

# Similarity handling of sparse rating

*by* Rifki Perwira

---

**Submission date:** 20-Oct-2022 12:24PM (UTC+0700)

**Submission ID:** 1930339561

**File name:** Inass\_Hanafi\_Asrani\_Rifki.pdf (1.07M)

**Word count:** 6138

**Character count:** 34162



2

## Handling Sparse Rating Matrix for E-commerce Recommender System Using Hybrid Deep Learning Based on LSTM, SDAE and Latent Factor

Hanafi<sup>1\*</sup>    Eli Pujastuti<sup>1</sup>    Arif Laksito<sup>1</sup>    Richki Hardi<sup>2</sup>    Rifki Perwira<sup>3</sup>  
 Arief Arfriandi<sup>4</sup>    Asroni<sup>5</sup>

<sup>1</sup>Universitas Amikom Yogyakarta, Jl. Ringroad Utara, Condongcatur, Depok, Sleman, Yogyakarta, Indonesia

<sup>2</sup>Universitas Mulia, Jl. ZA Maulani No. 9 Damai Bahagia, Balikpapan, Kalimantan Timur, Indonesia

<sup>3</sup>Universitas "UPN" Veteran Yogyakarta, Jl. SWK Jl. Ringroad Utara, Condongcatur, Depok, Sleman, Yogyakarta, Indonesia

<sup>4</sup>Universitas Negeri Semarang, Sekaran, Gunung Pati, Semarang, Jawa Tengah, Indonesia

<sup>5</sup>Universitas Muhammadiyah Yogyakarta, Jl. Brawijaya, Geblagan, Tamantirto, Kasihan, Bantul, Yogyakarta, Indonesia

\*Corresponding author's Email: hanafi@amikom.ac.id

**Abstract:** E-commerce is the most essential application for conducting business transactions. Delivering product information to customers require an essential machine called recommender system. Recommender systems have been adopted in many large e-commerce companies such as Amazon, e-Bay, Alibaba, YouTube, iTunes, and so on. Ratings have become an essential factor to calculate product information. They are users' expressions about their satisfaction regarding a product or service. Unfortunately, the number of ratings is extremely sparse. Generating rating prediction is a major issue in the recommender system research field. The most popular model using latent factor or matrix factorization to generate rating prediction faced the problem in accuracy performance. This research aimed to develop a novel model to generate rating prediction using two deep learning variants based on Stack Denoising Auto Encoder (SDAE), Long Short Term Memory (LSTM), and combining with a latent factor model based on Probabilistic Matrix Factorization (PMF). This study considered integrated information resources, including user information and document product information. Following the experiment report involved in Movielens and Amazon Information Video dataset, our model outperformed previous works using PMF, Collaborative Deep Learning (CDL), Probabilistic Hybrid Deep Learning (PHD) and LSTM-PMF model, with more than 5% in average using Root Mean Square Error (RMSE) evaluation metrics.

**Keywords:** Deep learning, Latent factor, Recommender system, SDAE, LSTM.

### 1. Introduction

In the mid 90's, when Amazon's e-commerce business began under its founder Jeff Bezos, it pioneered the recommender system, a system intelligence for e-commerce services, which can calculate the suitable product information to customer of customer candidate. The availability of relevant product information impacts e-commerce customers' loyalty. On the other hand, recommender systems are essential in increasing the business target revenue [1].

<sup>1</sup> The adoption of recommender systems by e-

commerce services has proven to be a success in increasing marketing targets. For example, YouTube achieved 40 % of viewers, Netflix achieved 60 % of their marketing targets, and iTunes increased 20 % of its target sales. According to Bezos, "[i]f we have 3 million items, we also require 3 million salesman" [2].

Recommender systems obtain the suitable product information for customers using their previous activity, such as ratings. This popular model is called model based or also famously named as the latent factor model. However, majority of e-commerce customers are lazy when it comes to give a rating after their deal transaction in an e-commerce

				
Hanafi	3	?	5	?
Mel	?	4	?	?
Bert	2	?	?	5
Yana	?	?	3	?
Siti	2	4	?	3

Figure. 1 Product sparse rating matrix

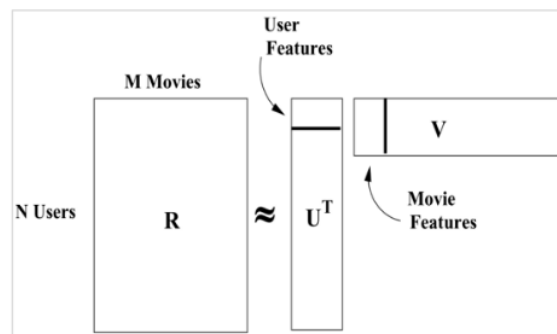


Figure. 2 Basic completing matrix mechanism to handle sparse rating using latent factor

service. This is the main reason for sparse data in e-commerce problem territory [3].

In the early latent factor model established using the matrix factorization variant, they considered employing only rating as the main factor to produce rating prediction. The major latent factor model includes the singular value decomposition (SVD) proposed by Sarwar [4], non negative matrix factorization (NMF) [5], probabilistic matrix factorization (PMF) proposed by Salahudinov [6], SVD++, and TimeSVD++ proposed by Koren [7]. According to Koren [8], sparse data is difficult to handle due to the missing extremely sparse. If the extreme sparse condition forces to train, it impacts over fitting. Fig. 1 illustrates sparse data due to minimum customer ratings. The red sign represents the missing value cause by unknown rating.

Koren proposed a new concept involving the temporal effect of time stamp rating information called TimeSVD++ [7]. The time stamp represents users' time when they put gave a rating in the system. Koren claimed that using the time stamp variable succeeded in improving the performance of traditional SVD. Koren also proposed integrating user reviews of products into matrix factorization [9]. Matrix factorization handles sparse ratings by

rotating, reducing, and completing them to produce a dense matrix.

Matrix factorization is the most essential factor in a model based on product recommendation. It was first popularized by Koren when Netflix competition was held [10]. The majority of the researchers and academicians moved to the latent factor model based on matrix factorization from memory based using neighborhood due to matrix factorization's benefits, such as its capability to integrate implicit feedback and external information. The matrix factorization mechanism to generate rating prediction is illustrated in Fig. 2.

