Journal of Applied Data Sciences
Vol. 6, No. 4, December 2025, pp. 2792-2809

ISSN 2723-6471
2792

AMIKOM-RECSYS: Enhancing Movie Recommender System using Large Language Model (ChatGpt), Deep Learning and Probabilistic Matrix Factorization

Hanafi[1,*], Anik Sri Widowati[2], Sri Ngudi Wahyuni[3]

[1,2,3]*Universitas Amikom Yogyakarta, Jl. Ringroad Utara, Condongcatur, Depok, Sleman, Yogyakarta, 55523, Indonesia*

(Received: March 10, 2025; Revised: June 04, 2025; Accepted: September 08, 2025; Available online: October 04, 2025)

Abstract

E-commerce has become one of the most widely used digital applications globally, enabling personalized product discovery and purchasing. To support these services, recommender systems are essential, offering item suggestions based on user preferences. Most recommender systems rely on machine learning algorithms to estimate user-item relevance scores, often utilizing product ratings. However, a persistent challenge in this domain is the issue of data sparsity, where only a small fraction of users provides explicit ratings, leading to reduced accuracy in recommendation results. In this study, we introduce a novel hybrid recommendation algorithm, named AMIKOM-RECSYS, designed to address the sparsity problem and enhance rating prediction. Our model integrates three main components included a Large Language Model (LLM) using ChatGPT, a Transformer-based encoder (BERT), and Probabilistic Matrix Factorization (PMF). The LLM generates descriptive information about movies based on specific prompts, which is then passed to BERT to encode the content into meaningful 2D vector representations. These enriched embeddings are subsequently utilized by the PMF algorithm to predict missing user-item ratings. We evaluate the proposed model on two benchmark datasets, ML-1M and ML-10M using Root Mean Squared Error (RMSE) as the evaluation metric. The AMIKOM-RECSYS model achieved RMSE values of 0.8681 on ML-1M and 0.7791 on ML-10M under a 50:50 data split, outperforming several baseline models including CNN-PMF, LSTM-PMF, and Attention-PMF. These results highlight the effectiveness of integrating LLM and Transformer-based contextual understanding into matrix factorization frameworks. In future work, we plan to extend this framework by incorporating other matrix factorization techniques such as Singular Value Decomposition (SVD) and integrating additional sources of user information, including social media activity, to further improve recommendation performance.

Keywords: E-Commerce, Recommender System, Large Language Model, Deep Learning, Matrix Factorization

1. Introduction

E-commerce is specific term for online transaction through internet platform. Began in early 1990, Amazon as an online shop for book commerce initiated to developed product recommendation. The company become the most successful e-commerce business company. Amazon is the pioneer of recommender system adoption in e-commerce. Many large companies expanded to online commerce, for instance Netflix for movie business, iTunes for music, Google store for application, Uber and Grab car for online transportation, YouTube as a video portal, and Facebook for social media application. All of them are the example of major popular application that required recommender system [1].

The adoption of recommender system influenced the successful of e-commerce business. The recommender system made customer choose the product easily. The adoption of recommender system also influenced customer loyalty which will encourage customers to make repeat transactions. The loyal customer influenced growth revenue of e-commerce business. Recommender system also a part of automatically algorithm to support customer in making fit decision. The adoption of recommender system is a suitable strategy to handle flooding information in big data era due to social media application [2]. Collaborative filtering is the most popular algorithm to build e-commerce recommender system. Collaborative filtering owned some benefit such as effective in product recommendation result and supporting integration with various types of information. The algorithm of collaborative filtering relies on product rating to

***Corresponding author: Hanafi (hanafi@amikom.ac.id)**
DOI: https://doi.org/10.47738/jads.v6i4.897

produce recommendation. Rating is satisfied expression for product or services. In early developing collaborative filtering algorithm, they used traditional nearest neighbor commonly known as memory-based methods. Unfortunately, memory-based system faced the shortcoming in high computation cost, difficulty in applying them to small datasets, incompatibility with integrated information resources, and the need for extensive training. Indeed, memory-based systems owned the benefit in terms of ease of implementation, no need for training, and simple algorithm Most nearest neighbor model-based recommendation systems adopt statistical approach such as Cosine, Adjust Cosine, Spearman rank, and Pearson [3].

Model based also popular called latent factor model become a solution approach to handle the weakness of memory-based model. Most model-based algorithm involve matrix factorization. Matrix factorization is responsible for generating rating prediction, where they computed the intersection between product and customer based on rating. The adoption of matrix factorization has successfully improved the performance of memory-based approach. The big momentum of model-based approach performance was shown in the Netflix competition in 2006. The challenge in this competition was that the new model had to improve performance by 10% over the existing Netflix recommender system algorithm [4].

The adoption of model-based using matrix factorization successfully enhanced the performance of memory-based approaches. There are several popular matrix factorization models in model-based approaches such as Singular Value Decomposition (SVD) [5], [6], [7], Probabilistic Matrix Factorization (PMF) [8], and SVD++ [9]. The matrix factorization work by decreases the density of the consuming-product engagement matrix, thereby facilitating the identification of hidden correlations in customer preferences and item attributes. Even though the matrix factorization achieved better performance, the matrix factorization faces cold start and matrix sparsity problem. The cold start problem arises from new consumers who lack past activity, while matrix sparsity is caused by minimum ratings in the dataset. Specifically for matrix sparsity issue, the performance of matrix factorization task decreases significantly when faced with minimum rating [10].

The issue of matrix sparsity is a serious concern for both researchers and industrialists. To overcome the sparsity problem, most researchers attempt to enhance algorithm by using auxiliary or side information [11]. Side information refers to the information owned by consumer of product information [12]. Consumer information includes demographic information, consumer purchasing activity, and social media activity. Product information includes product description, product review information, product testimony, and product category. There is a vast amounts of information about product and consumer information in Big Data era. Moreover, the adoption of social media application in large society population produced huge information. This huge information has the potential to be implemented in recommender system application.

In the initial model, Koren [13] proposed incorporating product information in the form of timestamps into matrix factorization. According to experimental report, the adoption of timestamp into SVD demonstrated improved performance compared to the generic SVD. Aiming to enhance the adoption of side information, Koren implemented product taxonomy [14], [15]. The shortcoming of several studies above is the adoption of simple interpretation of product information.

The integration of matrix factorization model became popular approach in the early 2010s, as seen in the adoption of statistical approach to calculate product information understanding. They used a popular NLP approach called topic modelling to capture product information meaning. The product information in this study refers to product review [16]. According to previous works on enhancing matrix factorization with side information, the model faces the shortcoming in contextual understanding from an NLP perspective. The shortcoming of the model is its failure to capture the contextual understanding of product reviews.

Capturing the contextual understanding of product document in recommender system application meets serious issue. To address the problem, several researchers proposed novel models with deep learning algorithm, i.e. a recommender system model with an Auto Encoder (AE) and PMF model called Collaborative Deep Learning (CDL) [17]. The model was successfully implemented on real datasets and achieved better performance over statistical approach in previous work. From a contextual perspective, AE algorithm fails to capture product document contextual. To overcome this problem, some researcher proposed novel model on deep learning algorithm to increase performance in capturing

 ISSN 2723-6471

contextual understanding such as Graph Convolutional Neural Network (CNN) [18], Long Short-Term Memory (LSTM), Stack Denoising Auto Encoder (SDAE) and Attention mechanism. Several models also integrated the matrix factorization with Word Embedding model such as GloVe, Fasttext, and Word to Vec. The adoption of deep learning model and word embedding platform aim to enhance the algorithm in capturing context with two mechanisms including detected word order and subtle word. Indeed, capturing contexl in recent year is challenges, therefore, we attempt to enhance matrix factorization with novel hybrid algorithm in this study.

Building on the earlier explanation, the issue extends beyond just contextual understanding to the inherent bias present in customer reviews. Each review reflects a unique perspective shaped by individual preferences, experiences, and subjective opinions, making it difficult for the recommender system to accurately interpret the true value of a product or movie. Unlike numerical ratings, which provide a clear scalar value, textual reviews often convey conflicting sentiments within a single response. This presents a challenge, as a reviewer might praise certain aspects of a movie while simultaneously offering critical feedback.

