al. [34] proposed an improved CF recommender system by integrating SVD, ensemble clustering, and context-aware algorithms. By combining these methods, the system aims to produce accurate predictions and address issues of scalability and sparsity. Experimental results indicate that the ensemble-based "clustering" with "SVD" and context-aware technique performs better than the ensemble-based clustering with KNN and context-aware approach.

Barathy and Chitra [35] suggested an incorporation-based recommendation method to tackle the problem of sparsity in "SVD-based strategies". The process begins by identifying similar users and items, followed by generating data based on co-rated values. These data are then integrated into the "SVD" framework. This method is implemented on

the MovieLens 100K dataset, and experimental results indicate that the prediction of the proposed method outperforms the existing system.

Srivastava et al. [36] focused on “gray sheep” behavior within the realm of e-commerce recommendations. Although recommender systems are valuable for creating personalized recommendations for individual users, there is a particular subset of users who can undermine their effectiveness. These users, known as “gray sheep”, exhibit very little similarity to the majority of users, making the system’s recommendations less effective. The presence of “gray sheep” users diminishes the quality of recommendations for the majority of other users. Therefore, identifying and removing these gray sheep users from the recommender system is crucial. The authors proposed and empirically validated through four experiments that “personality” based user models can identify “gray sheep” users. Removing these users from the system can significantly enhance the overall effectiveness of collaborative filtering-based recommender systems.

Kaur and Rani [37] used various personality parameters to evaluate individual user tastes, and based on these parameters, gray sheep users are identified and excluded from the recommendations to improve accuracy. This paper uses only one personality model, the Big Five. The system uses three machine learning algorithms: “boosted decision trees, two neural networks, logistic regression, and decision forest” for chocolate bar recommendations. Evaluation of these algorithms based on the flavors_of_cacao dataset shows that the accuracy of the decision forest is superior to that of the other machine learning algorithms.

Nguyen et al. [38] introduced an enhancement to the “KNN”-based CF method for the recommender system by taking user cognition similarity into account. This enhancement aims to refine the accuracy of user grouping and deliver more pertinent recommendations for users. Experimental results highlight the superiority of the proposed model, which surpasses benchmark methods across multiple evaluation metrics, such as “MAE”, “RMSE”, “Mean Average Precision (MAP)”, and “Normalized Discounted Cumulative Gain (NDCG)”.

A. Fazziki et al. [39] presented a novel model-based CF that combines SVD and Association Rule methods with an enriched rating matrix. This matrix includes not only the actual users but also virtual users created from the opposite ratings given by real users. By introducing these virtual users, this method increases the density of similar user profiles, facilitating more accurate and useful recommendations. The comparative study demonstrated that this approach significantly outperforms traditional CF methods in terms of accuracy.

The autoencoder DL model was utilized by El Youbi El Idrissi et al. [40] to develop an e-learning recommendation system through CF. This model excels in reducing data, extracting features, and reconstructing data. Employing the dataset established by Kulkarni et al. [41], the AE demonstrated superior accuracy and outperformed traditional methods like “KNN”, “SVD”, “Non-Negative Matrix Factorization (NMF)”, and “SVD++”. It achieved lower “RMSE” and “MAE”, indicating more precise student preference predictions.

A brief summary of some of the above-discussed methodologies with the current research is provided in Table 2.

Traditional CF techniques primarily rely on user ratings to predict users’ preferences. However, a significant limitation of this approach is the sparsity of the “user-item matrix”, where many items are rated by only a few users, and the problem of gray sheep users those whose preferences do not align closely with any user group, making accurate recommendations difficult.

Few studies tackle the problem of grey sheep, which refers to users with unique or uncommon tastes. These users are frequently seen as a burden on the system because they are believed to negatively affect model accuracy. Consequently, these users are removed from the system and placed in a separate group in the majority of studies [36,42,43]. In this paper, we will keep the “grey sheep” users in the same system.

Table 2. Overview of Existing Methods in Relation to the Current Research.

Reference	Machine Learning Method	Approach	Metric	Data Source	Addressing Sparsity	Addressing Gray Sheep
[25]	User-based CF	CF	- MAE - RMSE	MovieLens 100K	Yes	Yes
[26]	KNNI	CF	- RMSE	MovieLens 100K	Yes	No
[27]	k-means	CF	- Sorting Accuracy - Precision - Recall - Novelty - Diversity	- MovieLens 100K - Netflix	Yes	No
[28]	AE	CF	- MAE - RMSE	- Epinions - FilmTrust - Ciao	Yes	No
[31]	- LSTM - GRU	CF	- MAE - RMSE - Accuracy	- Amazon Fine Food Reviews - OpinRank	Yes	No
[32]	- LDA - BERT	CF	- MAE - F1-Score	- Amazon food - Amazon Clothing	Yes	No
[37]	- Boosted decision tree - logistic regression - decision forest - two neural networks	CF	- Accuracy - Precision - F1-Score - Recall	- flavors_of_cacao	No	Yes
[38]	- KNN	CF	- MAE - RMSE - MAP - NDCG	- MovieLens 100K - MovieLens-1 M	No	No
[39]	- Association Rule (AR) - SVD - Funk SVD (FSVD)	CF	- MAE - RMSE	MovieLens 100K	Yes	Yes
[40]	- AE - SVD - KNN - SVD++ - NMF	CF	- MAE - RMSE	dataset created by Kulkarni et al. [41]	Yes	No
Present Approach	AE	CF	- MAE - RMSE	MovieLens 100K	Yes	Yes

Recently, numerous studies have been carried out to address the issue of data sparsity [30,31,40]. These studies not only aim to enhance existing CF methods but also explore various methods incorporating DL techniques [44–47]. In this work, we will use an AE for CF tasks.