For example,  $N$  denotes user representation and  $M$  movie representation. The rating value is represented from 1 to  $K$ .  $R_{ij}$  represents user  $i$  for movie  $j$ .  $U \in R^{D \times N}$  and  $V \in R^{D \times M}$  are the latent representations of the movie and the user feature matrices. The value of rating user  $i$  on item  $j$  can be computed as follows:

$$\hat{R} = U_i^T V_j = \sum_k U_{ik} V_{jk}$$

Following the probabilistic point of view, the user and movie features' corresponding rating formula is as follows:

$$p(R_{ij}|U_i, V_j, \sigma^2) = \mathcal{N}(R_{ij}|U_i^T V_j, \sigma^2)$$

Traditional latent factor model using matrix factorization such as PMF, SVD, NMF, SVD++, TimeSVD++ faced several essential problems. The major problem is the performance of the machine degrade significantly when applied in sparse data matrix. The main reason of sparse data matrix due to minimum of rating number from customers. According to datasets review, Movielens collect only less than 4 % rating, Amazon as the most popular datasets for recommender system study only collect less than below 1 % rating [11][12].

Many experts and researchers have attempted to handle the essential problems as mentioned above, with majority of them considering product feature information such as product categories, which included image product feature [13], acoustic product characteristic for music recommendation [14], product document information for news recommendation [15], and information tags and keywords of product information [16]. Ling proposed a novel model that integrates the document of the product into matrix factorization. This method calculates the product review document information using the traditional bag-of-word model [17]. Their experiment report achieved a significantly better performance compared to the traditional latent factor based on matrix factorization. Unfortunately, it failed to general sufficient contextual understanding in order to fully understand the document.

The hybrid model involving matrix factorization and product review document representation has become the popular approach. Wang [18] proposed a new model involving document product representation and latent factor. The framework model popular as collaborative topic regression (CTR). CTR implemented probabilistic matrix factorization (PMF) that is responsible to produce rating prediction. It also used latent dirichlet allocation (LDA) to capture the document context representation, where this method used a statistical approach to detect the context of the document. Finally, the LDA approach succeeded in increasing the effectiveness of latent factor in generating rating prediction compare over traditional models on based latent factor such as PMF. Similar to previous work, exploiting traditional machine-learning based on statistical approach was popular in the early decades. They are responsible for interpreting product document representation. Koren [17] proposed new model considering the bag-of-word (BoW) model to capture the document's contextual meaning. After, they integrated with model based using matrix factorization. However, the adoption of statistical

approach to understand documents is still facing the challenge of understanding the context of the document.

In the last decade, deep learning has become the famous method that successfully implemented and achieved tremendous result in the realm of computer science research, such as image processing, sentence classification, and voice recognition [19][20]. Several researcher specifically in the recommender system field are trying to adopt the deep learning approach to their model, such as collaborative deep learning (CDL) [21] and multi layer perceptron (MLP) [22], CDL adopted stack denoising auto encoder (SDAE) aims to extract item document representation. As we know, SDAE is a feed forward neural network that is very useful in handling missing values. The output of the product review document representation calculated by SDAE becomes the input of item representation that is then integrated into collaborative filtering based on matrix factorization.

Similar to CDL, another deep hybrid collaborative filtering, which adopted the auto encoder (AE) model for collaborative filtering. Wei attempted to combine the different latent factor model with the popular SVD model and called it hybrid collaborative deep learning (HCDL). This is the basic mechanism of HCDL: AE is responsible for capturing the product document information, and SVD is responsible for rating prediction. In this experiment, they executed the deep learning model into Netflix's dataset and combined it with IMDB dataset that contained movie reviews. HCDL was successful as it demonstrated enhanced performance over traditional latent factor models such as SVD, SVD++, PMF, and CTR. The researchers believed that HCDL's performance was largely because of SDAE's understanding of product reviews including the contextual knowledge of documents in the recommender system using a collaborative filtering method called ConvMF was proposed by Kim Dong [23]. ConvMF aimed to adopt a convolutional neural network (CNN) model in sentence classification that was proposed by Kim [24] as a state-of-the-art model for document classification.

The majority of the researchers believed that additional information could help increase the effectiveness of rating prediction. Xinhua proposed a novel information model involving social media reviews integrated into matrix factorization based on PMF. They also applied CNN to capture product document understanding [25]. This model is called social review enhanced convolutional matrix factorization (SRCMF). The method integrates users' social media activity into the matrix vector. The Yelp

Table 1. The previous work of collaborative filtering hybridization model

Ref.	Method	Model based / latent factor		Explicit feedback	Auxiliary information		BoW (LDA)	Deep learning		
		SVD	PMF	Rating	Item side document	User side information		AE	CNN	LSTM
[6]	PMF	-	√	√	-	-	-	-	-	-
[17]	RMR	-	√	√	√	-	√	-	-	-
[30]	CDL	-	√	√	√	-	√	-	-	-
[31]	LSTM-PMF	-	√	√	√	-	-	-	-	√
[32]	HCDR	√	-	√	√	-	-	√	-	-
[29]	CONVMF	-	√	√	√	-	-	-	√	-
[25]	SRCMF	-	√	√	√	√	-	-	√	-
[26]	PHD-PMF	-	√	√	√	√	-	√	√	-
[33]	BERT-LSTM	-	-	-	√	√	-	-	√	√
Our model (DDL-PMF)			√	√	√	√	-	√	-	√

dataset was chosen to represent users' social media activity. CNN was used to capture users' past social media behavior and integrate it into the latent factor model based on matrix factorization using the PMF approach. The SRCMF model succeeded in enhancing the performance of the previous models that used ConvMF.

In the CNN proposed by Liu, user implicit feedback information, captured using SDAE, is combined with product information. The hybrid of dual deep learning is called PHD-PMF [26]. The model inspired three state-of-the-art past models including SDAE [27], CNN [28] [29], and PMF [6]. PHD-PMF, implemented on popular real datasets, including MovieLens, and its movie review documents were obtained by Amazon. SDAE is responsible for handling missing values on user representation in terms of demographic information. The detailed explanation model of several state-of-the-art models are shown below in Table 1.

In this research, we adopted a different model, named dual deep learning and probabilistic matrix factorization (DDL-PMF), aiming to handle the major problems concerning sparse data. Dual deep learning is a hybrid of SDAE and long short-term memory (LSTM). Our model inspired the PHD-PMF model, which combines SDAE, CNN, and PMF [26]. This research makes four major contributions:

Novel model to capture product document representation using word embedding (Glove) and LSTM to enhance a deeper understanding of the contextual meaning of a product document.

