WJCMS

Wasit Journal of Computer and Mathematics Science

Journal Homepage: https://wjcm.uowasit.edu.iq/index.php/WJCM

e-ISSN: 2788-5879 p-ISSN: 2788-5879

Dietary Behavior Based Food Recommender System Using Deep Learning and Clustering Techniques

Ammar Abdulsalam Al-Asadi[1,*] and Mahdi Nsaif Jasim[2]

[1]Informatics Institute for Postgraduate Studies, Iraqi Commission for Computers and Informatics, Iraq
[2]College of Business Informatics, University of information technology and communications, Iraq

*Corresponding Author: Ammar Abdulsalam Al-Asadi

DOI: https://doi.org/10.31185/wjcm.126
Received: April 2023; Accepted: June 2023; Available online: June 2023

ABSTRACT: Deep learning algorithms have been highly successful in various domains, including the development of collaborative filtering recommender systems. However, one of the challenges associated with deep learning-based collaborative filtering methods is that they require the involvement of all users to construct the latent representation of the input data, which is then utilized to predict the missing ratings of each user. This can be problematic as some users may have different preferences or interests, which may affect the accuracy of the prediction generation process. The research proposed a food recommender system, which tries to find users with similar dietary behavior and involve them in the recommendations generation process by combining clustering technique with denoising autoencoder to generate a rate prediction model. It is applied to "Food.com Recipes and Interactions" dataset. RMSE score was used to evaluate the performance of the proposed model which is 0.1927. It outperformed the other models that used autoencoder and denoising autoencoder without clustering where the RMSE values are 0. 4358 and 0.4354 consequently.

Keywords: Deep neural network, Clustering, Food recommender system, Autoencoders, K-means.

1. INTRODUCTION

The A huge amount of information is generated every day, due to the rapid development of applications and internet services. This phenomenon is called information overload, which makes the process of finding the information that are relevant to the users' interest more difficult [1]. This issue has encouraged re-searchers to develop recommender systems (RSs) that are able to filter out the services and items and pro-vide the appropriate ones for users [2]. RS uses the users' historical interactions with items to extract their preferences and generate recommendations. Additional information could be used by RS such as users' profiles or items' features [3]. The primary categorization of RSs consists of three types, namely content-based (CB), collaborative filtering (CF), and hybrid filtering [4].

CF is the most efficient and simple method among them. It has been used in many real-world systems such as Amazon. CF is divided into user-based and item-based filtering depending on the adopted prediction technique [5]. Hybrid filtering attempts to merge the methods of CF and CB to surpass the constraints of recommendation systems [6].

Clustering is one of the unsupervised learning methods, it is widely used in RS in order to group the users or items based on their similarity [7]. Each group is called a cluster and each cluster contains very similar members by a given data properties. The members of different clusters are dissimilar [8].

Deep learning achieves great success in several fields of applications such as speech recognition, computer vision, and natural language processing [9]. Academia and industry are becoming interested in applying deep learning to different applications because it is able to solve many problems and achieve high-quality results [10]. Deep learning-based rec-

*Corresponding author: phd202010553@iips.icci.edu.iq

ommender systems have been intently studied in recent years [11], where the combination of user and item features is utilized to create predictions via the use of multiple layers of perceptron. Deep structured semantic models [12] analyses the users' written content and behaviors to map the users and items into a latent representation to maximize the similarity between users and their preferred items. Lately, many research works use CF and deep learning to generate collaborative deep learning-based recommender systems [13–16]. Wang [13] proposed a hierarchical Bayesian model which is named collaborative deep learning to perform deep representation learning of the CB and CF for the user-item interactions matrix. Autoencoder approach [17–20] has introduced recently into the RS field to compute non-linear matrix factorization by using autoencoders frameworks and user-item inter-actions matrix [21].

The problem with the reviewed deep learning-based CF methods is that, they require the involvement of all users to construct the latent representation of the input data, which is then utilized to predict the missing ratings of each user. This can be problematic as some users may have different preferences or interests and they will contribute in predictions generating process.