Our research explores a model-based approach using DL techniques. We integrate an AE to further advance and extend the concept of Rating Matrix Enrichment, which introduces a novel phase in the CF process that aims to enhance the traditional CF approach by augmenting the “user–item matrix”. This novel phase was used in [25] for a memory-based approach.

3. Research Aim

This study focuses on the following key objectives:

- Propose a new model based on an AE and an augmented rating matrix to improve CF performance.

- Evaluate the model's accuracy using established performance metrics, ensuring its robustness across various evaluation criteria.
- Demonstrate the effectiveness of the proposed method through experiments on the publicly available MovieLens 100K dataset.
- Address critical challenges in collaborative filtering, such as "data sparsity" and the "gray sheep" problem.

The rest of this paper is structured as follows: Section 2 includes related works. Section 3 outlines the research objectives. Section 4 details the methodology employed in our research. Section 5 covers the evaluation metrics applied, and methods for comparison utilized. Section 6 describes the experiments conducted and the results obtained. Section 7 provides the discussion. Finally, the conclusions are discussed in Section 8.

4. Methodology

This Section describes the preliminary concepts and AE method used in this work.

The AE operates by filling in missing values within the input sparse matrix [48]. This process provides two main features: (a) learning users' behavior, and (b) predicting the probability of liking specific items (movies).

The learning process relies on the interactions between users and items, offering a representation of how each student engages with the available items. The predicting feature estimates the probability of a user interacting with new items based on their past behavior.

4.1. Problem Definition

The main challenge of CF lies in effectively predicting user preferences for articles, especially in the presence of data sparsity and "gray sheep" users. Data sparsity arises when interactions between users and items are limited, making it difficult to gather sufficient information to make accurate recommendations. At the same time, "gray sheep" users pose an additional problem due to their highly varied and atypical interests, which differ considerably from those of other users, making it challenging for traditional recommendation systems to provide relevant suggestions in scenarios where available data are limited and user preferences are highly diversified.

In this study, we address these challenges by combining an AE with an enriched method, which operates on the principle that users should have similar interests. If the interests of the user are entirely opposed to those of the user, an imaginary user would have interests similar to the user's. This enriched approach aims to enhance the density of the rating matrix by adding rows that represent these opposing user preferences.

We concentrate on the task of predicting ratings in recommendation systems, specifically aiming to predict ratings for unknown user-movie pairs. This is widely adopted in recommender systems, where the accuracy of predictions is an indicator of the system's quality [29].

We also assessed the proposed method based on the accuracy of rating predictions. For this purpose, we used both "MAE" and "RMSE" simultaneously, as these metrics are commonly employed in recommender systems [49]. Additionally, we compared the performance of the model with state-of-the-art methods using the MovieLens 100K dataset and these same metrics. Our analysis showed that our model outperforms the state-of-the-art methods and enhances prediction accuracy.

4.2. Overview of Autoencoders

The AE represented in the study of Geng et al. [50] is a DL architecture designed for unsupervised learning. The primary characteristic of an AE is its ability to learn and represent input information. The model aims to reduce the difference between the input and output data by reconstructing the representation [50]. It utilizes the backpropagation algorithm and optimization methods such as gradient descent to learn the mapping between the input data and its reconstructed output data [51].

The architecture of an AE consists of two main components: the encoding part and the decoding part. The encoding section includes an input layer and a hidden layer, which are used to extract meaningful latent representations from a sparse input vector, obtained from input data, into a reduced-dimensional space. The decoding section, which includes a hidden layer and an output layer, is responsible for reconstructing the original input data from the reduced latent representations. Figure 3 depicts the principle.

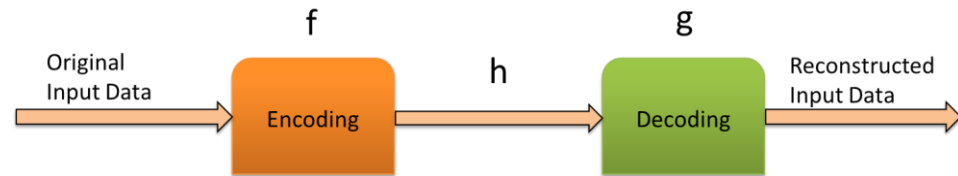


Figure 3. Structure of a basic AE.

The process of encoding and decoding can be formulated mathematically. Let us denote the input vector by x , the encoding process by $f(x)$, and the decoding process by $g(h)$, where h is the encoded representation.

- Encoding

The encoding process maps the input vector to a hidden representation through an encoding function:

$$h = f(x) = \sigma(w_{1x} + b_1) \quad (1)$$

where

- w_1 is the encoding layer's weight matrix.
- b_1 is the encoding layer's bias vector.
- σ is an activation function.

- Decoding

The decoding process maps the hidden representation h back to the reconstructed input x' through a decoding function $g(h)$:

$$x' = g(h) = \sigma(w_{2h} + b_2) \quad (2)$$

where

- w_2 is the decoding layer's weight matrix.
- b_2 is the decoding layer's bias vector.

The objective of training an AE is to minimize the reconstruction error, which measures the difference between the input x and the reconstructed input x' . A common loss function for this is the "Mean Squared Error (MSE)":

$$L(x, x') = \|x - x'\|^2 \quad (3)$$

AEs have found extensive use due to their superior performance in feature extraction, data reconstruction, and data dimensionality reduction [52].