Novel latent factor model framework that uses word embedding and LSTM to calculate product document representation and SDAE to calculate user demographical information representation; it is combined with the traditional latent factor model

based on PMF.

We conducted experiments on MovieLens datasets, including ML1M and Amazon information video (AIV) as real datasets to observe the effectiveness of our framework in big data application.

This paper is structured as follows: section 2 offers a short review of the latent factor model based on collaborative filtering using PMF, SDAE, word embedding, and LSTM. Section 3 explains the proposed framework and detail learning mechanism. Section 4 describes the experiment scenario and experiment report and discusses the results. Section 5 offers a comprehensive explanation of the findings and suggestions for future research.

## 2. Material and method

Began in the 2006 Netflix competition to achieve better performance, model-based collaborative filtering, also popularly called latent factor model, has attained excellent accuracy. The majority of them applied matrix factorization. A state-of-the-art model in collaborative filtering capable of excellent accuracy is PMF, which claimed the enhancement of SVD. To offer recommendations, PMF only considers product rating [6].

The majority of enhancement of model-based collaborative filtering uses side information of the product and user. ConMF is a state-of-the-art model that only considers product document information and implements PMF and CNN [29]. Another state-of-the-art model of collaborative filtering is PHD-PMF [26]. They adopted SDAE to represent user information; CNN was responsible for interpreting the document of the product representation, while PMF was responsible for generating rating prediction.

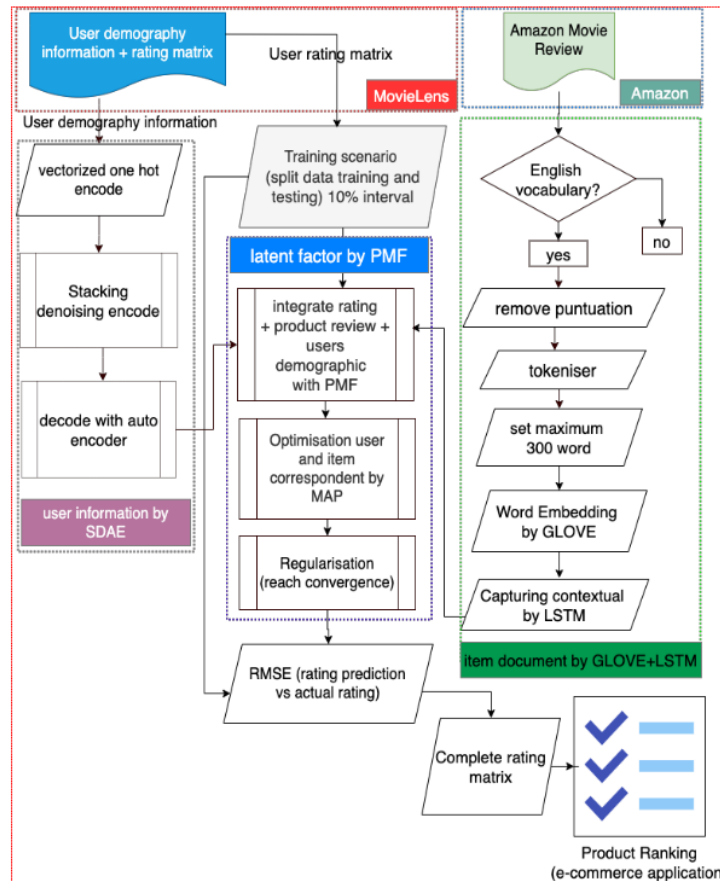


Figure. 3 The proposed DDL-PMF framework

The product documents play an important role in enhancing the quality of recommendations. The level of understanding of the product document is an essential factor in the effectiveness of the recommendation. CNN, though useful to enhance document understanding, has a drawback in capturing the context of the document. The document context be captured by considering the word order [34]. The variant of deep learning platform based on LSTM were chosen due to LSTM’s performance in word order detection. Our model involves SDAE, LSTM, and PMF and is called dual deep learning and probabilistic matrix factorization (DDL-PMF). The details of the proposed DDL-PMF and material are shown in Fig. 3.

### 2.1 LSTM

LSTM is an enhanced variant of recurrent neural network (RNN). RNN is advance of enhancement traditional feed-forward artificial neural network. The basic work mechanism is different from

traditional feed-forward neural network; RNN own inter cyclic relation on the prior and past process to create them useful in sequences aspect. We supposed a scenario model, where a sequence of input, the second section with the hidden sequence vector, and the third section with the output vector sequence by  $X, H,$  and  $Y$ . The input sequence section is represented by  $X=(x_1,x_2,x_3,x_4,x_5,\dots,x_n)$ ; and the hidden vector sequence can be calculated by  $H=(h_1,h_2,h_3,h_4,h_5,h_5,\dots,h_n)$ , and the output vector sequence can be obtained by  $Y=(y_1,y_2,y_3,y_4,y_5,y_6,\dots,y_t)$ , where  $t=1$  to  $T$  given by Eqs. (1) and (2).

$$h_t = \sigma_h(W_h x_t + U_h h_{t-1} + b_h) \tag{1}$$

$$y_t = \sigma_y W_y h_t + b_y \tag{2}$$

where  $\sigma$  is the representation of activation function,  $W$  is the representation of the weight matrix, and  $b$  is the bias factor.

