A Deep Learning Model for Context Understanding in Recommendation Systems

Ngo Le Huy Hien[1], Luu Van Huy[2], Hoang Huu[2] Manh, Nguyen Van Hieu[2,*]
[1]Leeds Beckett University, Leeds, United Kingdom
[2]The University of Danang - University of Science and Technology, Danang, Vietnam
E-mail: n.hien2994@student.leedsbeckett.ac.uk, lvhuy@dut.udn.vn, hoanghuumanh54@gmail.com, nvhieuqt@dut.udn.vn
*Corresponding author

Keywords: recommendation system, context understanding, convolutional neural network, matrix factorization, deep learning, text processing

Received: November 2, 2022

Due to the robust growth in the amount of data and Internet users, there has been a significant rise in information overload, hindering timely access to user demand. While information retrieval systems, such as Google, Bing, and Altavista have partially addressed this challenge, prioritization and personalization of information have yet to be fully implemented. Therefore, recommendation systems are developed to resolve the issue by filtering and segmenting important information from an enormous volume of data based on different criteria such as preferences, interests, and user behaviors. By collecting data on users' interests and purchased products, the system can predict whether a particular user would enjoy an item, thus delivering an appropriate suggestion strategy. However, the increased number of Internet users and items has resulted in sparseness in increasingly vast datasets, reducing the performance of recommendation algorithms. Therefore, this study developed a model integrating Convolutional Neural Network (CNN) and Matrix Factorization (MF) to add extra product and user information, extract contexts, and add bias to the observed ratings in the training process, attempting to enhance the recommendation accuracy and context understanding. This approach can take advantage of CNN to efficiently capture an image's or document's local features, with the combination of MF to create relationships between 2 main entities, users and items. The proposed model obtained the highest RMSE of 0.93 when predicting favorable movies for 4,000 users, with an ability to learn complex contextual features and suggest more relevant content. The results are promising and can act as a reference for developing context understanding in recommendation systems, and future work may focus on optimizing the performance and developing more text-processing techniques.

Povzetek: Razvit je nov model globokega učenja, ki združuje konvolucijske nevronske mreže (CNN) in matrično faktorizacijo (MF) za izboljšanje natančnosti in razumevanja konteksta v priporočilnih sistemih.

1 Introduction

Recommendation systems (also known as recommender systems [1]) are algorithms designed to deliver suggestions for the most pertinent items to a certain user by filtering out information from a pool of data using various factors [2]. Normally, the recommendations pertain to different decision-making processes, including what movies to watch, books to read, products to buy, music to listen to, online news to read, or other products based on the desired industry [3]. Recommendation systems are substantially beneficial when a person has to pick an item from an overwhelming number of options provided by a service [4]. Netflix [5, 6] and Amazon [7], for example, employ recommendation systems to assist their consumers in choosing a suitable product or movie. The recommendation system handles a huge amount of data by filtering the most significant information from data given by a user and other criteria that correlate to their interests and preferences [3]. It determines the match between the user and the item, then infers the similarities among them for suggestions [4].

Recommendation systems have been proven to provide decent benefits to both users and supplied services. They were characterized from the standpoint of E-commerce as a tool that assists users in searching through a source of data associated with users' preferences [8]. Especially, under a complex and large accumulation of information, recommendation systems might showcase their advantage to enhance the quality of decision-making strategies [9]. This utility may result in decreasing transaction costs associated with locating and selecting products in the E-commerce sector [10]. Even in several companies, an efficient recommendation system can generate colossal revenue, and serve as a means to differ considerably from their rivals [11].

It is prevailing to apply recommendation systems when having insufficient personal knowledge or expertise with the alternatives since the systems may support and enrich the social process of making decisions based on the [9]. For instance, recommender systems are utilized in scientific libraries to assist users by enabling them to go beyond catalog searches [3]. Therefore, these types of systems can address the information overloading issue, which is commonly encountered in recent years [12], by operating accurate and efficient recommendation algorithms to deliver individualized, distinctive service and content suggestions [13].

There are several recent techniques have been developed for constructing recommendation systems, including collaborative filtering, content-based filtering, and hybrid filtering [14]. The most developed and widely used technique is collaborative filtering, which finds users who own similar preferences and utilizes their views to suggest to another user [15]. Contrarily, the content-based approach links user attributes to content resources. It hence often disregards inputs from other users and delivers recommendations solely based on the information provided by the user [16]. Notwithstanding, hybrid filtering can improve the effectiveness and accuracy of recommendation systems, by combining two or more filtering approaches in various methods. It balances out the corresponding deficiencies of different filtering techniques while using their respective strengths. The methods can be weighted, switching, cascade, mixed, feature-combination, feature-augmented, or meta-level hybrid depending on the operations of the combined techniques [17].

However, the aforementioned filtering techniques retain a few drawbacks, notwithstanding their success. Overspecialization, limited content analysis, and data scarcity are a few issues with content-based filtering algorithms. In addition, cold-start, scalability, and sparsity issues remain to exist in collaborative techniques, reducing the effectiveness of recommendations [18]. It can be seen that the common problem with such filtering techniques is data sparsity. It is because of the explosive growth in the number of users and items in the fast-growing service market, which increased the sparseness of product review data from users [19]. This sparseness diminishes the prediction accuracy of traditional filtering techniques [20].

In order to address the above data sparseness limitation, in this paper, different factors have been added to the recommendation system such as user information, user interactions, and product description documents instead of only using review data, attempting to enhance the accuracy of the system. Moreover, traditional information retrieval methods mostly use the bag-of-words model, which ignores the context information of the text document [21]. To address this, the study proposed a model to apply a Convolutional Neural Network (CNN) in the recommendation system to better understand the text document. Owing to the fact that CNN can efficiently capture local features of documents or images through local receptive fields, shared weights, and pooling [22]. However, since CNN is primarily used in classification problems, this study proposed an approach to integrate it into Matrix Factorization (MF) to define relationships between users and items. The combination makes it possible to take full advantage of both CNN and MF [23]. Inspired by the work of Donghyun and colleagues [24], this study aims to enhance the model by adding bias for the training more objectively; and supplementing extra information from description documents of both users and items. The research outcomes are promising and can be used as a reference for further developing context understanding in recommendation systems.

2 Literature review

2.1 The development of recommendation systems

Recommendation systems have gained considerable interest since their initial introduction and have been widely utilized in various sectors, including e-commerce [8], e-library [31], e-tourism [32], education [33], news [34], information retrieval, and digital content services [35]. Table 1 indicates the eminent applications of recommendation systems in different domains.