- Configurable Parameters

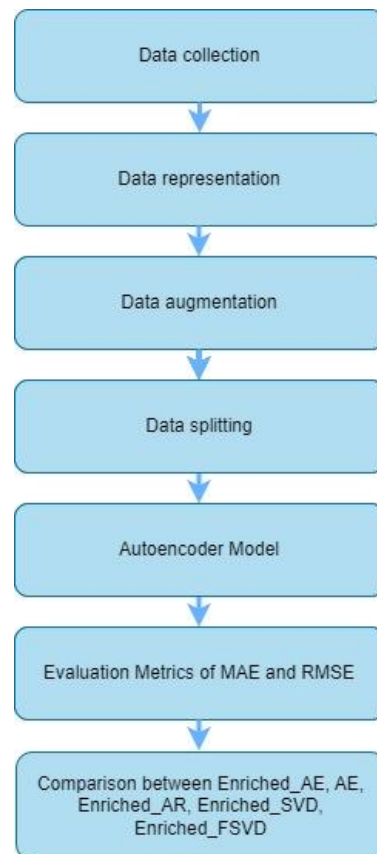
After studying several papers on autoencoders in the literature, it was discovered that every AE employs a distinct set of parameters. Choosing between them is challenging because nearly every author asserts that their algorithm or parameterization method is superior in some way. Consequently, the decision was made to use trial and error, experimenting with various criteria until the model yielded the most accurate results. Table 3 shows the hyperparameters employed in the model.

Table 3. The configuration of hyperparameters used in the experiments.

Activation	Relu
Batch size	64
Loss Function	MSE
Optimizer	Adam with a learning rate of 0.0001

4.3. Architecture of the Study

This Section presents the study plan for evaluating the model used in the later phases of our research. Figure 4 illustrates the detailed actions implemented throughout this study.

**Figure 4.** Study Workflow Diagram.

In step 1, we used the freely accessible benchmark dataset MovieLens 100K (<https://grouplens.org/datasets/movielens/100k/>) (accessed on 7 July 2024) [53], curated by the GroupLens research team. The dataset includes a substantial amount of movie rating data. It contains 100,000 ratings made by 943 users for 1682 movies. Each rating is an integer ranging from 1 to 5, with 1 indicating “bad” and 5 indicating “excellent”. Table 4 provides a comprehensive overview of the dataset details.

Table 4. Details regarding the experimental dataset.

Dataset	Users	Items	Ratings	Sparsity	Evaluation Scale
MovieLens 100K	943	1682	100,000	6.30%	1–5

Figure 5 shows the distribution of the number of ratings made by users (A) and the frequency of each rating from 1 to 5 (B).

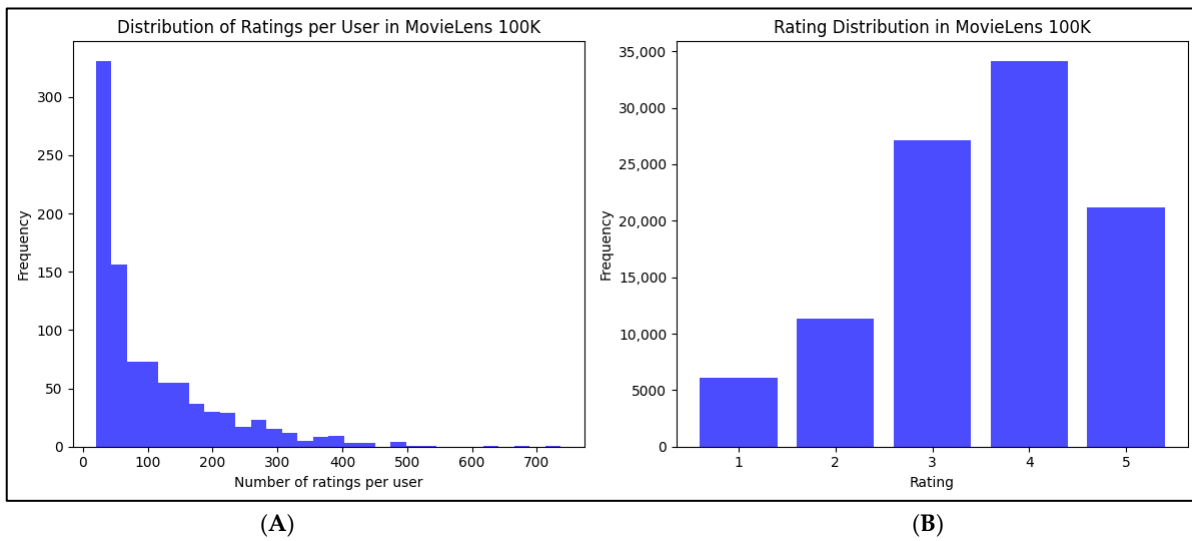


Figure 5. User’s rating distribution of the MovieLens 100K dataset: (A) Distribution of the number of ratings given by each user; (B) Distribution of different rating values.

In step 2, data representation revolves around capturing user–item interactions with a “user–item matrix” where rows represent users and columns represent items (movies). Each cell in the matrix indicates a user’s interaction (ratings). In step 3, we introduced an innovative phase called “Ratings Matrix Enrichment”, which builds upon the foundational principle laid by previous work in CF [25], aimed at expanding the traditional ratings matrix by incorporating rows that represent imaginary users with opposing preferences to real users. These imaginary users are constructed by calculating the inverse or opposite preference for each item that real users have evaluated using the formula:

$$\bar{r}_{\bar{u}i} = 2 \times \text{median} - r_{ui} \quad (4)$$

where

$\bar{r}_{\bar{u}i}$: The rating given by imaginary user \bar{u} for item i .

r_{ui} : The rating assigned by user u for item i .

median: The middle value on a specified numerical scale.

For example, using a 5-point rating scale where 1 is the lowest and 5 is the highest, the median is the middle value, which is 3, if user u_1 rates item i as 4, then the inferred rating for the opposing user \bar{u}_1 would be 2 for the same item i . Table 5 provides an example of a user with opposite preferences on a 5-point scale.

Table 5. An example of a user having opposing preferences on a scale of one to five.