The LSTM framework was first developed by

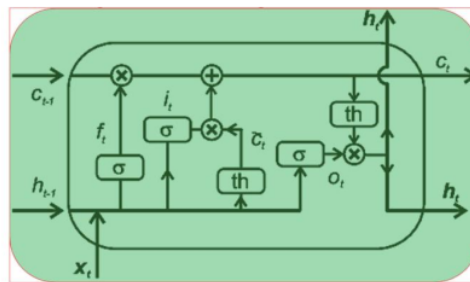


Figure. 4 Structure of a single LSTM cell

Hochreiter and Schmidhuber [35]. The basic idea of LSTM representation in a single cell is shown in Fig. 4 below and represent with Eq. (3).

where:

$x_t$ =input;  $f_t$  = forget gate;  $i_t$ =input cell;  $c_t$ =cell update;  $c_t$ =cell state;  $O_t$ =output gate;  $h_t$ =output.

$$\begin{aligned}
 i_t &= \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \\
 f_t &= \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \\
 c_t &= f_t c_{t-1} + i_t \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \\
 O_t &= \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \\
 h_t &= o_t \tanh(c_t)
 \end{aligned}
 \tag{3}$$

( $\sigma$ ) is the term of sigmoid rule; ( $i$ ) is the input gate rule; ( $f$ ) is the forget gate rule; ( $o$ ) is the output gate rule, and ( $c$ ) is the cell state rule.  $W_{ci}$   $W_{cf}$   $W_{co}$  represent the weight matrices through orifice connection. According to the LSTM framework mechanism, three gates include  $i$ ,  $f$ , and  $o$ , which are responsible for controlling the information stream. The input gate territory is responsible to calculate the input ratio. The calculating cell state step, the impact of the ratio shown on Eq. (5). The further step, the forget gate provides the prior memory  $h_{t-1}$  or not. Eq. (4) is responsible for calculating the ratio of the prior

memory. In Eq. (3), the output gate is responsible for producing the possibility output of the memory cell or not. According to the LSTM framework, we handled the vanishing and flooding gradient issue based on three gates.

In this proposed model, we applied LSTM to capture the product review document understanding to increase the effectiveness in handling sparse data matrix. A product review document is a part of a customer’s implicit feedback. It different from explicit feedback in terms of rating. LDA and CNN are the most popular approach for capturing implicit feedback documents. However, both are lacking when it comes to capturing the contextual point of view of a document since they ignore the word order and the sequential aspect of phrases. LSTM is part of a neural network that considers the sequential aspect. We designed a contextual framework mechanism including LSTM and GLOVE to capture the contextual understanding of users’ movie review feedback. The complete LSTM framework is shown in Fig. 5 below.

The LSTM framework and GIOVE have the essential task of transforming the text document into

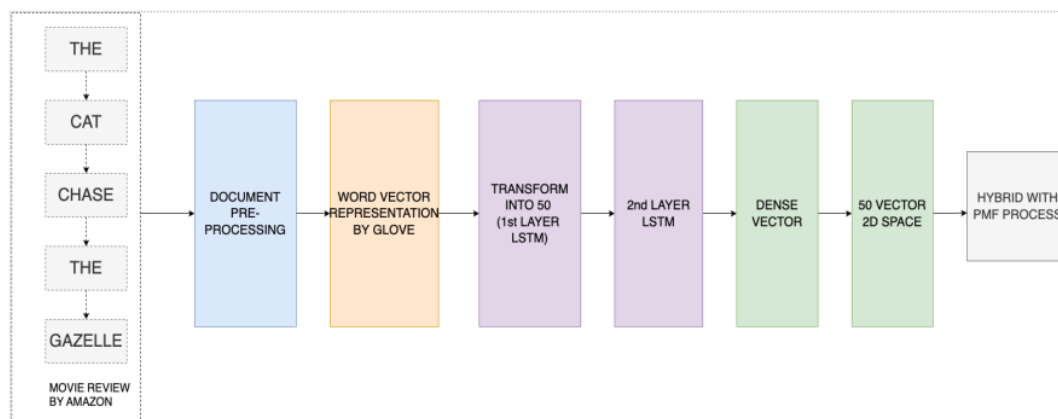


Figure. 5 The transformation of the product document into 2D vector by GLOVE and LSTM

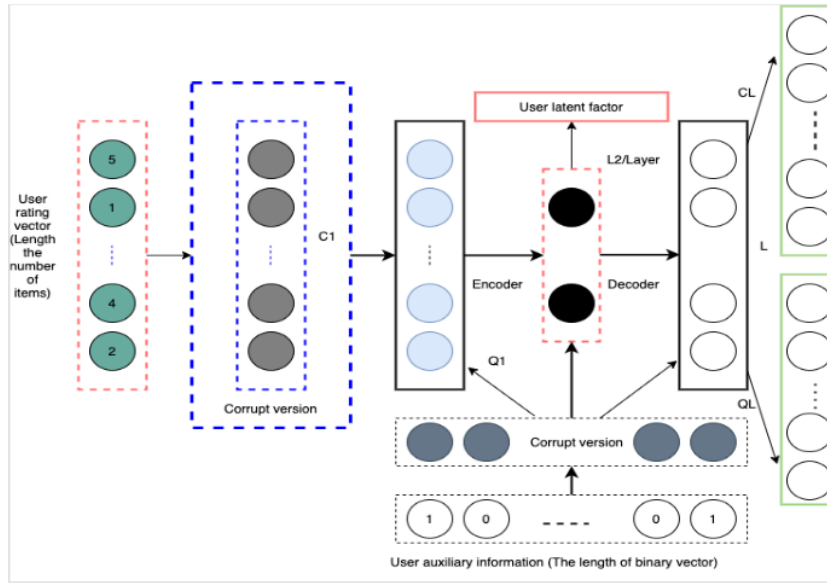


Figure. 6 User information representation using SDAE

a 2D vector space and, since they cannot directly generate rating predictions, the vector space has to be combined with the latent factor model based on PMF.

### 2.2 SDAE

SDAE was first proposed by Vincent [36] by implementing multiple jointly stacked networks aimed to produce a valuable representation of the hidden layer. This research inspired Liu [26], which combined two pieces of auxiliary information. The complete model to generate user information latent factor is show in Fig. 6. For the basic concept work, we utilized matrix U as the user representation matrix territory in the latent factor. The detailed formulation of the generative step of enhancing user latent factor is denoted by Eq. (4).

$$h_l = g(C_l h_l + Q_l \tilde{X} + b_l) \quad (4)$$

where  $C_l$  and  $Q_l$  represent the weight parameter of every section, and  $b_l$  is responsible for handling the bias vector for every layer. The symbol  $g()$  is responsible for the nonlinear activation function;  $R_i$  represents the corrupt version of  $h_0$ , and  $X$  represents the corrupt version of  $\tilde{X}$ . The output layer L can be calculated using the following Eq. (5).

$$\hat{R}_i = f(C_L h_L + b_{\hat{R}_i}) \text{ and } \hat{X} = f(Q_L h_L + b_{\hat{X}}) \quad (5)$$

$f()$  represents the second nonlinear activation

function. There are two  $\frac{L}{2}$  SDAE layers, where the first  $\frac{L}{2}$  layer represents the encoder and the second  $\frac{L}{2}$  layer represents the decoder.