This research proposes a food recommender system, which tries to find users with similar behavior and involve them in the recommendations generation process by combining clustering technique with denoising autoencoder to generate a rate prediction CF model. It is applied to "Food.com Recipes and Interactions" dataset [22] after several pre-processing steps including removing the outliers and the users with low interactions.

The remainder of the paper is organized as follows: The theoretical underpinnings of the approaches employed in the proposed method were described in Section 2; the proposed method was detailed in Section 3; findings and performance evaluation were offered in Section 4; and the paper conclusions were covered in Section 5.

2. PROPOSED METHODOLOGY

This paper proposes a new method for food recommendation. The recommendations are generated based on the dietary behavior of each user by utilizing clustering techniques such as k-means to categorize users based on their dietary behavior similarity, then each cluster trains a rate prediction model to generate recommendation to the cluster's members using deep learning network such as denoising autoencoder. The proposed model is called cluster-based denoising autoencoder (cluster-based DAE).

The next subsections (2.1, 2.2, 2.3, and 2.4) present the theoretical background of algorithms that are used to pre-process the data and build the proposed model.

2.1 BOX PLOT

Tukey in 1977 introduced the rules for the construction of a boxplot to screen outliers [23]. To construct a boxplot, five statistics are necessary: the smallest value, the first quartile (Q1), the second quartile (Q2), the third quartile (Q3), and the largest value. The data is divided into four equal sections, each containing around 25% of the values. Q1 separates the data into two sections: one with values below Q1 (25% of the data) and the other with the remaining 75% of the data. Similarly, Q2 splits the data into two sections with equal amounts of data above and below Q2, and the same applies to Q3. The boxplot consists mainly of a rectangle that extends from Q1 to Q3, and its length represents the inter-quartile range (IQR), which is the range between Q1 and Q3. The corresponding lengths are computed as:

$$Lowerboundary = Q1 - 1.5 \times IQR \tag{1}$$

$$Upperboundary = Q3 + 1.5 \times IQR \tag{2}$$

Characteristically, outliers are defined as data points beyond the boundaries [24].

2.2 CLUSTERING

K-Means is one of the clustering algorithms in partitioning methods. It is the most popular, straightforward, and computationally efficient clustering technique [25]. K-Means categorizes data points into K clusters based on the clusters' centers or centroids. The centroid of each cluster is obtained by calculating the average of all the data points in that cluster. Before training the clustering model the users need to specify the number of clusters (K). The main steps of K-means algorithm are [26]:

1. Select K points as centroids randomly.
2. Calculate the distance between each data point and each cluster's centroid using Euclidean distance.
3. Assign each data point to the closest cluster.

4. Compute the mean of the assigned data points to update the centroids.

5. Repeat 2, 3, and 4 until the number of maximum iterations is reached.

Silhouette is one of the methods that are used to find the optimal number of clusters (K). The goal of the silhouette method is to assess the coherence of clusters formed by data points. To determine the silhouette coefficients, the method calculates the similarity between a data point and its assigned cluster relative to the other clusters. This coefficient is determined using [27]:

$$\frac{(b-a)}{max(a,\ b)} \tag{3}$$

Here, 'a' denotes the average intra-cluster distance, while 'b' represents the average distance to the nearest cluster for each data point.

2.3 AUTOENCODER

Autoencoders (AE) were first presented in 1991 by Kramer [28]. It is a deep learning algorithm which obtains high-level representation of original features [29]. Autoencoders employ feed-forward neural networks to acquire a compact dimension input representation. The AE network attempt to reproduce the input through its output. The training process of the network involves back-propagation of the loss (e.g., mean squared error) through the reconstruction process, utilizing two components:

$$Encoder\varphi : x \rightarrow z$$
$$Decoder\Psi : z \rightarrow x$$

Here, φ and Ψ are defined as the values of the encoder and decoder, respectively, which minimize the expression $\|x - (\varphi * \Psi)\ x\|^2$. In the simplest case, the autoencoder contains only one hidden layer, where the encoder maps the input x to z, and then the decoder reconstructs x from z using the following equations:

$$Encoder : z = \sigma(W_X + b)$$
$$Decoder : x = \sigma(W'_Z + b')$$

Here, σ denotes a non-linear activation function, $x \in R^d$ is the input, $z \in R^d$ is the hidden node, $W \in R^{(dk)}$ is a weight matrix that maps the input to the hidden node, $W' \in R^{(dk)}$ is the weight matrix that maps the hidden node to the reconstructed node, and $b \in R^k$, $b' \in R^d$ are biases [21].