	Item1	Item2	Item3	Item3	Item4
u_1	5	4	3	2	1
\bar{u}_1	1	2	3	4	5

We then randomly partitioned the dataset into 80% for training and 20% for testing in step 4. The training set was used to train the model. The test set was reserved for the ultimate assessment of the model’s effectiveness on new, unseen data. Step 5 involved implementing the AE model. Ultimately, in step 6, we employed the MAE and RMSE metrics to assess the Enriched_AE, AE (without Augmentation rating matrix), Enriched_AR, Enriched_SVD, and Enriched_FSVD approaches.

5. Experiment

This Section details the extensive tests performed on the dataset MovieLens 100K to clearly prove the efficacy of our suggested method.

5.1. Implementation Details

The proposed model was implemented in Python 3.10.12 64-bit environment on Google Colab, leveraging Python machine learning (ML) libraries. Specifically, for implementing the AE model, TensorFlow 2.15.0, and Keras 2.15.0 deep learning frameworks were utilized. Additionally, to generate various visualizations, the Matplotlib Python library was employed.

5.2. Evaluation Protocol

In this article, the RMSE and the MAE are two metrics applied to calculate the accuracy of recommendations. RMSE and MAE are standard measures for calculating the accuracy of model predictions. In the recommender system, the MAE is frequently used to calculate the average of all absolute value deviations between the actual and predicted ratings. Lower values of MAE and RMSE indicate a closer match between predicted and actual ratings, signifying better performance of the recommendation algorithm. The following formula in Equation (5) is used to calculate the MAE:

$$\text{MAE} = \frac{1}{N} \sum_{j=1}^n |\text{PredictedValue}_j - \text{TrueValue}_j| \quad (5)$$

The RMSE is employed to evaluate multiple RS by calculating the average of all absolute value deviations between the predicted and true ratings. The following equation can be used to calculate the RMSE:

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{j=1}^n (\text{PredictedValue}_j - \text{TrueValue}_j)^2} \quad (6)$$

where N , PredictedValue , and TrueValue determined, respectively, the number of ratings, the prediction ratings, and the true ratings.

5.3. Baseline Models

To evaluate the suggested model's efficacy, we compared it against the "state-of-the-art" models using the MAE and RMSE as evaluation metrics.

Enriched_AR model employs SVD with an augmented rating matrix, which incorporates virtual users, derived from analyzing the opposing ratings given by real users, effectively addressing the issue of data sparsity and gray sheep [39].

The Enriched_SVD model utilizes AR methods with an enriched rating matrix, which extends beyond actual user data by incorporating virtual users based on opposing ratings provided by real users, successfully mitigating the challenge of data sparsity and gray sheep [39].

Enriched_FSVD model leverages FSVD with an expanded rating real, which includes ratings from actual users and also incorporates virtual users, who are derived from the contrasting ratings given by real users, effectively tackling the challenge of data sparsity and gray sheep [39].

6. Results

The Enriched_AE, a DL-based CF model, was developed using the TensorFlow framework. The model's performance was evaluated using three distinct batch size configurations, eight activation functions configurations, and three Optimizer algorithms. As shown in Table 6, the optimal performance was observed with a batch size of 64.

Optimal performance was demonstrated with the Relu activation function in Table 7. Table 8 shows that optimal performance was achieved using the Adam optimizer.

Figure 6 illustrates the model's performance that combines an AE with an augmented rating matrix on the MovieLens 100K dataset. The loss and validation loss show a consistent decrease throughout each epoch until they eventually stabilize.

Table 6. Impact of Batch Size on Model Performance.

Value of Batch Size	MAE	RMSE
32	0.3863	0.6151
64	0.3319	0.5440
128	0.4855	0.7523

Table 7. Effect of Activation Functions on Model Performance.

Activation Function	MAE	RMSE
Tanh	0.3059	0.6747
“Rectified Linear Unit (Relu)”	0.1344	0.4762
Sigmoid	0.2794	0.7002
“Scaled Exponential Linear Unit (Selu)”	0.3288	0.5734
“Exponential Linear Unit (Elu)”	0.2963	0.5064
Swish (or Silu)	0.2368	0.4682
Mish	0.2513	0.4679
Growing Cosine Unit (GCU)	0.3280	0.7357

Table 8. Influence of Optimizer Algorithm on Model Performance.

Optimizer Algorithm	MAE	RMSE
RMSprop	0.1944	0.6083
SGD	0.3293	0.8277
Adam	0.1344	0.4762

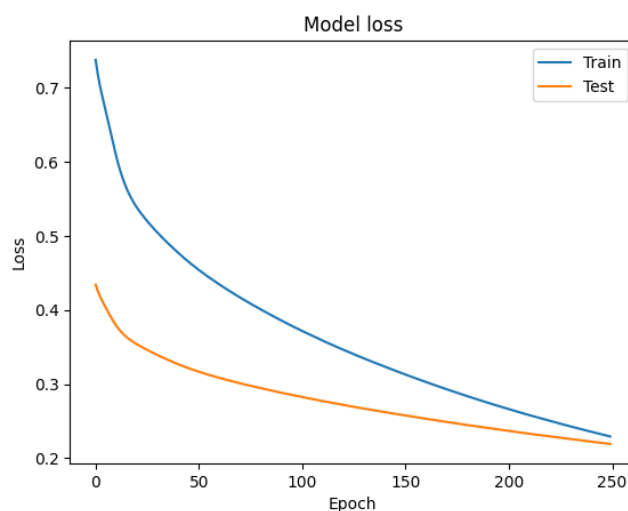
**Figure 6.** Model loss Enriched_AE.

Figure 7 illustrates the AE performance without an augmented rating matrix using the MovieLens 100K dataset.