### 2.3 Hybrid LSTM, SDAE & PMF

The model consists of combining three essential methods: SDAE, LSTM, and PMF. The probabilistic point of view can be described by the following Eq. (6).

$$p = (R|U, V, \sigma^2) = \prod_i^N \prod_j^M N(R_{ij}|u_i^T v_j, \sigma^2)^{I_{ij}} \quad (6)$$

We used the Gaussian normal distribution with  $N(x|\mu, \sigma^2)$  representing the probability density function. SDAE is responsible for producing users' latent factor representation, which is assumed to consist of three variables, including internal weight  $W^+$  in SDAE,  $X_i$ , which represents the side information of user i, and the varepsilon of Gaussian noise. So, the user latent factor obtained with Eq. (7) is as follows.

$$u_i = sdae(W^+, X_i) + \epsilon_i \quad (7)$$

Where  $\epsilon_i$  can be generated with the following Eq. (8).

$$\epsilon_i = N(0, \sigma_{\epsilon}^2 I) \quad (8)$$

We considered to apply the zero-mean spherical



Table 3. Characteristic datasets

Dataset category	User side information	Item side information	Number of users	Number of items	Ratings & review	Sparse level (%)
ML-1M	Gender / age / occupation / zip code	Movie descriptions	6,040	3,544	993,482	4.641%
Amazon	Demographic characteristics	Movie review	81,339	18,203	238,352	0.016%

Table 2. List of notation

notation & description		notation & description	
$U$	user representation	$V$	item representation
$\sigma^2$	variance value	$\epsilon_i$	epsilon variable of item
$\sigma_U^2$	variance value of user	$\sigma_V^2$	variance value of item
$W^+$	internal weight	$X_i$	represent side information of item
$R_{ij}$	actual value of rating	$M$	raw of movies
$N$	raw of users	$I_i$	diagonal matrix
$I_{ij}$	indicator function	$v_j$	product of item $j$
$\mu$	mean value	$\sigma$	standard deviation

Gaussian prior to Eq. (9).

$$P(W^+ | \sigma_{W^+}^2) = \prod_k N(w_k^+ | 0, \sigma_{W^+}^2) \quad (9)$$

The user latent factor can be produced by conditional distribution with Eq. (10).

$$p(U | W^+, X, \sigma_U^2) = \prod_i^N N(u_i | sdae(W^+, X_i), \sigma_U^2) \quad (10)$$

Similar to the user's latent factor, the side of the item's latent factor can be produced with the three essentials variables, the internal weight that is given by  $W$  in LSTM,  $Y_j$ , which represents item  $j$ , and varespsilon of Gaussian noise. The item latent factor can be produced by the following Eq. (11).

$$v_j = lstm(W, Y_j) + \epsilon_j \quad (11)$$

Where  $\epsilon_j$  can be obtained by the following Eq. (12).

$$\epsilon_j = N(0, \sigma_V^2 I) \quad (12)$$

The conditional distribution of the item latent factor can be given Eq. (13):

$$p(V | W, Y, \sigma_V^2) = \prod_j^M N(v_j | lstm(W, Y_j), \sigma_V^2 I) \quad (13)$$

The complete notation can be seen on Table 2 in below.

## 2.4 Dataset

To demonstrate the performance of our model in the context of generating rating predictions, we applied two datasets, MovieLens and Amazon, the most popular datasets in e-commerce recommender systems. MovieLens contains user information, the products' rating matrix, while Amazon contains the essential movie review documents. MovieLens also contains explicit expression in terms of ratings, which are represented on a value scale from 1 to 5 stars. However, MovieLens does not have item product descriptions. The Amazon dataset became the choice, and we implemented product reviews as the item document representation. The detailed characteristics of the datasets are shown below in Table 3.

## 2.5 Evaluation metrics

To investigate the performance of our model on the MovieLens datasets (ML-1M), we decided to split the dataset into three sets: a test set 10 %, a training set 80 %, and a validation set 10 %. We applied root mean squared error (RMSE) to evaluate the performance. The RMSE can be formulated as Eq. (14).

$$RMSE = \sqrt{\frac{1}{N} \sum_{i,j} Z_{i,j}^P (R_{ij} - \hat{R}_{ij})^2} \quad (14)$$

$N$  represents the total number of ratings, meanwhile  $Z_{i,j}^P$  represents the test rating.

## 3. Result and analysis

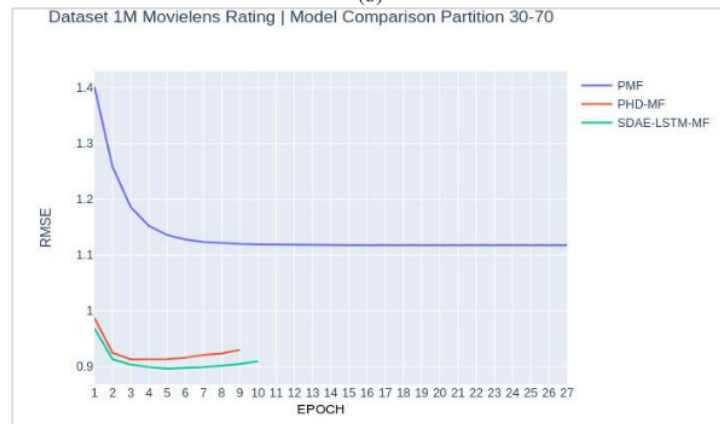
For the overall experiment and comparison method, we tried each model with specific percentages ratio from 10 % to 90 % using 10 % interval. There were thus nine experiments involving PMF, PHD-PMF, and DDL-PMF respectively. The performance was evaluated using RMSE to investigate our effectiveness model shown in Fig. 7 (Figs. a-i), where the blue color represents the PMF model, the red color represents the PHD-PMF model, and the green color represents our DDL-PMF model. The results of the training evaluation and comparison are shown in all the Fig. 7 below.