According to the shortcoming of the existing research on the e-commerce recommender system problem and the potential enhancement of recommendation task performance, we have described our contribution as follows. First, we generate movie knowledge information using LLM based on ChatGPT. This knowledge is a representation of the quality of product knowledge generated automatically using LLM. The LLM model produced product knowledge with deeper dimensional context, semantic and knowledge. Second, we adopt BERT for feature extraction to convert product information obtained from LLM. Third, the transformer model is utilized to gather information from product opinion and convert it into a 2D latent space. The transformer model is then integrated into matrix factorization, with the goal of producing rating predictions. Furthermore, we adopt a hybrid algorithm in which transformer models are used to represent both consumer and product information, and these representations are incorporated into matrix factorization to further improve recommendation accuracy.

2. Literature Review

Recommender systems play a crucial role in stimulating revenue growth for e-commerce businesses. Successful adoption of a recommender system has a significant impact on revenue growth. However, recommender systems continue to struggle with the matrix sparsity problem. The minimum rating is the root cause of the problem. It influences the recommendation result, which may be deemed unsuitable for the consumer. Eventually, this will lead to a decline in market sales. Therefore, the sparse rating matrix needs to be handled to enhance suitable e-commerce recommendations. Researcher aim to solve the problem through various scenario, including collecting kind of information, enhancing recommendation algorithm and developing hybrid algorithm of deep learning model.

The additional information to enhance matrix factorization began in early 2010 by Koren [16], who adopted item information. Another author proposed traditional bag of word mechanism to capture product document information. The topic modelling mechanism categorical bag of word model [16], [19]. The adoption of review for collaborative filtering has been proven better achievement over generic collaborative filtering based on matrix factorization. Unfortunately, bag of word model faces the weakness in contextual understanding. The issue influences the performance of collaborative filtering task to produce product recommendation. Most of e-commerce users are reluctant to share the product rating. Figure 1 represents sparse data due to minimum rating. According to scientific reference, ML.1M as popular dataset with rating matrix less than 4%, ML.10 only less than 1% [10].

ISSN 2723-6471

Figure 1. Illustration of movie sparsity data

At the beginning of the implementation of collaborative filtering algorithms for recommendation systems, SVD became favorite model [20], [6], [21]. Moreover, the adoption of matrix factorization or model-based successfully improved gthe performance of memory-based model. In the era of deep learning around the early 2010s, many researchers strived to enhance the performance of collaborative filtering work by using various deep learning model, such as adoption of Auto Encoder and matrix factorization [22], [23]. The Auto Encoder typically features extraction mechanism with encoder dan decoder approach. The Auto Encoder approach is responsible for calculating text information for product representation. The adoption of Auto Encoder has demonstrated better achievement compared to bag of word mechanism.

Product document becomes essential information to enhance collaborative filtering. According to NLP perspective, the capturing semantic and contextual understanding of text information is the major problem in collaborative filtering with product review document. Kim et al [24] proposed a novel model using CNN and word embedding based on GloVe. They used PMF to generate rating prediction, GloVe is in charge of text feature extraction and CNN responsible to capture text understanding information. The model succeeded in enhancing previous work based on auto encoder model and PMF. The hybrid model of CNN and PMF was implemented on two category datasets included ML.1M and ML.10M, which consist of movie reviews collected from the IMDB movie portal. According to the experimental report, the algorithm proposed by Kim achieved RMSE scores of 0.8549 (ML.1M) and 0.7930 (ML.10M) [18], whereas the Collaborative Deep Learning (CDL) model obtained 0.8879 (ML.1M) and 0.8186 (ML.10M) [16]. In addition, the Collaborative Topic Regression (CTR) achieved 0.8969 (ML.1M) and 0.8275 (ML.10M) [13].

Previous works using product review faced the shortcoming in contextual bias, for instance as illustrated in figure 2. A customer may comment the acting and storyline of a movie and highlight dissatisfaction with its pacing or plot development. This duality within a single review creates ambiguity, complicating the system's ability to gauge whether the overall sentiment is positive or negative. Therefore, addressing such biases and understanding the nuanced nature of customer feedback is crucial for improving the accuracy and effectiveness of recommendation systems.

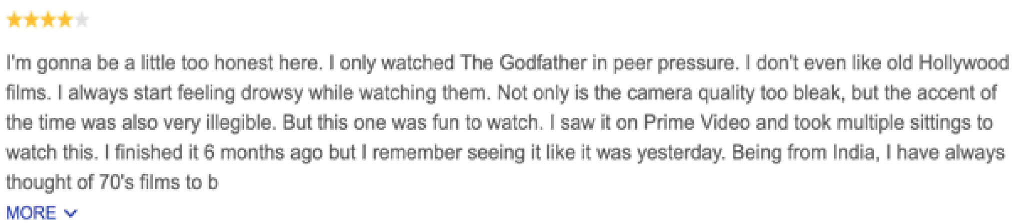

Figure 2. Bias context of a movie review from IMDb

The adoption of LLM model from collaborative filtering aim to handle unfair movie opinion. Figure 3 below shows a knowledge expression about movie without tendency. We believe that the result of movie expression produced by LLM are more objective and fair compared to movie review from customer.

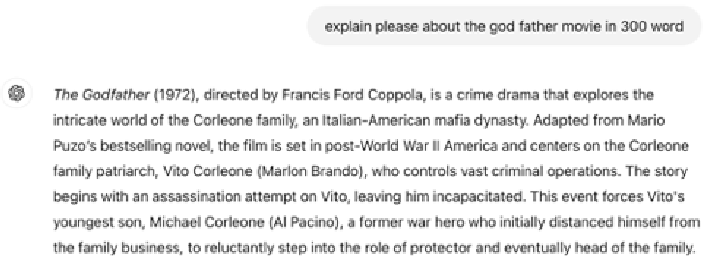

Figure 3. A movie knowledge understanding by LLM from Chat-GPT

Table 1 below demonstrates the previous work performance achievement and algorithm description of collaborative filtering in e-commerce.

Table 1. Previous state-of-the-art of collaborative filtering methods

Method	Description	RMSE
SVD	SVD is one version of traditional collaborative filtering. SVD employs matrix factorization based on low-rank dimensions to find correspondence between the item and the user. The correspondence between them can be found by generated rating prediction [20].	0.8638
PMF	PMF is an enhanced version of the SVD model. Like the SVD model, PMF generates correspondence between the user and the item. Different from the SVD, PMF considers adopting the probabilistic mechanism [25].	0.8971
LDA	The adoption of the matrix factorization version faced shortcomings in performance, degrading significantly against sparse ratings. The LDA is responsible for integrating product documents into PMF to produce robust models in rating prediction [18].	-
CTR	Collaborative Topic Regression (CTR) is an advanced version of the LDA model. Enhancement of understanding product documents is integrated with PMF. Topic regression is an essential algorithm to capture the word vector representation of product documents [26].	0.8969
CDL	Collaborative Deep Learning (CDL) is an advanced CTR model involving PMF. Auto Encoder in the term in deep learning context responsible for capturing product document meaning. The document is contextually integrated with PMF in generating rating prediction [16].	0.8879
DCCR	Deep Collaborative Conjunction Recommendation (DCCR). DCCR is developed using MLP and AE. The Multi-Layer Perceptron (MLP) is responsible for detecting user latent vectors, and the Auto Encoder (AE) is responsible for calculating item latent vectors [27].	0.8356
ConvMF	ConvMF is a kind of collaborative filtering model. ConvMF implements PMF and CNN. CNN, as a deep learning platform, is responsible for transforming item document representation. Meanwhile, PMF is a representation of the model-based collaborative filtering responsible for producing rating prediction [24].	0.8549
LSTM-PMF	LSTM-PMF is an enhanced version of PMF. LSTM is responsible for transforming product document vectors. Meanwhile, PMF is responsible for generating rating predictions. LSTM is implemented due to its ability to detect word order [28].	0.8407
SDAE-LSTM-PMF	SDAE-LSTM-PMF is an advanced of LSTM-PMF. Different from LSTM-PMF, SDAE-LSTM-PMF considered SDAE as user latent vector representation. LSTM is considered to transform product documents into document vector space. PMF is responsible for producing rating predictions [29].	0.8583

One of the major challenges in recommender systems is capturing contextual information, which has been addressed using deep learning algorithms such as LSTM and attention mechanisms. For instance, Hanafi et al [28], [30], [31], [32] proposed a hybrid model combining LSTM with PMF. The hybrid model implemented in real dataset includes ML.1M and ML.10M. The movie text information is collected from IMDB movie review. The experimental results show that the LSTM and PMF achieve better performance over adoption of CNN and PMF. The LSTM model achieved

0.84079 (ML.1M) and 0.7902 (ML.10M) based on RMSE. On the other hand, the adoption of LSTM achieved better performance than adoption of CNN and PMF.