Table 9 and Figures 8 and 9 present a comparison of MAE and RMSE results among the developed Enriched_AE model, AE model, Enriched_AR, Enriched_SVD, and Enriched_FSVD. Upon examining these findings, it is evident that Enriched_AE outperforms AE, Enriched_AR, Enriched_SVD, and Enriched_FSVD. Specifically, the Enriched_AE model achieved the lowest RMSE value at 0.4762, while AE showed an RMSE of 0.7315, Enriched_SVD exhibited an RMSE of 0.7273, Enriched_FSVD got 0.8869, and Enriched_AR resulted in 1.522, indicating the highest error rate. The MAEs for our developed Enriched_AE, AE model, Enriched_AR, Enriched_SVD, and Enriched_FSVD are 0.1344, 0.2608, 1.0647, 0.4408, and 0.6956, respectively. Overall, the Enriched_AE model produced the most favorable outcomes.

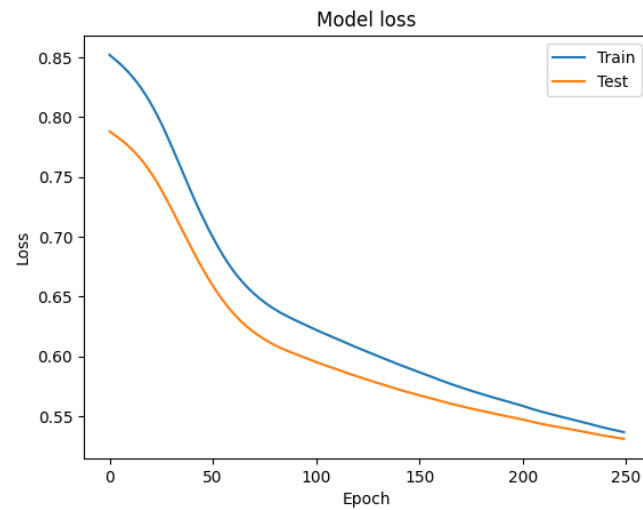


Figure 7. Model loss AE.

Table 9. Comparison of MAE and RMSE metrics across the Enriched_AE with benchmark models.

Models	RMSE	MAE
Enriched_AR	1.522	0.1344
Enriched_SVD	0.7273	0.2608
Enriched_FSVD	0.8869	1.0647
AE	0.7315	0.2608
Enriched_AE	0.4762	0.1344

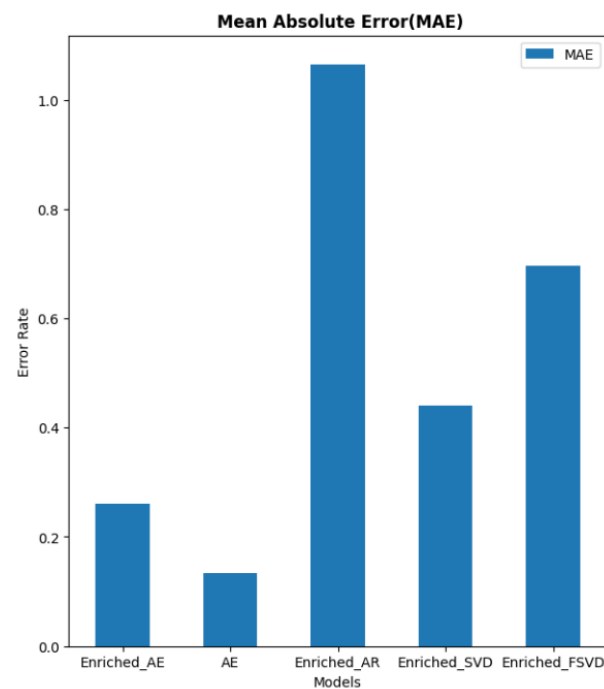


Figure 8. Comparison between the proposed method and benchmark methods based on MAE.

Figure 10 presents a detailed comparison between our proposed approach, Enriched_AE, and the traditional method, AE.

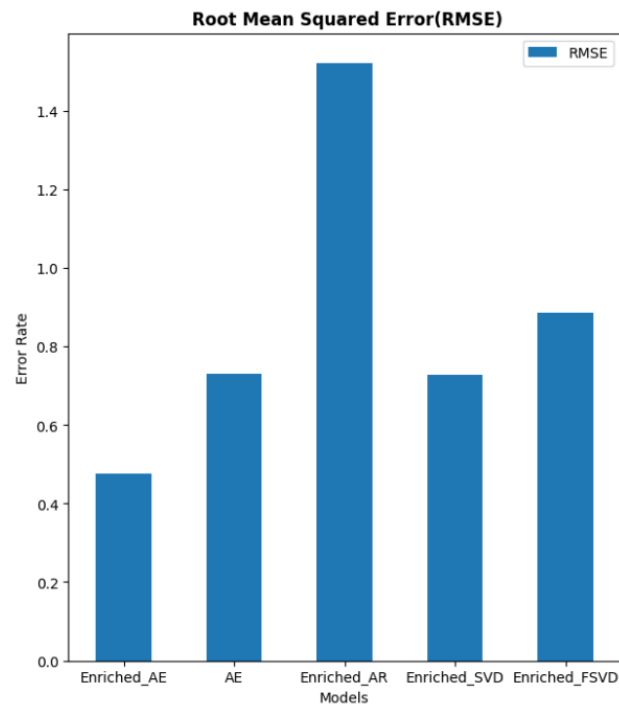


Figure 9. Comparison of the proposed method with benchmark approaches focusing on RMSE.