(a)



(b)



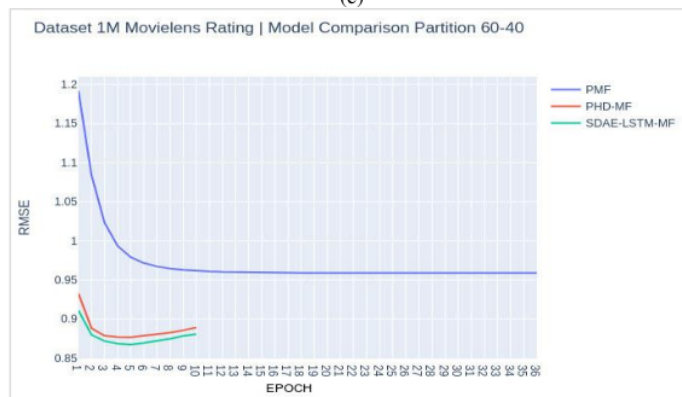
(c)



(d)



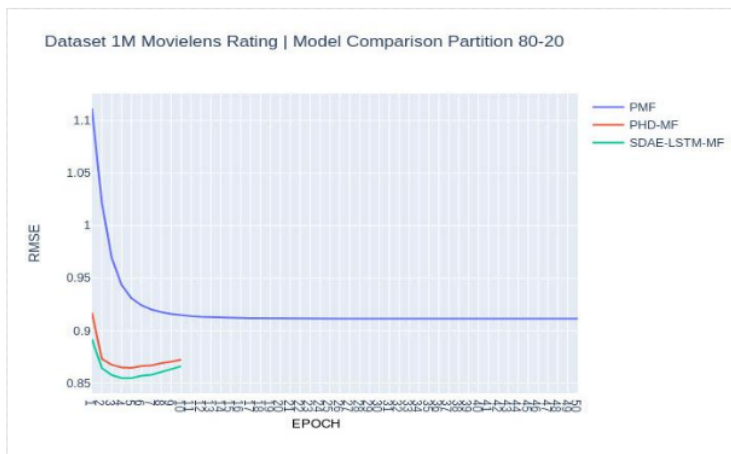
(e)



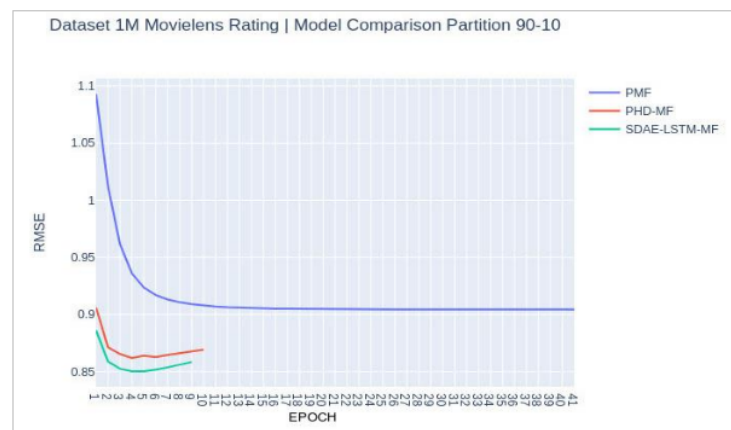
(f)



(g)



(h)



(i)

Figure. 7 RMSE evaluation result (a-i) on ML-1M and AIV

As Fig. 7 (a-i) show, there are very convincing and consistent results between PMF and its competitors. As a model without involving additional information, PMF required longer epochs to achieve maximum results. Moreover, when PMF couldn't effectively extract the latent factor, PMF achieved lower RMSE compared to additional information models.

As Fig. 8 and Table 3 show, the additional information models achieved significant performance over traditional latent factor models based on PMF. Additional information models also achieved quite significant performance when it came to generating rating prediction compared to PMF even in conditions of extreme sparseness (see 10 %, 20 %, 30 %) and when the level of sparseness was above 70 %. Additional user and item information, we believe, is adequate in an extremely sparse rating matrix.

The different levels of effectiveness among PMF, CDL, PHD-PMF, LSTM-PMF and DDL-PMF as additional information latent factor model representation was also shown. Compared to PHD-PMF, the DDL-PMF model, as it can be seen, was more effective when it came to extracting the latent factor. The essential aspect that influence the effectiveness level are explained in the following section.

### 3.1 Effect side information latent factor

PMF is popular method used to produce rating prediction without considering side information. In terms of transforming items and user data into 2D

latent factor, they only employed Gaussian normal distribution. In this research, we focused on enhancing user and item information. Employing side information, it was found, enhances the effectiveness of rating prediction; the adoption kind of machine learning also influences performance.

A different approach with PMF, both of PHD-PMF and DDL-PMF, involves SDAE to transform user information into 2D latent space. The benefit of SDAE is in its ability to extract features and transform them into 2D latent space to capture user information representation. Indeed, there is the different approach between both, where DDL-PMF involves LSTM to capture the product document representation. The difference in the effectiveness level between both determined the key success factor of LSTM, where its obtained higher share weight and was more effective in capturing the document latent factor.

As Table 4 presents, DDL-PMF outperformed PMF in each training scenario by 134 % and PMF by 14.8 % in average. Compared with PHD-PMF as a state-of-the-art recommender model system, DDL-PMF also achieved significant performance: in total 8.2 % and 0.91 % in average. Moreover, LSTM was found to have an impact in capturing product review document and an advantage in detecting the sequential aspects of phrases.

### 3.2 Effect of sequential aspect of item document

A different approach was employed between PHD-PMF, where CNN was employed to capture the