The adoption of hybrid attention mechanism, SDAE and PMF to improve rating estimation performance in recommender system have been conducted by Hanafi [29]. In this model, user information is transformed using SDAE, while product information is captured through the attention model. In the common collaborative filtering approach based on PMF, majority model implemented zero mean spherical Gaussian to convert user information representation. The adoption of Attention and SDAE that integrated into PMF demonstrated better performance over ConvMF, CDL, and PHDMF. They believe that the performance is influenced by the use of Attention and SDAE in user information extraction. According to the experimental result, RMSE value for ML.1M was 0.84315, and ML.10M was 0.755189. The performance indicated the Attention and SDAE have significant impact in collaborative filtering task performance.

MovieLens datasets contain two files including user information and rating of product information. The majority of proposed model use Gaussian normal distribution to transform user information. Meanwhile, product information is transformed using deep learning model such as CNN and LSTM. Unlike the previous work mentioned above, the DDL-PMF considers adopting SDAE to convert user information representation. This model is inspired by Kim's ConvMF [24] and Liu et al.'s PHDMF [33]. The adoption of SDAE and LSTM into PMF has proven to provide significant achievement over both ConvMF and PHDMF, achieving 0.85837 (ML.1M) based on RMSE evaluation matrix. They believe that the significant performance of DDL-PMF is influenced by dual side deep learning adoption to handling sparse rating matrix.

Aiming to handle the lack of contextual meaning in product review, a model using Attention and CNN were created, with PMF responsible for producing rating prediction [34]. The proposed model was implemented in real e-commerce datasets including ML.1M and ML.10M from MovieLens. The adoption of CNN and Attention with PMF achieved 0.7402 (ML.1M) and 0.7601 (ML.10M). The Attention mechanism is a novel algorithm in neural language model. Attention has the ability to capture essential value of word order in training process. The adoption of Attention plays important role in enhancement of recommender system task performance. The model achieved tremendous result compared with ConvMF and PHDMF.

A novel matrix factorization model based on hybrid ensemble learning using Adaboost was proposed in [35]. In the initial step, they considered to clustering user-item with fuzzy technique. The model trained with neural network to enhance the performance of rating matrix. Another model with adoption of fusion approach that integrated with PMF have been proposed by [36]. The adoption of fusion approach achieved better performance over traditional matrix factorization. They considered to implement on four types of datasets. The adoption of deep learning has been shown to provide better performance in several computer science application such as Cyber security [37], [38], and enhancing recommender system algorithm [39], [40]. Table 2 summarizes several previous collaborative filtering algorithms and highlights the positioning of the proposed AMIKOM-RECSYS improvement model within this research landscape.

Table 2. Summary of previous collaborative filtering algorithm and positioning of AMIKOM-RECSYS improvement model

Ref.	Collaborative Filtering	Rating	Item side document	LLM	Matrix factor		Deep learning				Pre-trained model		
					PMF	SVD	AE	CNN	LSTM	ATT.	GloVe	W2V	BERT
[15]	PMF	√	-	-	√	-	-	-	-	-	-	-	-
[12]	LDA	√	√	-	√	-	-	-	-	-	-	-	-
[13]	CTR	√	√	-	√	-	-	-	-	-	-	-	-
[16]	CDL	√	√	-	√	-	√	-	-	-	-	-	-
[23]	SVD+AE	√	√	-		√	√	-	-	-	-	-	-

[18]	CNN+PMF	√	√	-	√	-	-	√	-	-	-	-	-
[24]	LSTM+PMF	√	√	-	√	-	-	-	√	-	√	-	-
[29]	SDAE+ATT+PMF	√	√	-	√	-	-	-	-	√	√	-	-
Our	AMIKOM-RECSYS	√	√	√	√	-	-	√	-	-	√	-	√

3. Methodology

Our model consists of several phases to improve collaborative filtering recommender systems. The evaluation method for the detail enhancement algorithm and data collection described in the section below.

3.1. Hybrid Collaborative Filtering Model

Collaborative filtering is the most popular algorithm for producing product information. The algorithm was designed to compute basic user activity in the past such as rating, comment, and testimony. However, collaborative filtering faces the problem in minimum rating. Hence, the adoption of hybrid schema with several algorithm becomes the essential strategy. The illustration of hybrid model in AMIKOM-RECSYS can be seen in figure 4. The hybrid model consists of PMF, BERT and LLM mechanism.

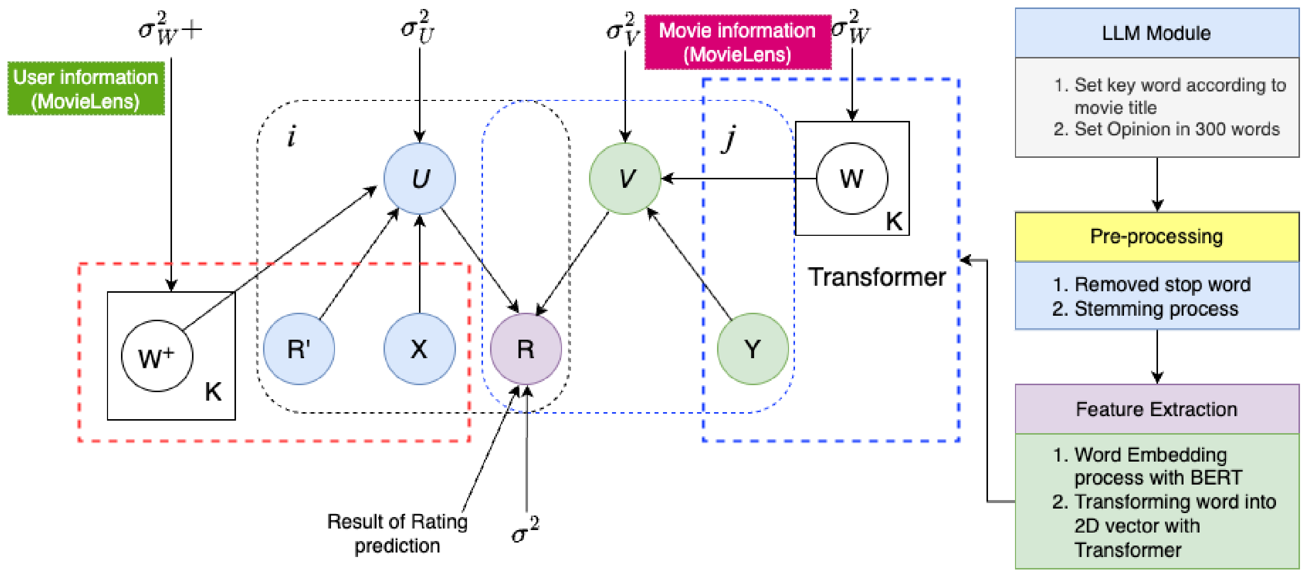

Figure 4. Architecture of AMIKOM-RECSYS

3.1.1. PMF for Generating Rating

PMF is advanced version of SVD that uses the Gaussian normal distribution to distribute ratings in line with the rules of probability. The main method for breaking down the rating matrix can be shown with two smaller matrices used for calculations as follows: Let M denote a movie, N signifies a user, and the rating values range from 1 to K. Meanwhile, Rij represents user i preference for movie j. Where U and V denote the latent matrices for users and movies, respectively. Variable U is derived from U∊RD×N, whereas variable V is derived from V∊RD×N. The PMF applies an ordinal probabilistic linear method that employs the Gaussian normal distribution. It depicts users and movies as vectors derived from the rating allocations of respondents. The exact formula of the distribution is presented in Eq. 1.