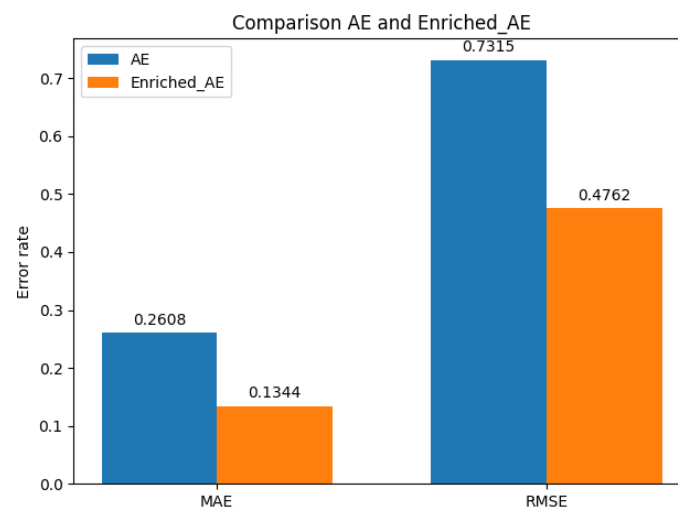


Figure 10. Comparison of AE and Enriched_AE.

7. Discussion

The goal of this work is to enhance the performance of CF recommendation systems by combining AE with an enriched rating matrix. This enriched matrix not only includes actual user ratings but also incorporates virtual users, which are inferred from the opposing ratings provided by real users. By doing so, the approach aims to increase the density of similar users within the matrix, thereby facilitating more accurate and useful recommendations. Experimental results indicate that this novel method achieves the lowest error values (MAE and RMSE) values compared to state-of-the-art approaches such as AE, Enriched_SVD, Enriched_FSVD, and Enriched_AR. This shows that the Enriched_AE model predicts movie ratings more accurately than these other techniques. The observed improvement can be attributed to the following reasons:

- Autoencoders can capture complex non-linear relationships in the data.
- The augmentation rating matrix through the incorporation of virtual users derived from the opposing ratings given by real users provides more data to the AE, allowing

it to better understand the complexities of user–item interactions and thus make more accurate predictions.

8. Conclusions

In this work, a novel approach called Enriched_AE is introduced to solve data sparsity and gray sheep problems present in the CF technique. The method combines an AE with an enriched matrix concept that includes fictional users with evaluations opposing those of real users. This enriched approach increased the density of the rating matrix by incorporating users with differing interests and preferences. Through a comparative analysis with state-of-the-art methods using the MovieLens 100K dataset, our approach consistently demonstrated superior performance, leading to enhanced prediction accuracy. These results conclusively validate that combining AE with the enriched method offers better outcomes compared to state-of-the-art approaches.

The study underscores the importance of innovation in recommendation systems, highlighting that efficiency and the strategic integration of new and traditional methodologies are pivotal in advancing research in this field.

We aim to apply the enriched approach to other domains including healthcare and education, where personalization and precise predictions are crucial. Additionally, we plan to explore further DL techniques with this approach to improve the accuracy of recommendation systems and uncover innovative applications across different fields.

Author Contributions: Conceptualization, L.E.Y.E.I. and A.A.; Methodology, L.E.Y.E.I., I.A. and A.E.O.; Software, L.E.Y.E.I.; Validation, L.E.Y.E.I., I.A., A.E.O. and A.A.; Formal analysis, I.A. and A.A.; Investigation, L.E.Y.E.I., I.A., A.E.O. and A.A.; Writing—original draft, L.E.Y.E.I.; Writing—review & editing, I.A., A.E.O. and A.A.; Visualization, L.E.Y.E.I. and A.A.; Supervision, I.A., A.E.O. and A.A.; Project administration, I.A. and A.E.O. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Dataset used in this study is publicly available and can be accessed at the following link: (<https://grouplens.org/datasets/movielens/100k/>) (accessed on 7 July 2024).

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

References

1. Benfield, J.A.; Szlemko, W.J. Internet-based data collection: Promises and realities. *J. Res. Pract.* **2006**, *2*, D1.
2. Kent, M.; Huynh, N.K.; Schiavon, S.; Selkowitz, S. Using support vector machine to detect desk illuminance sensor blockage for closed-loop daylight harvesting. *Energy Build.* **2022**, *274*, 112443. [[CrossRef](#)]
3. Zhang, S.; Bai, Z.; Li, P.; Chang, Y. Multi-Graph Convolutional Network for Fine-Grained and Personalized POI Recommendation. *Electronics* **2022**, *11*, 2966. [[CrossRef](#)]
4. Khusro, S.; Ali, Z.; Ullah, I. Recommender Systems: Issues, Challenges, and Research Opportunities. In *Information Science and Applications (ICISA) 2016*; Kim, K.J., Joukov, N., Eds.; Lecture Notes in Electrical Engineering; Springer: Singapore, 2016; Volume 376, pp. 1179–1189.
5. Ricci, F.; Rokach, L.; Shapira, B. Recommender Systems: Techniques, Applications, and Challenges. In *Recommender Systems Handbook*; Ricci, F., Rokach, L., Shapira, B., Eds.; Springer: New York, NY, USA, 2022; pp. 1–35. [[CrossRef](#)]
6. Pawlicka, A.; Pawlicki, M.; Kozik, R.; Choraś, R.S. A Systematic Review of Recommender Systems and Their Applications in Cybersecurity. *Sensors* **2021**, *21*, 5248. [[CrossRef](#)] [[PubMed](#)]
7. Taghavi, M.; Bentahar, J.; Bakhtiyari, K.; Hanachi, C. New Insights Towards Developing Recommender Systems. *Comput. J.* **2018**, *61*, 319–348. [[CrossRef](#)]
8. Ricci, F.; Rokach, L.; Shapira, B. *Introduction to Recommender Systems Handbook*; Springer: Boston, MA, USA, 2010.
9. Roy, D.; Dutta, M. A systematic review and research perspective on recommender systems. *J. Big Data* **2022**, *9*, 59. [[CrossRef](#)]
10. Chalkiadakis, G.; Ziogas, I.; Koutsmanis, M.; Streviniotis, E.; Panagiotakis, C.; Papadakis, H. A Novel Hybrid Recommender System for the Tourism Domain. *Algorithms* **2023**, *16*, 215. [[CrossRef](#)]
11. Altulyan, M.; Yao, L.; Wang, X.; Huang, C.; Kanhere, S.S.; Sheng, Q.Z. A Survey on Recommender Systems for Internet of Things: Techniques, Applications and Future Directions. *Comput. J.* **2022**, *65*, 2098–2132. [[CrossRef](#)]
12. Yu, M.; Quan, T.; Peng, Q.; Yu, X.; Liu, L. A model-based collaborate filtering algorithm based on stacked AutoEncoder. *Neural Comput. Appl.* **2022**, *34*, 2503–2511. [[CrossRef](#)]