Item Type	Recommendation Systems
E-commerce Products	Amazon [7], eBay [36], Shopify, Flipkart [37]
Videos	Netflix [5], YouTube [38], Dailymotion, Hulu [39], MovieLens, Nanocrowd, Jinni [40]
Online News	Google News, Yahoo! News, BBC, New York Times [41], Findory [42], Digg, Zite [43]
Music	Spotify, Apple Music, Amazon Music, Soundcloud, Pandora, Mufin [44]
Social Networking Contents	Facebook, TikTok, Twitter, LinkedIn, Instagram [45]

Table 1: Current eminent recommendation systems in different domains

Leading e-commerce company Amazon applies a collaborative filtering technique to address scalability challenges by offline generating a table of related items using an item-to-item matrix [7]. To enhance suggestion quality, it employs topic diversity algorithms. Following that, the algorithm suggests items that are comparable online based on the customers' past purchases [46]. Thanks to this, items that are not among the shop's 100,000 best-selling items have helped Amazon gain 20% to 40% of sales [47].

Netflix Recommendation Engine uses algorithms that filter its contents using each user's unique profile. The system uses 1,300 clusters based on user choices to filter over 3,000 titles at once [48]. Cinematch, a proprietary recommendation system used by Netflix, has a root mean squared error (RMSE) of 0.9525. In 2009, Netflix held a competition called 'Netflix Prize', attempting to produce a recommender system that outperformed its algorithm, with a million-dollar prize for the winner [6]. For that reason, 60% of Netflix's DVDs are rented thanks to recommendation algorithms, and 47% of North Americans prefer Netflix with a retention rate of 93%. [49]

TikTok, one of the most popular and rapidly expanding social media networks in the world, has its secret strength as a unique recommendation system for discovering and distributing content [50]. TikTok blends videos from newbies and celebrities in the 'For You' feed, rewards high-quality creative content based on page views, and encourages emerging users to share videos with other viewers. Therefore, every user has the opportunity to become famous on the platform, regardless of their fanbase or level of popularity. High-quality creative work may be easily shared thanks to TikTok's recommendation system, which regularly suggests videos to individuals with similar interests [51].

It can be seen that recommendation systems have been applied in numerous domains and have helped businesses not only generate colossal revenue but also serve as a means to differ considerably from their competitors.

2.2 Related works

For a system to deliver its customers reliable and helpful recommendations, the usage of accurate and efficient recommendation algorithms is essential. Therefore, it is critical to clarify the advantages and limitations of various recommendation approaches. There are several recent techniques for constructing recommendation systems, which are content-based filtering, collaborative filtering, and hybrid filtering, as depicted in Figure 2.1 [14].

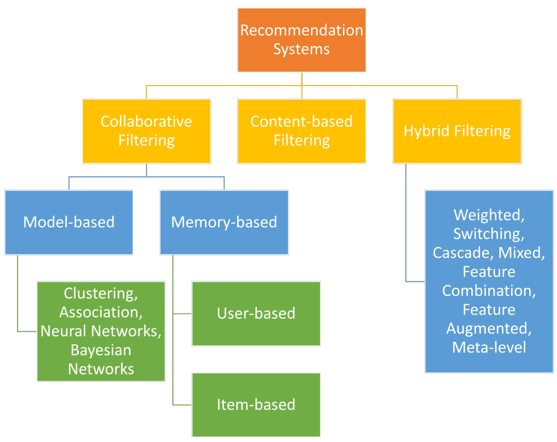

Figure 2.1: Different recommendation filtering techniques.

First of all, collaborative filtering is a technique to find users who own similar preferences and utilize their views to suggest to another user. It has become the most developed and widely used filtering technique in recommendation systems [15]. Collaborative filtering is prominent when the content cannot be accurately and simply represented by metadata, like music and movies [25]. This technique aims to build a database of user preferences for things called a user-item matrix. By comparing the commonalities between users' profiles, it connects people with shared interests and preferences in a so-called neighborhood to provide suggestions. The user then receives suggestions for unseen items that received favorable reviews from others in the neighborhood [26]. The suggestions can be in the form of recommendations or predictions. A recommendation is a list of the top items that the user would enjoy the best, whereas a prediction is an estimated favorable score of an item for the target user [27].

In contrast, content-based filtering links user characteristics to the attributes of items. It hence often disregards inputs from other users and delivers recommendations solely based on the information provided by the user [16]. This filtering technique is significant when the suggested documents can be metadata-represented, which could be books, news, and web pages. Content-based filtering extracts characteristics from the content of items previously rated by different users and then merges them into a training set. From there, the system recommends items that are greatly related to a user's favorability to them. The technique can deliver recommendations even when a user never offered ratings before [28]. As a result, users may receive suggestions without disclosing their profiles, ensuring their privacy. Furthermore, content-based filtering could handle circumstances in which different users might not have identical items, but only similar items that shared common characteristics [29].

Nevertheless, by integrating two or more filtering algorithms diversely, hybrid filtering can increase the efficacy and accuracy of recommendation systems. It compensates for the inadequacies of various filtering systems while maximizing their unique strengths [17]. Depending on the operations of the combined approaches, the methods can be weighted, switching, cascade, mixed, feature-combination, feature-augmented, or meta-level hybrid. Collaborative filtering and content-based filtering approaches can be used differently before being combined. Thereafter a unified model was formed that encompasses both content-based and collaborative filtering capabilities. Consequently, the data sparsity and cold-start issues could be solved by merging item ratings, characteristics, and demographic information [30].

Despite the success of the aforementioned filtering techniques, they come with certain drawbacks. Issues like over-specialization, limited content analysis, and data scarcity pose challenges for content-based filtering algorithms. Collaborative techniques also grapple with problems such as cold-start, scalability, and sparsity, ultimately hampering the effectiveness of recommendations [18]. A common underlying problem in these filtering techniques is data sparsity, which stems from the rapid expansion of users and items in the dynamic service market. This proliferation has increased the sparseness of product review data from users, leading to a decline in the prediction accuracy of traditional filtering methods [19, 20].

3 Methodology

To overcome the above limitation of data sparseness, this study aims to develop a model integrating Convolutional Neural Network (CNN) and Matrix Factorization (MF) to add extra product and user information and extract contexts before training, attempting to enhance the recommendation accuracy. In this section, the architecture of CNN and MF is briefly presented.

3.1 Convolutional neural network

Convolutional Neural Network (CNN/ ConvNet - proposed by Fukushima Kunihiko) is a variant of a feedforward neural network. Convolutional Neural Networks represent significant progress and influence in the development of Deep Learning [52]. Many CNN variations, including VGGNet, MobileNet, Inceptions, ResNet, RegNet, DenseNet, and EfficientNet have been developed robustly. These variants emphasize different facets of accuracy, efficiency, and scalability. The field of computer vision is mostly dominated by ConvNets models [53].