$$p(R|U,V,\sigma^2) = \prod_{i=1}^{N}\prod_{j=1}^{M}\left[N(R_{ij}|U_i^T V_j,\sigma^2)\right]^{I_{ij}} \quad (1)$$

The MovieLens datasets for recommender system consist of two information: user information and item information (user.dat). User information represents user demography information, while, item information (rating.dat) represents rating information of the product. The fundamental of probabilistic mechanism of PMF requires Probability Density

Function (PDF). The PDF mechanism is the basic approach to create PMF model. PDF is a crucial mathematical instrument for delineating the probability distribution of continuous random variables. It enables the calculation of interval probabilities and elucidates the distribution of potential values of a random variable. The formula to calculate PDF is shown in Formula 2 below.

$$N\left(R_{ij}\middle|U_i^T V_{j,}\sigma^2\right) \tag{2}$$

According to the perspective of user information, we can calculate the adoption of zero mean spherical Gaussian Normal Distribution. The detail formula to achieve zero mean spherical is explained in Equation 3. Statistically, the Zero Mean Spherical Gaussian Normal Distribution is a variant of the normal (Gaussian) distribution with two key features: the variance is isotropic and the mean is zero.

$$p\left(U\middle|\sigma_U^2\right) = \prod_{i=1}^{N} \mathcal{N}\left(U_i \middle|0, \sigma_U^2 I\right) \tag{3}$$

σ represents the standard deviation, I_{ij} represents indicator function that determined 1 if consumer i rated for movie j and 0 otherwise, and μ represents mean level. The Chatgpt variable in the formula represents the item knowledge from the ChatGPT-based LLM application. The Equation 4 below is the mechanism to be integrated into matrix factorization based on PMF.

$$p\left(\left(V\middle|W, X, \sigma_V^2\right) = \prod_j^M \mathcal{N}\left(V_j \middle| chatgpt\left(W, X_j\right)\sigma_w^2 I\right)\right) \tag{4}$$

Product or item is essential aspect to generate rating prediction. The symbol of item value represents vj ,which it was derived from Equation 5 as follow:

$$v_j = chatgpt\left(W, X_j\right)\varepsilon_j \tag{5}$$

The normal distribution of probabilistic mechanism can be generated with Equation 6 as follow:

$$\varepsilon_j \sim N\left(0, \sigma_v^2 I\right) \tag{6}$$

The document of product knowledge from ChatGPT can be transformed with Transformer and Bert mechanism. The formula to produce document latent factor can be calculated with equation 7 as follow:

$$p\left(W\middle|\sigma_W^2\right) = \prod_k N\left(W_k|0, \sigma_W^2\right) \tag{7}$$

To provide a detailed explanation of the complete formula in the AMIKOM-RECSYS model, it is essential to first define each of the components and variables in the context of the model. The AMIKOM-RECSYS formulation integrates elements of collaborative filtering, matrix factorization, and deep learning with external information (like ChatGPT-generated knowledge). The formula is based on probabilistic matrix factorization model, incorporating both user and item latent features to predict ratings. Table 3 presents this detailed breakdown; it might include definitions of these components and how they interact within the model.

Table 3. Detail description of symbol in equation

Symbol	Description	Symbol	Description
p	Probability of the ratings	W_k	Latent factors for user k
v_j	Item feature vector j	σ_W^2	Variance for (user feature variance)
ε_j	Error term for item j	W	User matrix (latent factors matrix)
σ_v^2	Variance of the item latent factors Vj	X_j	Feature matrix for item j
σ_U^2	Variance of the user latent factors Ui	U	User matrix (latent factors matrix)
R_{ij}	Rating for user i and item j	V	Item matrix (latent factors matrix)
σ^2	variance value	N	Total number of ratings observed
U_i^T	Transposed user feature vector for user i	M	Total number of items

3.1.2. LLM for Generating Movie Knowledge Information

The important phase of our proposed model is generated product information knowledge based on LLM using ChatGPT. ChatGPT is based on the Transformer model, introduced in the paper "Attention is All You Need" [41]. The Transformer uses a mechanism called self-attention, which allows it to process and relate words in a sentence regardless of their distance. Unlike older models such as RNNs and LSTMs, Transformers can process data in parallel, making training much faster and more scalable. Generative Pretrained Transformer (GPT) is a type of Transformer designed specifically for generating human-like text. GPT architecture consists of the decoder part of the original Transformer model. It is trained in an unsupervised manner on a huge amount of text data predicting the next word given previous words that is called causal language modelling. GPT model requires Pretraining process. The model reads a massive dataset like books, websites, scientific journal and learns the structure of language, grammar, facts, and reasoning patterns. After pretraining, it is fine-tuned using Reinforcement Learning from Human Feedback. Humans rate responses, and the model learns to generate better, safer, and more helpful answers. Special techniques like Proximal Policy Optimization are used during this step.

Each GPT model version has different sizes. GPT-2 owned up to 1.5 billion parameters, GPT-3 owned 175 billion parameters, and GPT-4 owned estimated hundreds of billions of parameters. A model is made of many Transformer blocks stacked on top of each other, each with multi-head self-attention layers, Feed-forward neural networks, Layer normalization and residual connections.

GPT represents key feature including contextual understanding, few-shot learning, and emergent abilities. Contextual understanding enables the model to remember the whole conversation context when generating each new response. Few-shot learning allows the model to perform tasks by seeing just a few examples. Emergent abilities as the model gets bigger, it develops surprising capabilities like reasoning, code generation, summarizing, etc. The architecture of ChatGPT show on figure 5.

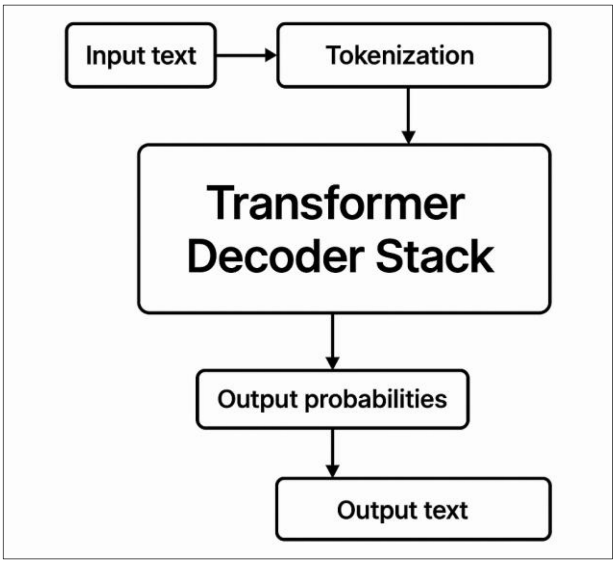

Figure 5. ChatGPT architecture

3.1.3. BERT Pre-Trained Model

BERT is one of the most popular pre-train models for transforming word into vector. BERT has revolutionized the process of creating and applying word embeddings in NLP. Unlike traditional models such as Word2Vec or GloVe,, which generate a single static vector for each word, BERT produces contextual embeddings. This means that the same word will have different vector representations depending on its surrounding the context. For example, the word "bank" in the sentence "I deposited money at the bank" will have a different embedding than "He sat by the river bank." This dynamic ability to understand context makes BERT embeddings much more powerful and flexible than earlier approaches.