13. Tegene, A.; Liu, Q.; Gan, Y.; Dai, T.; Leka, H.; Ayenew, M. Deep Learning and Embedding Based Latent Factor Model for Collaborative Recommender Systems. *Appl. Sci.* **2023**, *13*, 726. [[CrossRef](#)]
14. Khaledian, N.; Mardukhi, F. CFMT: A collaborative filtering approach based on the nonnegative matrix factorization technique and trust relationships. *J. Ambient Intell. Humaniz. Comput.* **2022**, *13*, 2667–2683. [[CrossRef](#)]
15. Ferreira, D.; Silva, S.; Abelha, A.; Machado, J. Recommendation System Using Autoencoders. *Appl. Sci.* **2020**, *10*, 5510. [[CrossRef](#)]
16. Mohbey, K.K.; Meena, G.; Kumar, S.; Lokesh, K. A CNN-LSTM-Based Hybrid Deep Learning Approach for Sentiment Analysis on Monkeypox Tweets. *New Gener. Comput.* **2024**, *42*, 89–107. [[CrossRef](#)]
17. Taye, M.M. Understanding of Machine Learning with Deep Learning: Architectures, Workflow, Applications and Future Directions. *Computers* **2023**, *12*, 91. [[CrossRef](#)]
18. Siddique, M.F.; Ahmad, Z.; Ullah, N.; Kim, J. A Hybrid Deep Learning Approach: Integrating Short-Time Fourier Transform and Continuous Wavelet Transform for Improved Pipeline Leak Detection. *Sensors* **2023**, *23*, 8079. [[CrossRef](#)]
19. Siddique, M.F.; Ahmad, Z.; Kim, J.-M. Pipeline leak diagnosis based on leak-augmented scalograms and deep learning. *Eng. Appl. Comput. Fluid Mech.* **2023**, *17*, 2225577. [[CrossRef](#)]
20. Siddique, M.F.; Ahmad, Z.; Ullah, N.; Ullah, S.; Kim, J.-M. Pipeline Leak Detection: A Comprehensive Deep Learning Model Using CWT Image Analysis and an Optimized DBN-GA-LSSVM Framework. *Sensors* **2024**, *24*, 4009. [[CrossRef](#)]
21. Haghighi, P.S.; Seton, O.; Nasraoui, O. An Explainable Autoencoder for Collaborative Filtering Recommendation. *arXiv* **2020**, arXiv:2001.04344. [[CrossRef](#)]
22. Li, Y.; Liu, K.; Satapathy, R.; Wang, S.; Cambria, E. Recent Developments in Recommender Systems: A Survey [Review Article]. *IEEE Comput. Intell. Mag.* **2024**, *19*, 78–95. [[CrossRef](#)]
23. Abdulrahman, R.; Viktor, H. Personalised Recommendation Systems and the Impact of COVID-19: Perspectives, Opportunities and Challenges. In Proceedings of the 12th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management, Virtual Event, 2–4 November 2020; SCITEPRESS-Science and Technology Publications: Budapest, Hungary, 2020; pp. 295–301. [[CrossRef](#)]
24. Passi, R.; Jain, S.; Singh, P.K. Hybrid Approach for Recommendation System. In Proceedings of the 2nd International Conference on Data Engineering and Communication Technology, Pune, India, 15–16 December 2017; Kulkarni, A.J., Satapathy, S.C., Kang, T., Kashan, A.H., Eds.; Advances in Intelligent Systems and Computing. Springer: Singapore, 2019; Volume 828, pp. 117–128. [[CrossRef](#)]
25. Fazziki, A.E.; Alami, Y.E.M.E.; Elhassouni, J.; Aissaoui, O.E.; Benbrahim, M. Employing opposite ratings users in a new approach to collaborative filtering. *Indones. J. Electr. Eng. Comput. Sci.* **2022**, *25*, 450. [[CrossRef](#)]
26. Aziz, R.A.; Lestari, S.; Fitria, F.; Arianto, F. Imputation missing value to overcome sparsity problems. *TELKOMNIKA Telecommun. Comput. Electron. Control* **2024**, *22*, 949. [[CrossRef](#)]
27. Huang, J.; Jia, Z.; Zuo, P. Improved collaborative filtering personalized recommendation algorithm based on k-means clustering and weighted similarity on the reduced item space. *Math. Model. Control* **2023**, *3*, 39–49. [[CrossRef](#)]
28. Bathla, G.; Aggarwal, H.; Rani, R. AutoTrustRec: Recommender System with Social Trust and Deep Learning using AutoEncoder. *Multimed. Tools Appl.* **2020**, *79*, 20845–20860. [[CrossRef](#)]
29. Pan, Y.; He, F.; Yu, H. Learning social representations with deep autoencoder for recommender system. *World Wide Web* **2020**, *23*, 2259–2279. [[CrossRef](#)]
30. Rajput, I.S.; Tewari, A.S.; Tiwari, A.K. An autoencoder-based deep learning model for solving the sparsity issues of Multi-Criteria Recommender System. *Procedia Comput. Sci.* **2024**, *235*, 414–425. [[CrossRef](#)]
31. Hiriyannaiah, S.; Siddesh, G.M.; Srinivasa, K.G. DeepLSGR: Neural collaborative filtering for recommendation systems in smart community. *Multimed. Tools Appl.* **2023**, *82*, 8709–8728. [[CrossRef](#)]
32. Liu, N.; Zhao, J. Recommendation System Based on Deep Sentiment Analysis and Matrix Factorization. *IEEE Access* **2023**, *11*, 16994–17001. [[CrossRef](#)]
33. Choi, S.-M.; Lee, D.; Jang, K.; Park, C.; Lee, S. Improving Data Sparsity in Recommender Systems Using Matrix Regeneration with Item Features. *Mathematics* **2023**, *11*, 292. [[CrossRef](#)]
34. Muhammad, D.; Chiroma, H.; Lamir, M.; Gital, A.; Musa, K.; Mustapha, A.; Lawal Abdulrahman, M.; Mohammed, D. An Ensemble Clustering Recommender Model Base on SVD Algorithms. *Int. J. Eng. Res. Technol.* **2023**, *12*, 197–204.
35. Barathy, R.; Chitra, P. Applying Matrix Factorization In Collaborative Filtering Recommender Systems. In Proceedings of the 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 6–7 March 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 635–639. [[CrossRef](#)]
36. Srivastava, A.; Bala, P.K.; Kumar, B. New perspectives on gray sheep behavior in E-commerce recommendations. *J. Retail. Consum. Serv.* **2020**, *53*, 101764. [[CrossRef](#)]
37. Kaur, B.; Rani, S. Are the customers receiving exact recommendations from the e-commerce companies? Towards the identification of gray sheep users using personality parameters. *Int. J. Perform. Eng.* **2023**, *19*, 425.
38. Nguyen, L.V.; Vo, Q.-T.; Nguyen, T.-H. Adaptive KNN-Based Extended Collaborative Filtering Recommendation Services. *Big Data Cogn. Comput.* **2023**, *7*, 106. [[CrossRef](#)]
39. Fazziki, A.; El, Y.; Alami, M.; Elhassouni, J.; Benbrahim, M. Enhancing Collaborative Filtering: Addressing Sparsity And Gray Sheep With Opposite User Inference. *J. Theor. Appl. Inf. Technol.* **2024**, *102*, 872–882.