The organization of the visual cortex and the human brain's neural network both had an influence on CNN's architecture [54]. Individual neurons can only respond to stimuli in the restricted visual field region known as the Receptive Field. A succession of similar fields that overlap encompasses the entire visual field [55]. There are four main types of layers for a convolutional neural network: the convolutional layer (to extract local features), the pooling layer (representing data of the previous layer in a more concise form, i.e., select only the typical features with the highest scores through activation functions), the ReLU correction layer and the fully-connected layer [56], as indicated in Figure 3.1.

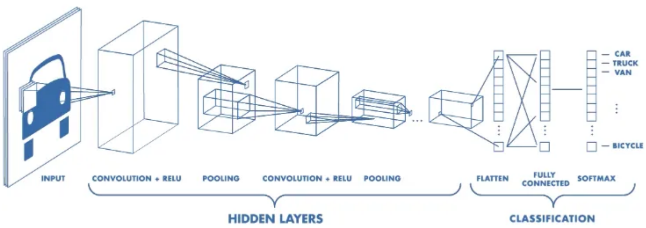

Figure 3.1: The Architecture of CNN. [57]

As shown in Figure 3.1, a CNN normally consists of two main components:

1. Hidden layers or feature extraction layers: in this component, the network will perform a series of convolution and pooling computations to detect features. For example, if an image of a zebra is inputted, in this component, the network will recognize its stripes, two ears, and four legs.
2. Classification: in this component, a class with full associations will act as a classifier of previously extracted features.

The CNN model in natural language processing often considers the local context aspect of the corpus [58]. These contexts are extracted through filters or the kernel and aggregated at the pooling layer [59]. However, since the CNN model is often used for classification problems, it is challenging to apply CNN directly to the recommendation system.

3.2 Matrix factorization

Matrix Factorization (MF) is a commonly used collaborative filtering method in recommendation systems proposed by Simon Funk [60]. Matrix Factorization decomposes the performance evaluation matrix into a product of two matrices U and V. While U represents the correlation between users, V represents the relationship between items, described in Figure 3.2.

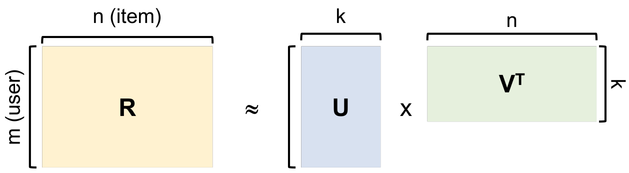

Figure 3.2: The concept of matrix factorization.

As shown in Figure 3.2, the Matrix Factorization technique involves decomposing a large matrix R into two smaller matrices U and V, such that the reconstruction of R from these smaller matrices is as accurate as possible, i.e., $R \approx U \times V^T$.

In which:

- U is a matrix of size $m \times k$, where each row represents k latent factors describing user m.
- V is a matrix of size $n \times k$, with each row being a vector comprising k latent factors describing item i (typically $k << m$ and $k << n$).
- V^T denotes the transpose matrix of V.

The key challenge in the MF technique lies in determining the values of the two parameters (matrices) U and V. These parameters are identified by optimizing an objective function. In the context of rating prediction, the objective function, denoted as L, is expounded upon in the subsequent section.

The concept of latent features that reflect the relationship between objects and users is fundamental in Matrix Factorization for Recommendation Systems. For example, in a movie recommendation system, the latent features can be criminal, political, action, comedy, etc.; may also be a combination of these features or anything that may not need to be named [61]. Each item can bring some latent features to some extent corresponding to the coefficients in its vector v. The higher the coefficient, the higher the possibility of having that feature. Similarly, each user will also tend to prefer certain latent features described by the coefficients in its vector u. The higher the coefficient, the more likely users prefer the movies with that latent feature. The value of the expression uv will be high if the corresponding components of v and u are both high. This means that the item has latent features that the user likes, thus the system recommends this item to that user.

Assume that there are m users and n items, with a user-item rating matrix R, in which R $\in R^{m\times n}$. In Matrix Factorization, latent models of user i and item j can be represented as k-dimensional models, $u_i \in R_k$ and $v_j \in R_k$. The observed rating r_{ij} of user i on item j is calculated by the inner product of respective latent models of user i and item j. A common approach to training latent models is minimizing a loss function L, which comprises sum-of-squared-error terms among the observed ratings and the predicted ratings. Therefore, the loss function in this situation can be expressed as:

$$L = \sum_{i}^{m}\sum_{j}^{n} I_{ij}(r_{ij} - u_i^T v_j)^2 + \\ + \lambda_u \sum_{i}^{m} ||u_i||^2 + \lambda_v \sum_{j}^{n} ||v_j||^2 \quad (1)$$

in which:

- I_{ij} is an indicator function that becomes 1 if user i rated item j and equals 0 if not.
- λ denotes the regularization term. When λ is excessively large, the model tends to underfit the data; conversely, if λ is overly small, the model may become overly complex, leading to overfitting. The fine-tuning of the λ value is a crucial aspect in optimizing the performance of the MF model.
- λ_u is the regularization parameter associated with user vectors u_i. Regularization serves as a technique to prevent overfitting in machine learning models. It is applied in the loss function by penalizing the squared Euclidean norm (L2 norm) of user vectors. This regularization constrains user vectors from becoming excessively large during the training process, mitigating the risk of overfitting to the training data and potentially enhancing the model's generalization ability to unseen data.
- Similarly, λ_v represents the regularization parameter for item vectors u_i. This regularization parameter is essential for preventing overfitting in the context of item vectors, analogous to its role in the regularization of user vectors (λ_u).

4 Proposed model

4.1 General architecture

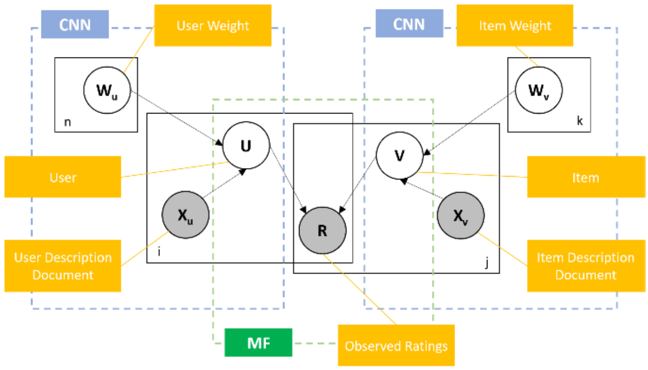

Figure 4.1: General Architecture.

As depicted in Figure 4.1, MF (Matrix Factorization, in the green box) is the decomposition of the observed rating matrix R of user-item into two matrices with lower weights. Matrix U represents the relationship between users, while matrix V represents the correlation between the items. The model aims to add product features to the recommendation system. CNN in natural language processing often considers the local context aspect of the text. Therefore, CNN is used to extract features with local contexts of the user and item description sets and then add the information to matrix U (matrix containing vectors describing characteristics of the user, such as age, gender, and occupation) and V (matrix containing vectors describing features of the item) respectively. This technique can complement and clarify the properties of the vectors in matrix U and V.