ISSN 2723-6471

The first step in BERT's embedding process is tokenization. BERT uses a technique called word piece tokenization, which breaks words down into smaller sub word units. This approach allows BERT to handle rare or unseen words effectively by composing them from known pieces. For example, the word "unbelievable" might be broken down into ["un", "##believ", "##able"]. Each of these sub words will have its own initial embedding, allowing BERT to represent a vast vocabulary without exploding the size of the model. Once the input text is tokenized, each token is mapped to an initial token embedding. However, this token embedding is insufficient. BERT enhances the input representation by combining three types of embeddings: token embeddings, segment embeddings, and position embeddings. Token embeddings capture the meaning of the individual token itself. Segment embeddings are used when BERT processes pairs of sentences, helping the model distinguish between tokens belonging to different sentences. Position embeddings provide information about the order of tokens in the sequence, as transformers by themselves do not have any inherent sense of word order.

The real power of BERT's embeddings comes after the input passes through its deep stack of Transformer encoder layers. Each layer applies self-attention mechanisms that allow every token to consider and incorporate information from every other token in the sequence, in both forward and backward directions. Through this process, the model dynamically adjusts each token's embedding based on the words surrounding it. As a result, the output embeddings at higher layers become rich, context-sensitive representations that understand the nuances of the input sentence. The basic architecture of BERT is illustrated in figure 6.

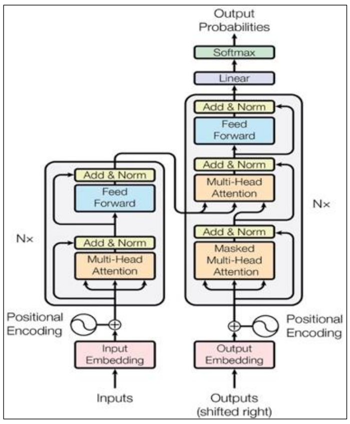

Figure 6. Basic architecture Transformer to develop BERT model

Figure 7 illustrates the attention-based architecture used for processing text document movie information, within the AMIKOM-RECSYS hybrid model. On the left, sequential text inputs (Text t-2, Text t-1, Text t) represent time-ordered movie information or opinions derived from an LLM. Each text block is encoded into a latent representation (Yt−2,Yt−1,Yt-Y) that captures semantic and contextual meaning. The red arrows show temporal dependencies between previous time steps and the current input, enabling the model to leverage historical context when processing the current text.

The right-hand section contains the attention network, where each input's latent vector is weighted according to its relevance. The parameters λM (movie-specific attention) and λT (temporal attention) are combined through a weighted sum using factor α\alpha, balancing movie feature importance against temporal sequence information. The resulting attention scores (ρt−2, ρt−1, ρt) quantify how much each prior context influences the current prediction.

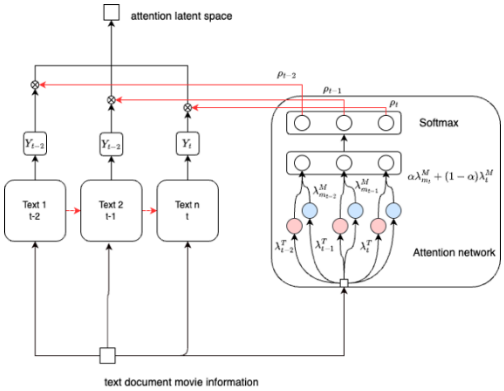

Figure 7. Transformer mechanism to transformed movie information

Finally, a softmax function normalizes these attention scores into probabilities, ensuring interpretability and stability in downstream rating prediction. By integrating temporal text dependencies and attention weighting, this architecture enhances collaborative filtering with richer, context-aware representations of movie content, thereby mitigating bias from raw user reviews.

3.2. Datasets for Recommender System

To assess the performance and ability of our proposed algorithm to handle sparse rating matrices, we employed the MovieLens datasets, specifically ML.1M and ML.10M. The ML.1M dataset contains approximately 1 million ratings, while ML.10M includes 10 million ratings, with the former exhibiting higher sparsity. Our algorithm also incorporates movie review data to improve rating predictions. Using a LLM, specifically ChatGPT, we automatically generated movie opinions, enriching the dataset with additional insights. Table 4 provides a detailed explanation of the datasets' characteristics, including sparsity levels and the integration of movie reviews into the recommendation process.

Table 4. Characteristics of MovieLens recommender system datasets

Dataset	LLM	Users	Items	Ratings	Sparse
ML.1M	3,706	6,040	3,706	1 million	4.47%
ML.10M	10,677	69,878	10,677	10 million	1.34%

To enhance the adoption of document generation with a deeper understanding, we integrated movie opinions from a LLM. LLMs possess a robust ability to comprehend context and the dynamics of document meaning, which significantly improves how collaborative filtering models generate rating predictions. Our model distinguishes itself from previous approaches, which typically relied on product reviews provided directly by users. In many cases, customer reviews can be biased or unfair, limiting their effectiveness. By using LLM-generated movie opinions, our model mitigates these biases, offering a more balanced and comprehensive understanding of product features, ultimately leading to improved recommendation accuracy.

3.3. Evaluation result with RMSE

This inquiry aims to assess the algorithm's efficacy using the RMSE assessment metric. This study entails partitioning the information set into two distinct segments: training data and testing data. The rating sparsity level is established at 10% increments, with 10% of the data designated for training and 90% for testing. The data training is allotted the highest level of sparseness, whereas the data testing is assigned the lowest level of sparseness. The formula for computing the RMSE to evaluate the outcomes of rating prediction is below.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i,j} Z^P_{i,j}(R_{ij} - \hat{R}_{ij})^2} \quad (7)$$

The parameter N represents the quantity of rating numbers, whereas $Z^P_{i,j}$ denotes the result of the test rating value. It pertains to the juxtaposition of the real rating value derived from information and the rating number forecasted by a model.

ISSN 2723-6471

4. Results and Discussion

In this study, we implemented several scenario experiments, including the adoption of two types of datasets. The scenario aims to observe how our model faced two typical datasets and level of sparsity data. The detail of the analysis and discussion of our experiment in two categories of datasets is described below.

4.1. Experiment on ML.1M Datasets

The performance of overall hybrid deep learning algorithm on ML.1M can be seen on figure 6. The experiment applies five algorithms in collaborative filtering where they adopted five major algorithms incorporating both product document information and matrix factorization based on PMF. The first collaborative deep learning considers only the PMF algorithm [15]. The result of the rating prediction, evaluated using RMSE, are presented on figure 6 (ML.1M). As shown by the blue line color, the rating prediction result achieved lower performance over PMF and deep learning approach with product document enhancement.

The second previous work was implemented on ML.1M with hybrid model using GloVe, CNN and PMF [18]. They collected movie document information from IMDB. IMBD is the most popular movie portal review. In this study, they integrated movie document information with latent factor. The result show in orange line color (figure 6). The adoption of product document achieved significant performance over traditional collaborative filtering using PMF only. The adoption of word embedding model using GloVe influences capturing contextual of the document. The proposed model by Kim [24] faced the shortcoming for failing in detecting product document understanding. The major reason of the shortcoming is the adoption of CNN where this deep learning algorithm cannot consider word order in the product review. On the other hand, the share weight value achieved high score representation of product document that represent W to support V value as represent product document.

The drawback of CNN was improved by LSTM and Attention adoption. The performance of LSTM and Attention mechanism show in figure 6. The yellow and grey color represent LSTM and Attention, where they demonstrated better achievement over CNN. Compared to LSTM [33], the adoption of Attention achieved slightly better performance in rating prediction. LSTM and Attention owned the different characteristic in capturing product document context. Attention focused to the area with contrast value of the document information [28].

In the final experiment, we implemented a novel hybrid algorithm that incorporated movie knowledge information generated by the ChatGPT model. In contrast with the previous work, our model called AMIKOM-RECSYS considered essential movie knowledge information while the previous work considered movie review information. Movie review possible to face the bias information from user because each user owns unique perspective for a movie. The result of AMIKOM-RECSYS are presented in figure 8 below. The dark blue color in figure 8 represents the experiment result on ML.1M.