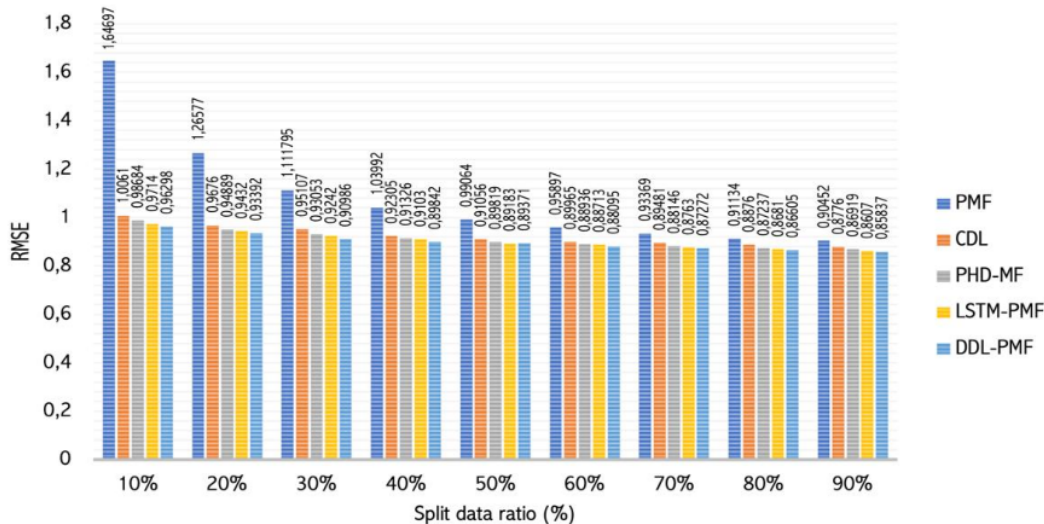


Figure. 8 Comparison result of DDL-PMF over PMF, CDL, PHD-PMF, LSTM-PMF

Table 4. The effectiveness of DDL-PMF over PMF and PHD-PMF

Ratio (%)	PMF	PHD-PMF	DDL-PMF	PMF VS DDL-PMF (improvement in %)	PHD-PMF VS DDL-PMF (improvement in %)
10%	1,64697	0,98684	0,96298	<b>41,50%</b>	<b>2,4%</b>
20%	1,26577	0,94889	0,93392	<b>26,20%</b>	<b>1,58%</b>
30%	1,11180	0,93053	0,90986	<b>18,16%</b>	<b>2,22%</b>
40%	1,03992	0,91326	0,89842	<b>13,60%</b>	<b>1,65%</b>
50%	0,99064	0,89819	0,89371	<b>9,78%</b>	<b>0,49%</b>
60%	0,95897	0,88936	0,88095	<b>8,13%</b>	<b>0,95%</b>
70%	0,93369	0,88146	0,87272	<b>6,53%</b>	<b>0,99%</b>
80%	0,91134	0,87237	0,86605	<b>4,96%</b>	<b>0,72%</b>
90%	0,90452	0,86919	0,85837	<b>5,10%</b>	<b>1,24%</b>
$\Sigma$ (Total)				<b>134%</b>	<b>8,2%</b>
$\bar{X}$ (Average)				<b>14,8%</b>	<b>0,91%</b>

contextual meaning of documents. The convolution process necessary to capture phrases. This step obtains a dimensional reduction of the sentence length. Meanwhile, DDL-PMF employed LSTM to capture the phrase where they considered the sequential aspect of sentences. Compared to the dimensional reduction mechanism based on CNN and the sequential aspect using LSTM, it can be observed that the sequential aspect is more effective to observe latent factor representation. Moreover, in conditions of extreme sparsity (see Table 3, section 10:90, 20:80, 30:70), DDL-PMF demonstrated significant performance over PHD-PMF. However, in normal sparsity data condition (see Table 3, section 50:50 to 90:10) DDL-PMF only had a slightly better performance over PHD-PMF.

Another state-of-the-art that employs enhance LSTM and PMF [31] reach good performance than previous work using deep learning approach such as CNN or AE. Unfortunately, LSTM-PMF only consider normal distribution to transform user information representation. Our model is advance model to produce better effectiveness over previous work. Our tremendous achievement can be seen on Fig. 8.

#### 4. Conclusion and future work

In this research we demonstrated a novel hybrid deep learning and latent factor model, one that incorporates SDAE and LSTM with matrix factorization. Our model succeeded in training effective user and item information with rating matrix. Our model DDL-PMF significantly outperformed recent state-of-the-art and traditional models based

on PMF. Moreover, compared to previous hybrid deep learning models, our model achieved a better performance (over 2 %) in high sparse data condition. It believed the impact of using sequential aspects to represent the contextual meaning of documents.

For future research, we will consider employing bidirectional word embedding to increase the effectiveness in capturing the contextual understanding of product documents. Bidirectional word embedding may promise a more advanced and deep understanding of the meaning of the documents. It is believed to increase the learning of latent document representation to handle sparse data matrix.

#### Conflicts of interest

The authors declare no conflict of interest on this study, authorship and publishing of this manuscript.

#### Author contribution

Hanafi, Elly Pujastuti, and Arif Laksito conveyed the idea of the system; Hanafi, Arif Laksito created algorithm and formal analysis, investigation, and data preparation; Richki Hardi prepared and wrote the original draft; Rifki Indra, Arief Arfriandi and Asroni reviewed and edited the revised draft; Hanafi and Arif Laksito analyzed and verified the research results and findings.

#### Acknowledgments

We would to thank to Universitas Amikom Yogyakarta for fully funding for this study and support with facilities and tools.