In Figure 4.1, X_u and X_v act as the set of documents describing the user and item respectively, and W_u and W_v are the weights of the CNN model for the user and item correspondingly. The outputs of the CNN are latent feature vectors of those input documents. The difference between those latent feature vectors with matrix U and V is the integration between CNN and MF in fully analyzing descriptive documents and evaluation data.

This research employs a Convolutional Neural Network (CNN) to extract local features from embedding vectors, consisting of the following layers:

- Input layer: receives embedding vectors describing product narratives with a length of 100 tokens.
- The token and position embedding layer comprises two main components:
  - Token embedding: transforms each word in the product narrative into a dense vector representation. This representation captures the semantic meaning of the word as well as its relationships with other words in the vocabulary.

  - Position embedding: encodes the position of each word in the product narrative into a vector representation. This representation helps the model understand the context of each word and its relationships with other words in the product narrative.

- The output of the embedding layer, comprising token and position information, is a sequence of embedding vectors, where each embedding vector represents a token (word) in the product narrative and incorporates its position in the product narrative.

- Subsequently, the embedding vectors are fed into a CNN layer, consisting of fundamental layers such as Convolutional, pooling, and incorporating dropout techniques to extract more complex features from the text. The CNN layer learns to identify patterns and relationships among the embedding vectors, which are then utilized to predict user rankings for different products.

Details of the CNN model architecture are illustrated in Figure 4.2.

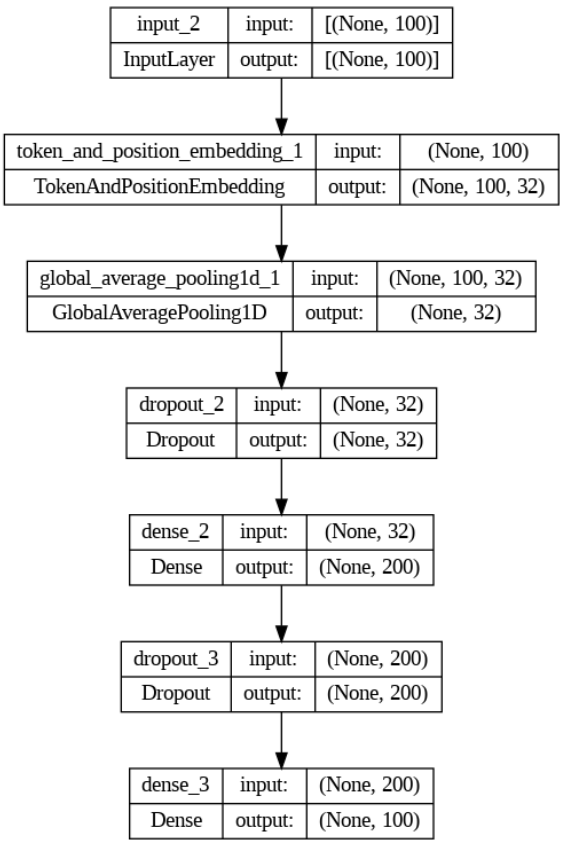

Figure 4.2: The Used CNN Architecture.

The rationale behind the utilization of this CNN structure is predicated upon the model's input being comprised of embedded vectors used to depict products, typically of relatively modest dimensionality (dim = 100). Consequently, a CNN architecture with fundamental layers, as expounded above, is employed in this study to extract local features from the embedded vector.

4.2 Adding bias

As mentioned in Section 3.2, the observed rating r_ij of user i on item j is calculated by the inner-product of respective latent models of user i and item j, which can be indicated as:

$$r_{ij} \approx \hat{r}_{ij} = u_i^T v_j \tag{2}$$

However, to avoid overfitting issues, this study adds bias to the observed rating:

$$\hat{r}_{ij} = u_i^T v_j + d_i + b_j \tag{3}$$

in which:

- d_i is a coefficient representing the pleasantness of user i. The higher the coefficient, the better the user i tends to rate the products.

- b_j is a coefficient illustrating product quality, the higher the coefficient. The more users tend to rate that product better.

4.3 Loss function

From there, the loss function now can be depicted as:

$$\begin{aligned}
L(U,V,W) &= \sum_i^m \sum_j^n \frac{I_{ij}}{2}(r_{ij} - \hat{r}_{i_j})^2 \\
&+ \frac{\lambda_U}{2} \sum_j^m ||v_j - cnn(Wv, Xv_j)||_2 \\
&+ \frac{\lambda_{Wu}}{2} \sum_k^{|w_{u_k}|} ||w_{u_k}||_2 \\
&+ \frac{\lambda_{Wv}}{2} \sum_n^{|w_{v_n}|} ||w_{v_n}||_2
\end{aligned} \tag{4}$$

The loss function is minimal when the derivative of the above equation is 0. The loss function uses coordinate descent to find the function that updates u and v. This optimizes having to iterate over and over one variable while correcting the others.

Assuming W_u, W_j, and V (or U) are constants, the above equation becomes a quadratic function with respect to U (or V). Therefore:

$$\begin{aligned}
&u_i \leftarrow (VIiV^T + \lambda_u I_k)^{-1}(VR_i + \lambda_u nn(Wu, Xu_i)) \\
&d_i \leftarrow (r_{ij} - u_i^T v_j - b_j) \\
&v_j \leftarrow (UI_jU^T + \lambda VI_k)^{-1}(UR_j + \lambda_v cnn(Wv, Xv_j)) \\
&b_j \leftarrow (r_{ij} - u_i^T v_j - d_j)
\end{aligned} \tag{5}$$

W_u and W_j will be updated through the backpropagation of the CNN.

5 Experiment and results

5.1 Dataset

This research utilizes Movielens 1M [62], a user's movie review dataset, which contains 6000 users and 4000 movies. It was released in 2003 with a rating rate of 4.6%. This dataset includes:

– Movie information: id, movie name, genre, release year;

– User information: gender, age, occupation;

– List of user reviews corresponding to movies (1 million samples).

The training was conducted on Google Colab with the configuration specified in Table 2.

Type	Specifications
CPU	Intel(R) Xeon(R) CPU @ 2.20GHz
Number of CPUs	2
RAM	12.0 GB
Memory	108.0 GB [44]
GPU	Nvidia Tesla K80

Table 2: Device Specification.

5.2 Dataset pre-processing

The input of the model is the item description document set. Particularly in this experiment, it contains 4000 movie description texts corresponding to 4000 movies in the dataset. A sample data used in the dataset is presented in Figure 5.1.

The user quantity within the dataset was partitioned for experimental purposes, comprising subsets of 1000 users, 2000 users, and so forth. This approach facilitated the evaluation of the model across varying dataset scales, allowing an examination of potential impacts. Statistics of the number of users, items, and ratings are presented in Table 3 for reference and analysis.