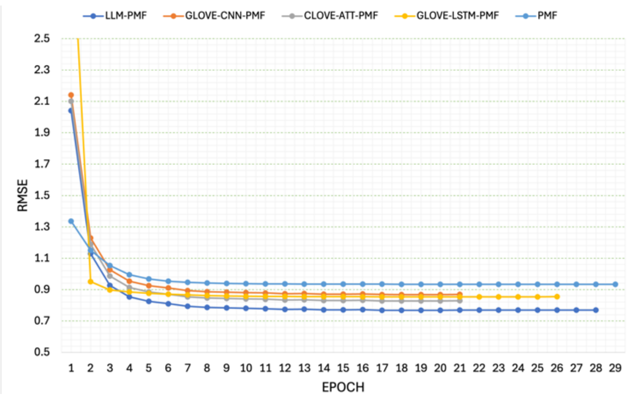

Figure 8. Comparison collaborative filtering on ML.1M

The experimental report for ML.1M is demonstrated in table 3. The data were split into training and testing sets at 10% interval. The experimental report shows that PMF performance achieved low performance compared to modern collaborative filtering with product document in overall scenario splitting datasets. However, PMF achieved best performance in splitting data scenario in 10:90 where 90% for data training and 10% representation for testing data representation. In this experiment scenario, PMF achieved 0.9045.

The first initial deep learning method for collaborative filtering was introduced by Kim et al [24]. This model adopted word pre-trained model using GloVe, where CNN were employed to transform product document information into vector, and PMF was used to produce rating prediction. This Kim model achieved 0.9954 in 90.10 splitting ratio datasets. Kim model achieved the best performance in 0.8497 in 10:90 data ratio. They claimed that the performance influences word embedding and CNN. CNN are particularly effective in detecting contextual patterns using convolutional process. In contrast, LDA relied solely on word frequency calculation and therefore failed to detect product document context.

Similar to the model proposed by Kim, another deep learning algorithm, LSTM, was adopted. LSTM is a deep learning model that consider sequential aspect mechanism. The adoption of sequential mechanism aims to detect word order in a document. Furthermore, LSTM is able to capture product document contextual. The experimental results, presented in table 5, demonstrate that LSTM outperformed both CNN and AE. The adoption of LSTM achieved 0.9928 in 90:10 splitting data ratio and 0.8407 in 10:90 splitting data ratio. These findings confirm that the adoption of sequential mechanism provides better performance compared to CNN.

The adoption of novel neural language model based on attention mechanism proposed by researchers [28]. The model inspired by previous work of Kim [24] and Hanafi [37], [39], [38], [42]. The goal of this model was to enhance deep contextual understanding based on sequential-to-sequential aspect. The model successfully improved upon the previous state-of-the-art in recommender system based on document context. Their models achieved 0.8407 in 10:90 data splitting ration on ML.1M. Attention mechanism only consider essential value on the training process of word vector. This technique becomes key essential performance of capturing document contextual. In addition, the result demonstrated that the share weight of W value achieved representative value.

Table 5. Experiment result of collaborative filtering on ML.1M

Ratio	[43]	[44]	[33]	[28]	our
90:10	1.6469	1.2657	0.9868	0.9928	0.9478
80:20	1.2657	0.9276	0.9488	0.9321	0.9188
70:30	1.1118	0.9057	0.9305	0.8999	0.8947
60:40	1.0399	0.8852	0.9132	0.8845	0.8833
50:50	0.9906	0.8778	0.8981	0.8711	0.8681
40:60	0.9589	0.8677	0.8893	0.8615	0.8556
30:70	0.9336	0.8687	0.8814	0.8547	0.8441
20:80	0.9113	0.8557	0.8723	0.8474	0.8313
10:90	0.9045	0.8497	0.8691	0.8407	0.8231

4.2. Experiment on ML.10M Datasets

Sparsity issue becomes serious problem for e-commerce business. Our experiment considered to implement on large-scale datasets characterized by extreme sparsity, specifically the ML.10M datasets. The result, summarized on figure 9, demonstrate the performance of five algorithm in handling huge sparse data problem. The purple line represents the performance of PMF, which exhibited low performance in handling extreme sparse data. Furthermore, PMF also required more epoch to reach convergence. In contrast, the orange line represents the performance of GloVE, CNN and PMF. They achieved better performance compared to traditional PMF. The benefit of the hybrid model is the ability in reducing the number of epochs. Moreover, the adoption of product document influenced the performance of the model.

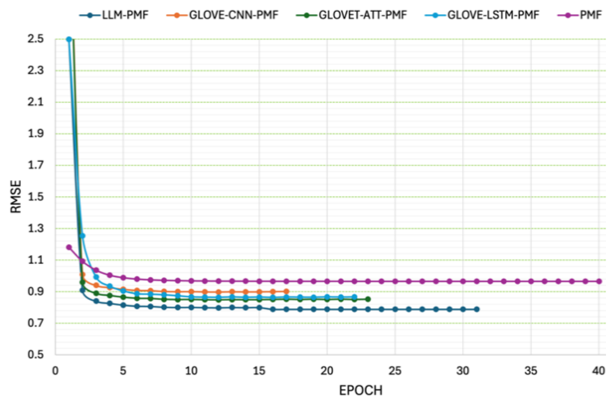

Figure 9. The comparison of collaborative filtering on ML.10M

Another adoption of product document and deep learning model show on blue and green line. The blue line represents LSTM and PMF implementation. The experimental results show that the adoption of LSTM achieve better performance than CNN. We believe that the performance of LSTM is influenced by the ability in capturing document contextual understanding. The CNN mechanism employes dimensional reduction, while the LSTM implements time series mechanism. The benefit of time-series models is their ability to capture word order and subtle word. This is the mechanism to catch the semantic and contextual understanding of the document.

In contrast to CNN and LSTM, the green line represents the Attention mechanism, a novel neural language model with ability to capture deeper contextual understanding of product document. The adoption of Attention achieved better performance than both CNN and LSTM. Unlike these models, the Attention mechanism focuses only on the most relevant parts of the data sequence during training, which is the key factor behind its superior performance. The majority of novel language model considers to implement Attention mechanism including transformer algorithm.

In the last experiment on ML.10M, we consider to implement Bert platform. Bert is a novel word embedding pre-trained model inspired by Transformer algorithm. In this model, we consider to generate a novel model of product knowledge from ChatGPT. Unlike the previous work that relied solely on customer review, our method leveraged ChatGPT to produce deeper and more representative product knowledge. This deeper knowledge understanding contributed to improving the performance of PMF in generating rating prediction. The experimental result, illustrated in dark green line show the adoption of ChatGPT and Bert outperformed the previous work.

The experiment result, explained in table 6, shows the comparison result of recommender system. The experiments were conducted using training ratio data interval in 10%. The results indicate that PMF alone achieved relatively low performance [43] whereas deep learning models such as CNN [44], LSTM [33] and Attention [28] demonstrated superior performance over traditional PMF. The model mentioned above used product review and deep learning algorithm. Compared with the previous experiment on ML.1M, the experiment on ML.10M achieved tremendous performance with significant effectiveness. For example, PMF achieved 0.8883 in data ratio 50:50, CNN was 0.8360, LSTM was 0.7996, Attention was 0.8156 and LLM model achieved 0.7791.

The result demonstrates that the adoption of various product information significantly improves performance. Moreover, a deeper understanding of product knowledge also influences significant performance of rating prediction. For instance, the adoption of LLM and PMF achieved tremendous result on 10:90 data training ratio with 0.7490. We attribute this performance influenced the adoption of Bert, PMF and LLM implementation for product knowledge information.

Table 6. The experimental result of collaborative filtering on ML.10M

Ratio	[43]	[44]	[33]	[28]	Our
90:10	1.2753	0.9362	1.1782	0.9550	1.3794
80:20	1.0523	0.8933	0.8353	0.8911	0.9006

70:30	0.9651	0.8662	0.8190	0.8518	0.7984
60:40	0.9182	0.8467	0.8065	0.8273	0.7856
50:50	0.8883	0.8360	0.7996	0.8156	0.7791
40:60	0.8667	0.8279	0.7922	0.8096	0.7701
30:70	0.85071	0.8205	0.7825	0.8027	0.7640
20:80	0.8405	0.8127	0.7799	0.7973	0.7551
10:90	0.8279	0.8050	0.7618	0.7902	0.7490

Our proposed model, AMIKOM-RECSYS, incorporates novel movie knowledge information. Knowledge generated by ChatGPT provided more representative insights compared to product reviews written by consumers. It can be concluded that the product knowledge suitable to integrate with matrix factorization in enhancing sparse rating matrix. The product knowledge from LLM achieved tremendous performance over product review. One key reason is that product knowledge is derived from credible sources, whereas product reviews are based on individual customer opinions.