2.4 DENOISING AUTOENCODER

Denoising Autoencoder (DAE) was presented by Vincent [30] to discover more robust features over autoencoding and learn the identity function. It utilizes a distorted input x represented as $\tilde{x}$ to train the network to remove noise and restore the original input x. Various distortion options can be used including the additive Gaussian noise and multiplicative mask-out/ drop-out noise. In this paper, the drop-out noise is utilized to randomly mask entries of the input (based on the noise ratio) by setting them to zero [31].

3. EXPERIMENTAL DESIGN

This section will discuss the design of the proposed system and the evaluation results.

3.1 DATASET DESCRIPTION

The proposed system has been applied to "Food.com Recipes and Interactions" Dataset, which has 230K+ recipes and 1M+ interactions (rates) given by 226K+ users over 18 years (2000-2018) from Food.com [22]. Two tables have been selected from this dataset for use:

1. RAW_interactions: it contains ratings provided by users for recipes, with values ranging from 1 to 5..

2. RAW_recipes: it contains all recipes with their nutrition information and recipes description.

The dataset is pre-processed in order to prepare it for further steps as follows:

1. Extracting the nutrition values from “nutrition” column of (RAW_recipes) table, and put them in separated columns: calories, total fat, sugar, sodium, protein, saturated fat, and carbohydrate.
2. Identifying and removing the outlier values of all seven nutrition columns by using box plot method. IQR is multiplied by 2 instead of 1.5 in Eq. (1, 2) to reduce the number of outliers. The output of this step is 130K+ recipes. Figure 1 shows the distribution of ‘calories’ column’s values before and after removing its outliers..
3. emoving the users and items with less than 20 interactions which is determined empirically, to reduce the sparsity.

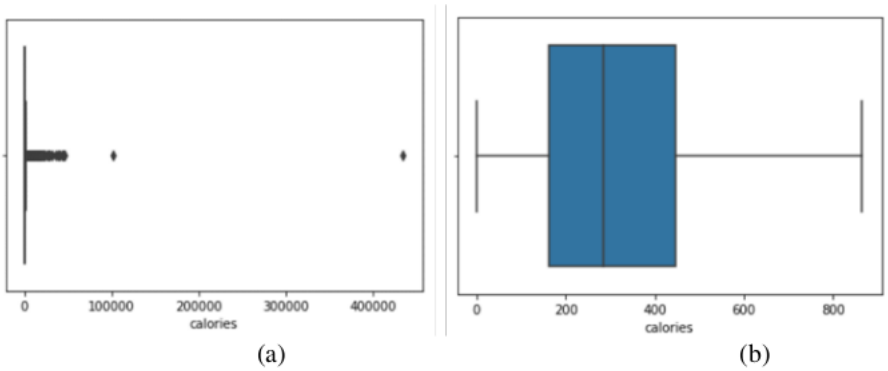

FIGURE 1. The distribution of calories values: (a) before removing the outliers, (b) after removing the outliers

3.2 EXPERIMENTAL DESIGN AND HYPER PARAMETERS

The proposed system involves two core phases:

a) Users clustering

The proposed system groups the users into clusters based on the similarity of their dietary behavior by applying k-means algorithm. The similarities among users are found by extracting the average nutrition consumption of each user. Silhouette method is used to determine the best number of clusters using (3). Figure 2 shows part of the users-nutrition matrix. This matrix is achieved by merging interactions table with recipes table.