From the description text of the movies, latent features were extracted to add to the training model. The input text set of movie descriptions has been through different preprocessing steps, as shown in Figure 5.2, starting with cleaning to remove the noise in the text like HTML tags. The next step is word splitting, meaning splitting the sentences into single words. Those words were then normalized to the same font and type. And finally, stopwords will be eliminated, which are words that appear frequently but contain trivial meanings, such as 'is', 'that', or 'this' in English. A sample of a movie description after the pre-processing process is presented in Figure 5.3.

movie id: 39

movie name: Toy Story

description: A little boy named Andy loves to be in his room, playing with his toys, especially his doll named "Woody". But, what do the toys do when Andy is not with them, they come to life. Woody believes that his life (as a toy) is good. However, he must worry about Andy's family moving, and what Woody does not know is about Andy's birthday party. Woody does not realize that Andy's mother gave him an action figure known as Buzz Lightyear, who does not believe that he is a toy, and quickly becomes Andy's new favorite toy. Woody, who is now consumed with jealousy, tries to get rid of Buzz. Then, both Woody and Buzz are now lost. They must find a way to get back to Andy before he moves without them, but they will have to pass through a ruthless toy killer, Sid Phillips.

user rating: [5. 0. 0. ... 0. 5. 4.]

vector movie: [4289, 940, 4912, …, 4173, 396, 7352]

Figure 5.1: Sample data used in the dataset.

Number of Users	Number of Items	Number of Ratings
1000	3280	154212
2001	3452	337262
3001	3477	484775
4001	3505	660411
5001	3532	826438

Table 3: Statistics of the number of users, items, and ratings.

5.3 Training

The dataset was divided into 3 subsets, which are training, validation, and testing sets. Corresponding to each user, the number of user reviews will be divided by the ratio of 80% for the training set, 10% for the test set, and 10% for the validation set.

λU	λV	Dimension	Train. Loss	Val. Loss	Test. Loss
10	40	500	0.76	0.88	0.88
10	60	500	0.77	0.88	0.88
10	50	50	0.78	0.89	0.88
10	10	50	0.7	0.90	0.90
100	10	100	0.87	0.90	0.90
50	100	100	0.88	0.91	0.91

Table 4: Loss results in different hyperparameters.

From Table 4, it can be seen that the ratio between λU and λV significantly affects the results. If λU is much larger than λV, meaning a higher priority is given to learning the parameters of U, a good result could not be attained. While the goal of the problem is to use data from the item, it

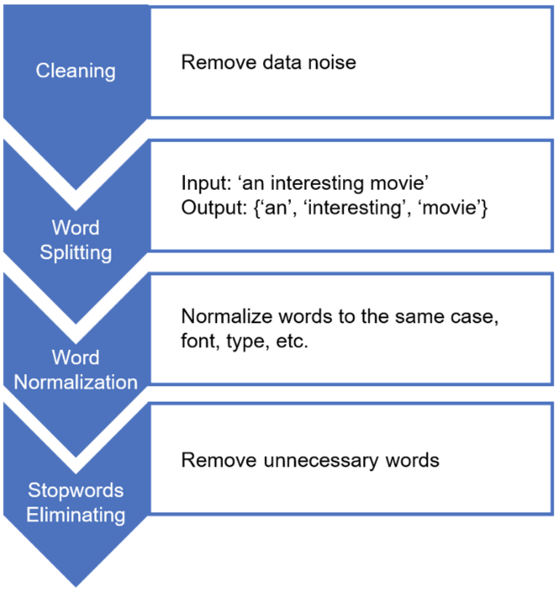

Figure 5.2: Text Pre-processing Process.

is better to give preference to λV, making it slightly higher than λU, to obtain a better result.

5.4 Evaluation

To evaluate the model's general performance, this study uses Root-mean-square error (RMSE) and mean-square error (MSE), which represent the dispersion of the predicted data relative to the actual data.

$$RMSE = \sqrt{\frac{\sum_i^m (\hat{r}_i - r_i)^2)}{m}} \tag{6}$$

$$MSE = \frac{1}{m}\sum_i^m (\widehat{r_i} - r_i)^2 \tag{7}$$

The RMSE function evaluates the results after each iteration for all 3 training, validation, and testing sets. The model training process was repeated for about 100-200 iterations until the loss function gave the smallest value on the validating and testing sets. RMSE results of the model on the training, validating, and testing sets are illustrated in Figure 5.4.

As can be seen from Figure 5.4, in the 8^{th} iteration, the results began to deteriorate, and the validation RMSE increased while the training RMSE continued to be overfitting. Therefore, the result was obtained in the 8th iteration. The evaluation of results for the entire data is shown in Table 5.

Table 5 evaluates the proposed model using two metrics: Root Mean Square Error (RMSE) and Mean Squared Error (MSE). These metrics gauge the disparity between predicted rankings and actual rankings. Based on the tabulated data, it is evident that the proposed model demonstrates strong performance on the test set, yielding an RMSE of 0.89 and MSE of 0.78. This signifies the model's ability to accurately predict user rankings for diverse products.

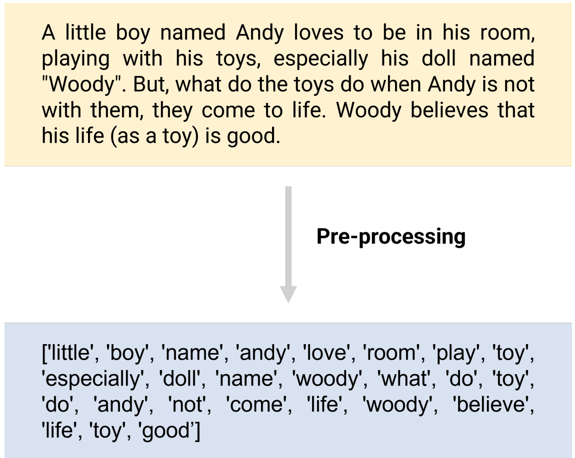

Figure 5.3: Sample Movie Description Text After Pre-processing Process.

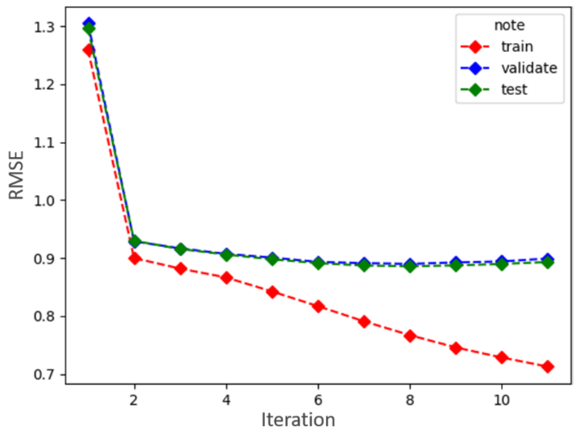

Figure 5.4: Plotting RMSE results.