## References

- [1] F. Ricci, L. Rokach, and B. Shapira, *Recommender Systems Handbook*, 2nd ed. London: Springer New York Heidelberg Dordrecht London, 2015.
- [2] Hanafi, N. Suryana, and A. S. B. H. Basari, "An understanding and approach solution for cold start problem associated with recommender system: A literature review", *Journal of Theoretical and Applied Information Technology*, Vol. 96, No. 9, 2018.
- [3] N. Idrissi and A. Zellou, "A systematic literature review of sparsity issues in recommender systems", *Social Network Analysis and Mining*, Vol. 10, No. 1, 2020.
- [4] B. M. Sarwar, G. Karypis, J. A. Konstan, and J. T. Riedl, "Application of Dimensionality Reduction in Recommender System - A Case Study", *Architecture*, Vol. 1625, pp. 264–8, 2000.
- [5] X. Luo, M. Zhou, Y. Xia, and Q. Zhu, "An Efficient Non-Negative Matrix-Factorization-Based Approach to Collaborative Filtering for Recommender Systems", *IEEE Transactions on Industrial Informatics*, Vol. 10, No. 2, pp. 1273–1284, 2014.
- [6] R. Salakhutdinov and A. Mnih, "Probabilistic Matrix Factorization", In: *Proc. of Advances in Neural Information Processing Systems 20 (NIPS 07)*, pp. 1257–1264, 2007.
- [7] Y. Koren, "Collaborative filtering with temporal dynamics", *Communications of the ACM*, Vol. 53, No. 4, p. 89, 2010.
- [8] Y. Koren, R. Bell, and C. Volinsky, "Matrix Factorization Techniques for Recommender Systems", *IEEE*, Vol. 40, No. 8, pp. 42–49, 2009.
- [9] Y. Hu, C. Volinsky, and Y. Koren, "Collaborative filtering for implicit feedback datasets", In: *Proc. of IEEE International Conference on Data Mining, ICDM*, No. December 2008, pp. 263–272, 2008.
- [10] R. M. Bell, Y. Koren, P. Ave, and F. Park, "Scalable Collaborative Filtering with Jointly Derived Neighborhood Interpolation Weights \*", In: *Proc. of Knowledge Data Discovery (KDD) Cup and Workshop, ACM*, 2007.
- [11] 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, 2015.
- [12] J. McAuley, "Amazon Product Data", 2021. [Online]. Available: <https://jmcauley.ucsd.edu/data/amazon/>. [Accessed: 23-Aug-2021].
- [13] S. Jaradat, "Deep Cross-Domain Fashion Recommendation", In: *Proc. of the Eleventh ACM Conference on Recommender Systems - RecSys '17*, pp. 407–410, 2017.
- [14] A. V. D. Oord, S. Dieleman, and B. Schrauwen, "Deep Content-Based Music Recommendation", *NIPS*, pp. 2643–2651, 2013.
- [15] S. Purushotham, Y. Liu, and C. C. J. Kuo, "Collaborative Topic Regression with Social Matrix Factorization for Recommendation Systems", In: *Proc. of International Conference on Machine Learning 29th*, 2012.
- [16] K. Ji and H. Shen, "Addressing cold-start: Scalable recommendation with tags and keywords", *Knowledge-Based Systems*, Vol. 83, No. 1, pp. 42–50, 2015.
- [17] G. Ling, M. R. Lyu, and I. King, "Ratings meet reviews, a combined approach to recommend", In: *Proc. of the 8th ACM Conference on Recommender systems - RecSys '14*, pp. 105–112, 2014.
- [18] C. Wang and D. M. Blei, "Collaborative Topic Modeling for Recommending Scientific Articles", In: *Proc. of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '11*, p. 448, 2011.
- [19] Hanafi, N. Suryana, and A. S. B. H. Basari, "Deep learning for recommender system based on application domain classification perspective: A review", *Journal of Theoretical and Applied Information Technology*, Vol. 96, No. 14, 2018.
- [20] Hanafi, A. Pranolo, and Y. Mao, "Cae-covidx: Automatic covid-19 disease detection based on x-ray images using enhanced deep convolutional and autoencoder", *International Journal of Advances in Intelligent Informatics*, Vol. 7, No. 1, pp. 49–62, 2021.
- [21] H. Wang, N. Wang, and D. Y. Yeung, "Collaborative Deep Learning for Recommender Systems", In: *Proc. of Knowledge Discovery Data Conference*, pp. 1235–1244, 2015.
- [22] S. B. Bose and S. S. Sujatha, "Location and Time Aware Real Time Cloud Service Recommendation System Based on Multilayer Perceptron", *International Journal of Intelligent Engineering and Systems*, Vol. 14, No. 5, pp. 395–409, 2021, doi: 10.22266/ijies2021.1031.35.
- [23] D. Kim, C. Park, J. Oh, S. Lee, and H. Yu, "Convolutional Matrix Factorization for Document Context-Aware Recommendation", In: *Proc of the 10th ACM Conference on*

- Recommender Systems - RecSys '16*, 2016, pp. 233–240.
- [24] Y. Kim, “Convolutional Neural Networks for Sentence Classification”, In: *Proc of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1746–1751, 2014.
- [25] X. Wang, X. Yang, L. Guo, Y. Han, F. Liu, and B. Gao, “Exploiting Social Review-Enhanced Convolutional Matrix Factorization for Social Recommendation”, *IEEE Access*, Vol. 7, pp. 82826–82837, 2019.
- [26] J. Liu and D. Wang, “PHD: A Probabilistic Model of Hybrid Deep Collaborative Filtering for Recommender Systems”, In: *Proc. of ACML*, pp. 224–239, 2017.
- [27] F. Strub and J. Mary, “Collaborative Filtering with Stacked Denoising AutoEncoders and Sparse Inputs T”, In: *Proc. of Neural Information Processing Systems, NIPS 2016*, 2016.
- [28] 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”, *International Journal on Advanced Science, Engineering and Information Technology*, Vol. 9, No. 3, 2019.
- [29] D. Kim, C. Park, J. Oh, and H. Yu, “Deep hybrid recommender systems via exploiting document context and statistics of items”, *Information Sciences*, Vol. 417, pp. 72–87, 2017.
- [30] H. Wang, N. Wang, and D. Y. Yeung, “Collaborative Deep Learning for Recommender Systems”, In: *Proc. of Knowledge Data Discovery (KDD) Conference*, 2015.
- [31] Hanafi and B. M. Aboobaider, “Word Sequential Using Deep LSTM and Matrix Factorization to Handle Rating Sparse Data for E-Commerce Recommender System”, *Computational Intelligence and Neuroscience*, Vol. 2021, No. 1, 2021.
- [32] J. Wei, J. He, K. Chen, Y. Zhou, and Z. Tang, “Collaborative filtering and deep learning based recommendation system for cold start items”, *Expert Systems with Applications*, Vol. 69, pp. 1339–1351, 2017.
- [33] C. N. Dang, M. N. M. García, and F. D. L. Prieta, “An approach to integrating sentiment analysis into recommender systems”, *Sensors*, Vol. 21, No. 16, 2021.
- [34] C. C. Aggarwal, *Machine Learning for Text*. NEWYORK: Springer International Publishing AG, 2018.
- [35] S. Hochreiter and J. Uergen Schmidhuber, “Lstm”, *Neural Computation*, Vol. 9, No. 8, pp. 1735–1780, 1997.
- [36] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P. A. Manzagol, “Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion Pierre-Antoine Manzagol”, *Journal of Machine Learning Research*, Vol. 11, pp. 3371–3408, 2010.



# Similarity handling of sparse rating

---

## ORIGINALITY REPORT

---

4%

SIMILARITY INDEX

%

INTERNET SOURCES

%

PUBLICATIONS

4%

STUDENT PAPERS

---

## PRIMARY SOURCES

---

1

Submitted to University of Computer Studies

Student Paper

3%

---

2

Submitted to De La Salle University

Student Paper

<1%

---

3

Submitted to East West University Center for  
Research and Training

Student Paper

<1%

---

4

Submitted to Victoria University

Student Paper

<1%

---

5

Submitted to University of Hong Kong

Student Paper

<1%

---

Exclude quotes On

Exclude matches Off

Exclude bibliography On