AMIKOM-RECSYS achieved better performance on ML.10M, with an RMSE of 0.7490 in 10:90 ratio splitting datasets. In comparison, AMIKOM-RECSYS achieved only 0.8231 on ML.1M. Notably, this improvement was obtained despite the extreme sparsity of ML-10M. This achievement can be attributed to the substantially larger number of ratings in ML.1M, which increases the contribution of product knowledge. Thereby, the results enhance the performance of PMF in predicting ratings.

5. Conclusion

The sparsity problem, which initially emerged as a significant challenge in early collaborative filtering recommender systems due to the insufficient availability of ratings, continues to hinder the accuracy of rating predictions. To address this issue, our proposed model, AMIKOM-RECSYS, enhances prediction performance by integrating advanced information from LLM. This novel incorporation of LLM-generated product knowledge significantly improves the performance of the PMF model, showcasing a marked improvement over traditional approaches that rely on customer-generated movie reviews. The key limitations of customer reviews are the inherent bias in the opinions shared, which can distort the recommendation process. In contrast, the adoption of LLMs, with their deeper and more comprehensive understanding of product knowledge, enables the model to mitigate these biases and offer more accurate, unbiased insights into user preferences. As a result, the model's ability to predict ratings is significantly enhanced, offering a more reliable recommender system. For future work, we intend to extend the scope of knowledge integration by incorporating additional perspectives, including customer profiling in social information, enhancing the PMF model with SVD, and adopting LLM with various schematic prompt.

6. Declarations

6.1. Author Contributions

Conceptualization: N.H., A.S.W., and S.N.W.; Methodology: A.S.W.; Software: N.H.; Validation: N.H., A.S.W., and S.N.W.; Formal Analysis: N.H., A.S.W., and S.N.W.; Investigation: N.H.; Resources: A.S.W.; Data Curation: A.S.W.; Writing Original Draft Preparation: N.H., A.S.W., and S.N.W.; Writing Review and Editing: A.S.W., N.H., and S.N.W.; Visualization: N.H.; All authors have read and agreed to the published version of the manuscript.

6.2. Data Availability Statement

The data presented in this study are available on request from the corresponding author.

6.3. Funding

This Study supported by Ministry of Education, Research and Technology in the 2024 Fundamental Research Program with grant number 126/C3/DT.05.00/PL/2025

6.4. Institutional Review Board Statement

Not applicable.

6.5. Informed Consent Statement

Not applicable.

6.6. Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] J. Munson, B. Cummins, and D. Zosso, "An introduction to collaborative filtering through the lens of the Netflix Prize," *Knowledge and Information Systems*, vol. 67, no. 4, pp. 3049–3098, 2025, doi: 10.1007/s10115-024-02315-z.

[2] E. G. Muñoz, J. Parraga-Alava, J. Meza, J. J. Proaño Morales, and S. Ventura, "Housing fuzzy recommender system: A systematic literature review," *Heliyon*, vol. 10, no. 5, pp. 1-14, Mar. 2024, doi: 10.1016/j.heliyon.2024.e26444.

[3] W. Zhang and Z. Wu, "E-commerce recommender system based on improved K-means commodity information management model," *Heliyon*, vol. 10, no. 9, p. 1-15, May 2024, doi: 10.1016/j.heliyon.2024.e29045.

[4] J. Wang, L. Zhu, T. Dai, Q. Xu, and T. Gao, "Low-rank and sparse matrix factorization with prior relations for recommender systems," *Applied Intelligence*, vol. 51, no. 6, pp. 3435–3449, 2021, doi: 10.1007/s10489-020-02023-5.

[5] S. Ahmadian, K. Berahmand, M. Rostami, S. Forouzandeh, P. Moradi, and M. Jalili, "Recommender Systems based on Non-negative Matrix Factorization: A Survey," *IEEE Transactions on Artificial Intelligence*, vol. 2025, no. 1, pp. 1-21, 2025, doi: 10.1109/TAI.2025.3559053.

[6] Y. Koren, R. Bell, and C. Volinsky, "Matrix Factorization Techniques for Recommender Systems," *IEEE,* vol. 40, no. 8, pp. 42–49, 2009, doi: 10.1109/MC.2009.263

[7] F. Colace, D. Conte, M. De Santo, M. Lombardi, D. Santaniello, and C. Valentino, "A content-based recommendation approach based on singular value decomposition," *Conn Sci*, vol. 34, no. 1, pp. 2158–2176, 2022, doi: 10.1080/09540091.2022.2106943.

[8] J. Wu, J. Xu, L. Yang, and Y. Chen, "IT-PMF: A Novel Community E-Commerce Recommendation Method Based on Implicit Trust," *Mathematics*, vol. 10, no. 14, pp. 1-16, Jul. 2022, doi: 10.3390/math10142406.

[9] Y. Yan and C. Fu, "Transforming Movie Recommendations with Advanced Machine Learning: A Study of NMF, SVD, and K-Means Clustering," *Proc. IEEE ISCTIS*, vol. 2024, no. 1, pp. 178-181, 2024, doi: 10.1109/ISCTIS63324.2024.10698876

[10] I. Saifudin and T. Widiyaningtyas, "Systematic Literature Review on Recommender System: Approach, Problem, Evaluation Techniques, Datasets," *IEEE Access*, vol. 12, no. 1, pp. 19827–19847, 2024, doi: 10.1109/ACCESS.2024.3359274.

[11] W. Pan, "A Survey of Transfer Learning for Collaborative Recommendation with Auxiliary Data," *Neurocomputing,* vol. 177, no. 1, pp. 447–453, 2016, doi: 10.1016/j.neucom.2015.11.059.

[12] Y. Koren, "Collaborative filtering with temporal dynamics," *Commun ACM,* vol. 53, no. 4, pp. 72-89, 2010, doi: 10.1145/1721654.1721677.

[13] Y. Hu, C. Volinsky, and Y. Koren, "Collaborative filtering for implicit feedback datasets," Proceedings - IEEE International Conference on Data Mining, ICDM, vol. 2008, no. December 2008, pp. 263–272, 2008, doi: 10.1109/ICDM.2008.22.

[14] G. Dror, N. Koenigstein, and Y. Koren, "Yahoo! Music Recommendations: Modeling Music Ratings with Temporal Dynamics and Item Taxonomy," in *Proc. 5th ACM Conf. Recommender Syst. (RecSys '11)*, vol. 2011, no. 1, pp. 165–172, 2011, doi: 10.1145/2043932.2043964

[15] C. Wang and D. M. Blei, "Collaborative Topic Modeling for Recommending Scientific Articles," in *Proc. 17th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD '11)*, vol. 2011, no. 1, pp. 448–456, 2011, doi: 10.1145/2020408.2020480.

[16] H. Wang, N. Wang, and D.-Y. Yeung, "Collaborative Deep Learning for Recommender Systems," in *Proc. 21st ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD '15), Sydney, Australia*, vol. 2015, no. 1, pp. 1235–1244, 2015, doi: 10.1145/2783258.2783273.

[17] P. G. Duran, A. Karatzoglou, J. Vitrià, X. Xin, and I. Arapakis, "Graph Convolutional Embeddings for Recommender Systems," *arXiv preprint, arXiv:2103.03587, vol. 2021, no. 1, pp. 1-12, 2021*. [Online]. Available: http://arxiv.org/abs/2103.03587

[18] G. Ling, M. R. Lyu, and I. King, "Ratings meet reviews: a combined approach to recommend," in *Proc. 8th ACM Conf. Recommender Syst. (RecSys '14)*, vol. 2014, no. 1, pp. 105–112, 2014, doi: 10.1145/2645710.2645728.