The RMSE and MSE values across all three sets—training, validation, and testing—indicate that the model exhibits robust predictive capabilities on the test dataset. Both RMSE and MSE values remain stable, with minimal deviation observed between the validation and test datasets. This suggests that the model does not encounter issues related to overfitting or underfitting.

To determine how the results correlate with the user amount, a comparison of RMSE with different numbers of users is presented in Table 6.

Evaluation metric	Training	Validation	Testing
RMSE	0.76695	0.88974	0.88563
MSE	0.58821	0.79163	0.78435

Table 5: Result Evaluation in different metrics.

No. of users	Train. RMSE	Val. RMSE	Test. RMSE	Exec. time (s)	Train. time (s)
1000	0.87865	0.91478	0.90093	0.0062	110
2000	0.87205	0.91791	0.93004	0.0052	75
3000	0.87168	0.91896	0.92671	0.0053	91
4000	0.86955	0.91383	0.92973	0.005	159
5000	0.87865	0.91478	0.90093	0.0062	110

Table 6: Comparison of the RMSE with different numbers of users.

Table 6 demonstrates when increasing the number of users in the dataset, from 1000 to 5000, the accuracy increases, but with a longer convergence time. Therefore, in order to produce appropriate recommendations, recommendation system applications need to employ a large dataset.

5.5 Utilizing the training results

The results obtained after training the model are 2 matrices U and V. An evaluation matrix Y[i,j] can be generated as:

$$Y[i,j] = U[i] * V[j]^T \tag{8}$$

in which:
- i: i-th user
- j: j-th item

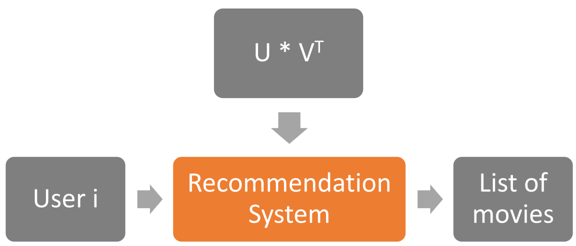

Figure 5.5: Using the training results for creating recommendations.

As depicted in Figure 5.5, the evaluation matrix can be applied in the recommendation system for further usage, which outputs a list of recommended movies for the i^{th} user.

6 Conclusions and future work

In this research, a deep learning model for recommendation systems is proposed by integrating Convolutional Neural Network and Matrix Factorization to add extra information and extract contexts before training, attempting to enhance recommendation accuracy and context understanding. Despite substantial previous efforts [21, 63, 64], this study adds additional information on both user and item description documents and applied Convolutional Neural Networks to efficiently capture their local features. Furthermore, this research adds bias to the observed ratings to avoid overfitting issues and uses Matrix Factorization to create relationships between users and items. The proposed model can be further used as a benchmark for developing context comprehension in recommendation systems, hence delivering more relevant recommendations for users.

It is observed that the model obtained a very good RMSE of 0.89 in the testing set, which means the model can relatively predict favorable movies of users accurately. Testing on different amounts of users reveals that the more users, the higher the accuracy, but the longer the convergence time. It is noted that this study subdivides the dataset to assess each subset independently, as opposed to providing a comprehensive evaluation of the entire dataset. Consequently, the rationale for refraining from comparing with other models stems from the divergence in data partitioning strategies. Hence, the evaluation process becomes inherently untenable due to the dissimilarity in data distribution methodologies across models.

Future research may aim to overcome the scant user information (e.g., hobbies, location, marital status) by looking for a large dataset with more user information, including more features in the user description documents, leading to a higher impact on the prediction. Moreover, the proposed model could be developed further by swapping out Matrix Factorization with more efficient techniques, such as singular value decomposition (SVD).

Acknowledgement

This research was funded and implemented for the Rising-Star project of the University of Science and Technology, The University of Danang, Vietnam.

References

[1] L. Lü, M. Medo, C. H. Yeung, Y.C. Zhang, Z.K. Zhang and T. Zhou (2012) Recommender systems, *Physics reports* , vol. 519, no. 1, pp. 1-49. https://doi.org/10.1016/j.physrep.2012.02.006

[2] H. Ko, S. Lee, Y. Park, A. Choi (2022) A survey of recommendation systems: recommendation models, techniques, and application fields, *Electronics*, vol. 11, no. 1, p. 141. https://doi.org/10.3390/electronics11010141

[3] F. O. Isinkaye, Y. O. Folajimi and B. A. Ojokoh (2015) Recommendation systems: Principles, methods and evaluation, *Egyptian informatics journal*, vol. 16, no. 3, pp. 261-273. https://doi.org/10.1016/j.eij.2015.06.005

[4] G. Shani and A. Gunawardana (2011) Recommender systems handbook, *Recommender systems hand-*

book, pp. 257-297. https://doi.org/10.1007/978-0-387-85820-3_8

[5] Gomez-Uribe, C. A. and N. Hunt (2015) The netflix recommender system: Algorithms, business value, and innovation, *ACM Transactions on Management Information Systems (TMIS)*, vol. 6, no. 4, pp. 1-19. https://doi.org/10.1145/2843948

[6] J. Bennett, S. Lanning (2007) The netflix prize, *in KDD cup and workshop*, p.35.

[7] B. Smith, G. Linden (2017) Two decades of recommender systems at Amazon. com, *IEEE internet computing*, , vol. 21, no. 3, pp. 12-18. https://doi.org/10.1109/MIC.2017.72

[8] S. S. Li, E. Karahanna (2015) Online recommendation systems in a B2C E-commerce context: a review and future directions, *Journal of the association for information systems*, vol. 16, no. 2, p. 2. https://doi.org/10.17705/1jais.00389

[9] K. Al Fararni, B. Aghoutane, J. Riffi, A. Sabri and A. Yahyaouy (2020) Comparative study on approaches of recommendation systems, *Embedded Systems and Artificial Intelligence*, pp. 753-764. https://doi.org/10.1007/978-981-15-0947-6_72

[10] C.-S. Juan-Pedro, I. Ramos-de-Luna, E. Carvajal-Trujillo and Á. F. Villarejo-Ramos (2020) Online recommendation systems: Factors influencing use in e-commerce, *Sustainability Publisher*, pp. 01-15. https://doi.org/10.3390/su12218888

[11] L. Ebrahimi, V. R. Mirabi, M. H. Ranjbar and E. H. Pour (2019) A customer loyalty model for e-commerce recommendation systems, *Journal of Information Knowledge Management* , vol. 18, no. 3 pp. 12-18. https://doi.org/10.1142/S0219649219500369

[12] Z. Wang, X. Yu, N. Feng and Z. Wang (2014) An improved collaborative movie recommendation system using computational intelligence, *Journal of Visual Languages Computing*, vol. 25, no. 6, pp. 667-675. https://doi.org/10.1016/j.jvlc.2014.09.011