40. El Youbi El Idrissi, L.; Akharraz, I.; Ahaitouf, A. Personalized E-Learning Recommender System Based on Autoencoders. *Appl. Syst. Innov.* **2023**, *6*, 102. [[CrossRef](#)]
41. Kulkarni, P.V.K.; Rai, S.R.; Sachdeo, R.S.; Kale, R.K. *Personalised eLearning Recommendation System*; IEEE DataPort; IEEE: Piscataway, NJ, USA, 2022. [[CrossRef](#)]
42. Jindal, H.; Agarwal, S.; Sardana, N. PowKMeans: A Hybrid Approach for Gray Sheep Users Detection and Their Recommendations. *Int. J. Inf. Technol. Web Eng.* **2018**, *13*, 56–69. [[CrossRef](#)]
43. Rashidi, R.; Khamforoosh, K.; Sheikahmadi, A. An analytic approach to separate users by introducing new combinations of initial centers of clustering. *Phys. Stat. Mech. Its Appl.* **2020**, *551*, 124185. [[CrossRef](#)]
44. Chetana, V.L.; Seetha, H. Handling Massive Sparse Data in Recommendation Systems. *J. Inf. Knowl. Manag.* **2024**, *23*, 2450021. [[CrossRef](#)]
45. Yannam, V.R.; Kumar, J.; Babu, K.S.; Sahoo, B. Improving group recommendation using deep collaborative filtering approach. *Int. J. Inf. Technol.* **2023**, *15*, 1489–1497. [[CrossRef](#)]
46. Ganesan, T.; Vellaiyan, P. An Enhanced Neural Network Collaborative Filtering (ENNCF) for Personalized Recommender System. In Proceedings of the International Conference on Recent Innovations in Computing, Tokyo, Japan, 16–18 March 2024; Singh, Y., Singh, P.K., Gonçalves, P.J.S., Kar, A.K., Eds.; Lecture Notes in Electrical Engineering. Springer Nature: Singapore, 2024; Volume 1194, pp. 183–195. [[CrossRef](#)]
47. Zhang, Z. Personalized resource recommendation method of student online learning platform based on LSTM and collaborative filtering. *J. Intell. Syst.* **2024**, *33*, 20240017. [[CrossRef](#)]
48. Gomedede, E.; de Barros, R.M.; de Souza Mendes, L. Deep auto encoders to adaptive E-learning recommender system. *Comput. Educ. Artif. Intell.* **2021**, *2*, 100009. [[CrossRef](#)]
49. Su, Z.; Lin, Z.; Ai, J.; Li, H. Rating Prediction in Recommender Systems Based on User Behavior Probability and Complex Network Modeling. *IEEE Access* **2021**, *9*, 30739–30749. [[CrossRef](#)]
50. Geng, Y.; Zhu, Y.; Li, Y.; Sun, X.; Li, B. Multi-Feature Extension via Semi-Autoencoder for Personalized Recommendation. *Appl. Sci.* **2022**, *12*, 12408. [[CrossRef](#)]
51. Zhang, Y.; Xu, H.; Yu, X. The Recommendation Algorithm Based on Improved Conditional Variational Autoencoder and Constrained Probabilistic Matrix Factorization. *Appl. Sci.* **2023**, *13*, 12027. [[CrossRef](#)]
52. Chen, S.; Guo, W. Auto-Encoders in Deep Learning—A Review with New Perspectives. *Mathematics* **2023**, *11*, 1777. [[CrossRef](#)]
53. Harper, F.M.; Konstan, J.A. The MovieLens Datasets: History and Context. *ACM Trans. Interact. Intell. Syst.* **2016**, *5*, 1–19. [[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.