[19] F. M. Harper and J. A. Konstan, "The MovieLens Datasets: History and Context," *ACM Trans. Interact. Intell. Syst*., vol. 5, no. 4, pp. 19:1–19:19, Dec. 2015, doi: 10.1145/2827872.

[20] S. Ge and X. Ge, "An SVD-based Collaborative Filtering approach to alleviate cold-start problems," in *Proc. 9th Int. Conf. Fuzzy Syst. Knowl. Discovery (FSKD '12)*, vol. 2012, no. 1, pp. 1474–1477, 2012, doi: 10.1109/FSKD.2012.6233900.

[21] R. M. Bell, Y. Koren, P. Ave, and F. Park, "Scalable Collaborative Filtering with Jointly Derived Neighborhood Interpolation Weights," in *Proc. 2007 ICDM Workshops (KDD Cup and Workshop), Omaha, NE, USA*, vol. 2007, no. 1, pp. 43–52, 2007, doi: 10.1109/ICDMW.2007.90.

[22] H. Wang, N. Wang, and D.-Y. Yeung, "Collaborative Deep Learning for Recommender Systems," in *Proc. 21st ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD '15), New South Wales, Australia*, vol. 2015, no. 1, pp. 1235–1244, 2015, doi: 10.1145/2783258.2783273.

[23] J. Wei, J. He, K. Chen, Y. Zhou, and Z. Tang, "Collaborative filtering and deep learning based recommendation system for cold start items," *Expert Syst. Appl*., vol. 69, no. Mar., pp. 1339–1351, Mar. 2017, doi: 10.1016/j.eswa.2016.09.040.

[24] D. Kim, C. Park, J. Oh, S. Lee, and H. Yu, "Convolutional Matrix Factorization for Document Context-Aware Recommendation," in *Proc. 10th ACM Conf. Recommender Syst. (RecSys '16)*, vol. 2016, no. 1, pp. 233–240, 2016, doi: 10.1145/2959100.2959165.

[25] R. Salakhutdinov and A. Mnih, "Bayesian Probabilistic Matrix Factorization using Markov Chain Monte Carlo," in *Proc. 25th Int. Conf. Mach. Learn. (ICML '08),* vol. 2008, no. 1, pp. 880–887, 2008, doi: 10.1145/1390156.1390267.

[26] S. Purushotham, Y. Liu, and C.-C. J. Kuo, "Collaborative topic regression with social matrix factorization for recommendation systems," in *Proc. 29th Int. Conf. Mach. Learn. (ICML), Edinburgh, Scotland, UK*, vol. 2012, no. June,, pp. 759–766.

[27] Q. Wang, B. Peng, X. Shi, T. Shang, and M. Shang, "DCCR: Deep collaborative conjunctive recommender for rating prediction," *IEEE Access*, vol. 7, no. 1, pp. 60186–60198, 2019, doi: 10.1109/ACCESS.2019.2915531.

[28] Hanafi and B. M. Aboobaider, "Word sequential using deep LSTM and matrix factorization to handle rating sparse data for e-commerce recommender system," *Comput. Intell. Neurosci*., vol. 2021, no. 1, pp. 1–12, 2021, doi: 10.1155/2021/8751173.

[29] Hanafi, "Enhance Rating Prediction for E-commerce Recommender System Using Hybridization of SDAE, Attention Mechanism and Probabilistic Matrix Factorization," *International Journal of Intelligent Engineering and Systems,* vol. 15, no. 5, pp. 427–438, 2022, doi: 10.22266/ijies2022.1031.37.

[30] Hanafi, N. Suryana, and A. S. H. Basari, "Dynamic convolutional neural network for eliminating item sparse data on recommender system," *Int. J. Adv. Intell. Inform*., vol. 4, no. 3, pp. 200–207, Nov. 2018, doi: 10.26555/ijain.v4i3.291.

[31] Hanafi, N. Suryana, and A. S. H. Basari, "Involve Convolutional-NN to generate item latent factor consider product genre to increase robustness in product sparse data for e-commerce recommendation," *J. Phys. Conf. Ser*., vol. 1201, no. 1, pp. 012004:1–8, 2019, doi: 10.1088/1742-6596/1201/1/012004.

[32] Hanafi, N. Suryana, and A. S. B. H. Basari, "Convolutional-NN and word embedding for making an effective product recommendation based on enhanced contextual understanding of a product review," *Int. J. Adv. Sci. Eng. Inf. Technol.*, vol. 9, no. 3, pp. 1063–1070, 2019, doi: 10.18517/ijaseit.9.3.8843.

[33] J. Liu and D. Wang, "PHD: A probabilistic model of hybrid deep collaborative filtering for recommender systems," in *Proc. Asian Conf. Mach. Learn. (ACML), Seoul, South Korea*, vol. 2017, no. Nov., pp. 224–239, 2017.

[34] B. Zhang, H. Zhang, X. Sun, G. Feng, and C. He, "Integrating an attention mechanism and convolution collaborative filtering for document context-aware rating prediction," *IEEE Access*, vol. 7, no. 1, pp. 3826–3835, 2019, doi: 10.1109/ACCESS.2018.2887100.

[35] Z. Zhang, Q. Wu, Y. Zhang, and L. Liu, "Movie recommendation model based on probabilistic matrix decomposition using hybrid AdaBoost integration," *PeerJ Comput. Sci.*, vol. 9, no. 1, pp. e1338:1–19, 2023, doi: 10.7717/peerj-cs.1338.

[36] C. Feng, J. Liang, P. Song, and Z. Wang, "A fusion collaborative filtering method for sparse data in recommender systems," *Inf. Sci. (N. Y.)*, vol. 521, no. 1, pp. 365–379, May 2020, doi: 10.1016/j.ins.2020.02.052.

[37] H. Hanafi, A. Pranolo, Y. Mao, T. Hariguna, L. Hernandez, and N. F. Kurniawan, "IDSX-Attention: Intrusion detection system (IDS) based hybrid MADE-SDAE and LSTM-Attention mechanism," *Int. J. Adv. Intell. Inform.*, vol. 9, no. 1, pp. 121–135, 2023, doi: 10.26555/ijain.v9i1.942.

[38] Hanafi, A. H. Muhammad, I. Verawati, and R. Hardi, "An intrusion detection system using SDAE to enhance dimensional reduction in machine learning," *J. Inform. Visualization (JOIV)*, vol. 6, no. 2, pp. 371–377, 2022, doi: 10.30630/joiv.6.2.990.

[39] Hanafi, N. Suryana, and A. S. B. H. Bashari, "Paper survey and example of collaborative filtering implementation in recommender system," *J. Theor. Appl. Inf. Technol.*, vol. 95, no. 16, pp. 3821–3832, 2017.

[40] Hanafi, N. Suryana, and A. S. H. Basari, "Deep contextual of document using deep LSTM meet matrix factorization to handle sparse data: Proposed model," *J. Phys. Conf. Ser.*, vol. 1577, no. 1, pp. 012002:1–8, 2020, doi: 10.1088/1742-6596/1577/1/012002.

[41] A. Vaswani et al., "Attention is all you need," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 30, no. Dec., pp. 5998–6008, 2017.

[42] S. Mhammedi, N. Gherabi, H. E. Massari, Z. Sabouri, and M. Amnai, "A highly scalable CF recommendation system using ontology and SVD-based incremental approach," *Bull. Electr. Eng. Inform.*, vol. 12, no. 6, pp. 3768–3779, 2023, doi: 10.11591/eei.v12i6.6261.

[43] R. Salakhutdinov and A. Mnih, "Probabilistic matrix factorization," *in Adv. Neural Inf. Process. Syst. (NeurIPS),* vol. 20, no. Dec., pp. 1257–1264, 2007.

[44] D. Kim, C. Park, J. Oh, and H. Yu, "Deep hybrid recommender systems via exploiting document context and statistics of items," *Inf. Sci. (N. Y.),* vol. 417, no. 1, pp. 72–87, Nov. 2017, doi: 10.1016/j.ins.2017.06.026.