[13] M. Robillard, R. Walker and T. Zimmermann (2009) Recommendation systems for software engineering, *IEEE software*, vol. 27, no. 4, pp. 80-86. https://doi.org/10.1109/MS.2009.161

[14] L. Shah, H. Gaudani and P. Balani (2016) Survey on recommendation system, *International Journal of Computer Applications*, vol. 137, no. 7, pp. 43-49. https://doi.org/10.5120/ijca2016908821

[15] S. K. Raghuwanshi and R. K. Pateriya(2019) Collaborative filtering techniques in recommendation systems, *Data, Engineering and Applications*, pp. 11-21. https://doi.org/10.1007/978-981-13-6347-4_2

[16] S. Eliyas and P. Ranjana (2022) Recommendation Systems: Content-Based Filtering vs Collaborative Filtering, *2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)*, pp. 1360-1365. https://doi.org/10.1109/ICACITE53722.2022.9823730

[17] P. B. Thorat, R. M. Goudar and S. Barve (2015) Survey on collaborative filtering, content-based filtering and hybrid recommendation system, *International Journal of Computer Applications*, vol. 110, no. 4, pp. 31-36. https://doi.org/10.5120/19308-0760

[18] G. Suganeshwari and S. P. S. Ibrahim (2016) A survey on collaborative filtering based recommendation system *3rd international symposium on big data and cloud computing challenges* , pp. 503-518. https://doi.org/10.1007/978-3-319-30348-2_42

[19] N. L. H. Hien, T. Q. Tien and N. V. Hieu (2020) Web crawler: Design and implementation for extracting article-like contents,*Cybernetics and Physics*, vol. 9, no. 3, pp. 144-151. https://doi.org/10.35470/2226-4116-2020-9-3-144-151

[20] P. Kumar and R. S. Thakur (2018) Recommendation system techniques and related issues: a survey *International Journal of Information Technology*, vol. 10, no. 4, pp. 495-501. https://doi.org/10.1007/s41870-018-0138-8

[21] S. Bhattacharya and L. Ankit (2019) Movie recommendation system using bag of words and scikit-learn, *Int J Eng Appl Sci Technol*, vol. 4, pp. 526-528. http://doi.org/10.33564/IJEAST.2019.v04i05.076

[22] M. Sheikh Fathollahi and F. Razzazi(2021) Music similarity measurement and recommendation system using convolutional neural networks,*International Journal of Multimedia Information Retrieval*, vol. 10, no. 1, pp. 43-53. https://doi.org/10.1007/s13735-021-00206-5

[23] A. F. Agarap(2017) An architecture combining convolutional neural network (CNN) and support vector machine (SVM) for image classification,*arXiv preprint*, p. arXiv:1712.03541. https://doi.org/10.48550/arXiv.1712.03541

[24] D. Kim, C. Park, J. Oh, S. Lee and H. Yu (2016) Convolutional matrix factorization for document context-aware recommendation,*Proceedings of the 10th ACM*

Conference on Recommender Systems " pp. 233-240. https://doi.org/10.1145/2959100.2959165

[25] B. S. Neysiani, N. Soltani, R. Mofidi and M. H. Nadimi-Shahraki (2019) Improve performance of association rule-based collaborative filtering recommendation systems using genetic algorithm,*International Journal of Information Technology and Computer Science*, vol. 11, no. 2, pp. 48-55. 10.5815/ijitcs.2019.02.06

[26] Z. Cui, X. Xu, X. U. E. Fei, X. Cai, Y. Cao, W. Zhang and J. Chen(2020), Personalized recommendation system based on collaborative filtering for IoT scenarios,*IEEE Transactions on Services Computing*, vol. 13, no. 4, pp. 685-695. https://doi.org/10.1109/TSC.2020.2964552

[27] W. Zhang and J. Wang(2018), Content-bootstrapped collaborative filtering for medical article recommendations,*2018 IEEE International Conference on Bioinformatics*, pp. 1184-1188. https://doi.org/10.1109/BIBM.2018.8621180

[28] U. Javed, K. Shaukat, I. A. Hameed, F. Iqbal, T. M. Alam and S. Luo (2021), A review of content-based and context-based recommendation systems,*International Journal of Emerging Technologies in Learning (iJET)*, vol. 16, no. 3, pp. 274-306. http://doi.org/10.3991/ijet.v16i03.18851

[29] J. Son and S. B. Kim (2017) Content-based filtering for recommendation systems using multiattribute networks,*Expert Systems with Applications*, vol. 89, pp. 404-41. https://doi.org/10.1016/j.eswa.2017.08.008

[30] A. A. Kardan and M. Ebrahimi(2013) A novel approach to hybrid recommendation systems based on association rules mining for content recommendation in asynchronous discussion groups, *Information Sciences*, vol. 219, pp. 93-110. https://doi.org/10.1016/j.ins.2012.07.011

[31] C. Tsai and M. Chen(2008) Using adaptive resonance theory and data□mining techniques for materials recommendation based on the e□library environment,*The Electronic Library*, vol. 26, no. 3, pp. 287-302. https://doi.org/10.1108/02640470810879455

[32] R. Khan, H. Ur, C. K. Lim, M. F. Ahmed, K. L. Tan and M. B. Mokhtar(2021) Systematic review of contextual suggestion and recommendation systems for sustainable e-tourism,*Sustainability*, vol. 13, no. 15, p. 8141. https://doi.org/10.3390/su13158141

[33] A. C. Rivera, M. Tapia-Leon and S. Lujan-Mora (2018) Recommendation systems in education: a systematic mapping study,*International Conference on Information Technology Systems*, pp. 937-947. https://doi.org/10.1007/978-3-319-73450-7_89

[34] C. Feng, M. Khan, A. U. Rahman and A. Ahmad (2020) News recommendation systems-accomplishments, challenges future directions,it IEEE Access , vol. 8, pp. 16702-16725. https://doi.org/10.1109/ACCESS.2020.2967792

[35] M. C. Kim and C. Chen (2015) A scientometric review of emerging trends and new developments in recommendation systems,*Scientometrics*, vol. 1, pp. 239-263. https://doi.org/10.1007/s11192-015-1595-5

[36] G. Wei, Q. Wu and M. Zhou (2021) A hybrid probabilistic multiobjective evolutionary algorithm for commercial recommendation systems, *IEEE Transactions on Computational Social Systems*, vol. 8, no. 3, pp. 589-598. https://doi.org/10.1109/TCSS.2021.3055823

[37] V. T. R. M. Vivek, C. Saravanan and K. V. Kumar(2022) A Novel Technique for User Decision Prediction and Assistance Using Machine Learning and NLP: A Model to Transform the E-commerce System, *Big data management in Sensing: Applications in AI and IoT*, p. 61.

[38] J. Davidson, B. Liebald, J. Liu, P. Nandy, T. V. Vleet, U. Gargi and S. Gupta (2010) The YouTube video recommendation system,*Proceedings of the fourth ACM conference on Recommender systems*, pp. 293-296. https://doi.org/10.1145/1864708.1864770

[39] W. Bellante, R. Vilardi and D. Rossi(2013) On Netflix catalog dynamics and caching performance, /it 2013 IEEE 18th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), pp. 89-93. https://doi.org/10.1109/CAMAD.2013.6708095

[40] X. H. Pham, T. N. Luong and J. J. Jung (2013) An black-box testing approach on user modeling in practical movie recommendation systems, *International Conference on Computational Collective Intelligence*, pp. 72-79. https://doi.org/10.1007/978-3-642-40495-5_8

[41] S. Raza and C. Ding (2021) News recommender system: a review of recent progress, challenges, and opportunities,*Artificial Intelligence Review*, pp. 1-52. https://doi.org/10.1007/s10462-021-10043-x

[42] F. Carmagnola, F. Vernero and P. Grillo(2009) Sonars: A social networks-based algorithm for social recommender systems,*International Conference on User Modeling, Adaptation, and Personalization*, pp. 223-234. https://doi.org/10.1007/978-3-642-02247-0_22

[43] W. Zou (2018) Design and application of incremental music recommendation system based on Slope one algorithm,*Wireless Personal Communications*, vol. 102, no. 4, pp. 2785-2795. https://doi.org/10.1007/s11277-018-5303-7

[44] W. Strank (2021) Analyzing Networks of Musical Context in the Digital Age1, *119: The Oxford Handbook of Music and Advertising*. pp.6080-6088 https://doi.org/10.1093/oxfordhb/9780190691240.013.48

[45] J. Sanz-Cruzado Puig(2021) Contact recommendation in social networks: algorithmic models,*diversity and network evolution*,pp. 519-569

[46] L. Lü, M. Medo, C. H. Yeung, Y.-C. Zhang and a. T. Z. Zi-Ke Zhang (2012) Recommender systems, *Physics reports*, vol. 519, no. 1, pp. 1-49. https://doi.org/10.1016/j.physrep.2012.02.006

[47] E. Brynjolfsson, Y. Hu and M. D. Smith (2003) Consumer surplus in the digital economy: Estimating the value of increased product variety at online booksellers, *Management science*, vol. 49, no. 11, pp. 1580-1596. https://doi.org/10.1287/mnsc.49.11.1580.20580

[48] X. Amatriain and J. Basilico (2015) Recommender systems in industry: A netflix case study, *Recommender systems handbook*, pp. 385-419. https://doi.org/10.1007/978-1-4899-7637-6

[49] H. Verma (2022) Netflix Recommendation Engine - How Netflix uses Big data and Analytics to Recommend you your Favourite Shows, *startuptalky* [Online]. Available: https://startuptalky.com/netflix-recommendation-engine/ [Accessed 29 10 2022].

[50] M. Zhang and Y. Liu (2021) A commentary of TikTok recommendation algorithms in MIT Technology Review 2021, *Fundamental Research*, vol. 1, no. 6, pp. 846-847. https://doi.org/10.1016/j.fmre.2021.11.015

[51] Z. Chen and C. Shi (2022) Analysis of Algorithm Recommendation Mechanism of TikTok, *International Journal of Education and Humanities*, vol. 4, no. 1, pp. 12-14.

[52] S. Albawi, T. A. Mohammed and S. Al-Zawi (2017) Understanding of a convolutional neural network, *international conference on engineering and technology*, pp. 1-6. https://doi.org/10.1109/ICEngTechnol.2017.8308186

[53] J. Chai, H. Zeng, A. Li and E. W. Ngai (2021) Deep learning in computer vision: A critical review of emerging techniques and application scenarios, *Machine Learning with Applications*, vol. 6, p. 100134. https://doi.org/10.1016/j.mlwa.2021.100134

[54] N. L. H. Hien, L. V. Huy and N. V. Hieu (2021) Artwork style transfer model using deep learning approach, *Cybernetics and Physics*, vol. 10, no. 3, pp. 127-137. http://doi.org/10.35470/2226-4116-2021-10-3-127-137

[55] L. Liu, F.-X. Wu, Y.-P. Wang and J. Wang (2020) Multi-receptive-field CNN for semantic segmentation of medical images, *IEEE Journal of Biomedical and Health Informatics*, vol. 24, no. 11, pp. 3215-322. https://doi.org/10.1109/JBHI.2020.3016306

[56] N. L. H. Hien and A.-L. Kor (year) Analysis and Prediction Model of Fuel Consumption and Carbon Dioxide Emissions of Light-Duty Vehicles, *Applied Sciences*, vol. 12, no. 2, p. 803. https://doi.org/10.3390/app12020803

[57] Prabhu (2018) Understanding of Convolutional Neural Network (CNN) — Deep Learning, *Medium* [Online]. Available: https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-d [Accessed 29 10 2022].

[58] N. L. H. Hien, H. M. Hoang, N. V. Hieu and N. V. Tien (2021) Keyphrase Extraction Model: A New Design and Application on Tourism Information, *Informatica*, vol. 45, no. 4. https://doi.org/10.31449/inf.v45i4.3493

[59] P. J. L. and G. W. Li (2018) Application of convolutional neural network in natural language processing, *15th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*, pp. 120-122. https://doi.org/10.1109/ICCWAMTIP.2018.8632576

[60] Y. R. B. and C. V. Koren (2009) Matrix factorization techniques for recommender systems, *Computer*, vol. 42, no. 8, pp. 30-37. https://doi.org/10.1109/MC.2009.263

[61] H.-J. Xue, X. Dai, J. Zhang, S. Huang and J. Chen (2017) Deep matrix factorization models for recommender systems, *IJCAI*, vol. 17, pp. 3203-3209. https://www.ijcai.org/proceedings/2017/447

[62] grouplens (2003) MovieLens 1M Dataset [Online]. Available: https://grouplens.org/datasets/movielens/1m/ [Accessed 29 10 2022].

[63] H. Liu, C. Zheng, D. Li, X. Shen, K. Lin, J. Wang, Z. Zhang, Z. Zhang and N. N. Xiong (2021) EDMF: Efficient deep matrix factorization with review feature learning for industrial recommender system, *IEEE*

Transactions on Industrial Informatics, vol. 18, no. 7, pp. 4361-4371. https://doi.org/10.1109/TII.2021.3128240

[64] H. Wu, Z. Zhang, K. Yue, B. Zhang, J. He and L. Sun(2018) Dual-regularized matrix factorization with deep neural networks for recommender systems, *Knowledge-Based Systems*, vol. 145, pp. 46-58. https://doi.org/10.1016/j.knosys.2018.01.003