user_id	calories	total_fat	sugar	sodium	protein	saturated_fat	carbohydrates
1533	0.356112	0.237778	0.186408	0.232000	0.252874	0.302924	0.301754
1535	0.347072	0.212169	0.344686	0.200106	0.161430	0.286086	0.378725
1634	0.328577	0.266667	0.065187	0.222857	0.256158	0.406015	0.172932
1676	0.351297	0.233333	0.160194	0.245000	0.362069	0.135965	0.263158
1792	0.565360	0.461111	0.177994	0.473333	0.436782	0.409357	0.368421
...	...	...	...	...	...	...	...
2001359614	0.189761	0.137500	0.269417	0.120000	0.055172	0.280702	0.197368
2001362355	0.498775	0.304167	0.286408	0.368571	0.340394	0.441729	0.490602
2001410644	0.338395	0.280952	0.142857	0.234286	0.242365	0.298246	0.225564
2001436530	0.423611	0.250000	0.197411	0.260000	0.390805	0.228070	0.333333
2001524408	0.321920	0.183333	0.377023	0.193333	0.167816	0.286550	0.377193

7470 rows × 7 columns

FIGURE 2. Part of users-nutrition matrix

b) Training DAE model

During the training stage, the dataset is split into K subsets based on the values in the "cluster" column. Each subset is then divided into two parts for training and testing with 80% and 20% of the data, respectively. All clusters use the same model structure for training.

The proposed model employs an item-based structure that uses an items-users matrix as input (r_i), where each row represents an item and each column represents a user, and the intersection between a row and a column indicates the rating given by a user to an item. This structure is chosen because the number of users involved in a sample is smaller than the number of items due to the clustering process, which reduces the number of nodes in the input and output layers. The model structure is illustrated in Figure 3 and is designed as follows:

1) Input layer

Input layer has nodes equal to the number of users in a specific cluster. Each input node represents the interaction between an item (denoted as i) and all the users within the cluster. This interaction is measured by a set of rate values (R_{1i}, R_{2i}, R_{3i}, R_{mi}) where R is the rating value and m is the number of users. Dropout noise technique is applied, which randomly sets some of the input values to zero. The noise ratio is set at 50% to generate a corrupted version of the input data, denoted as $\widetilde{r_i}$.

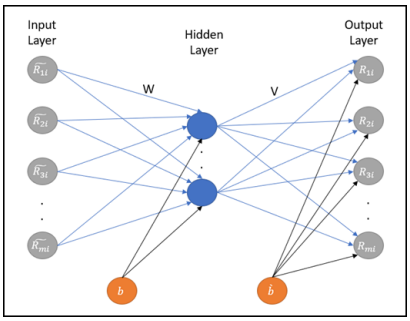

FIGURE 3. The structure of the proposed model

2) Hidden layer

The hidden layer is fully connected to the corrupted input data $\widetilde{r_i}$. This layer uses the sigmoid activation function (f). The output of the hidden layer is calculated using the formula [32]:

$$z = f(W * \widetilde{r_i} + b) \tag{4}$$

Where W and b are the weights and bias of the hidden layer, respectively.

3) Output layer

The output layer is also fully connected to the hidden layer and has the same number of nodes as the input layer. The weights of the output layer are represented by V. The predicted rates $\widehat{r_i}$ are the output of this layer, and they are calculated using the formula [32]:

$$\widehat{r_i} = f(V * z + \hat{b}) \tag{5}$$

Where bís the bias of the output layer and f is the linear activation function.

4) Cost function

Mean Squared Error (MSE) is used in the proposed model as a cost function. it is not appropriate to predict zeros in the item's representation vector R_{mi}. To overcome this issue, the approach from [15] uses Masked Mean Squared Error (MMSE) loss. MMSE is calculated using the actual rating r_i, the reconstructed or predicted rating $\widehat{ri}$, and a mask function m_i. The mask function is set to 1 if $r_i \neq 0$ and 0 otherwise. The MMSE formula is as follows [33]:

$$MMSE = \frac{m_i * (r_i - \widehat{r_i})^2}{\sum_{i=0}^{i=n} m_i} \tag{6}$$

5) Weights updating

The Adam optimization algorithm [31] has been employed to update the weights of the network using a learning rate of 0.0001. This algorithm merges the 'Root Mean Square Propagation algorithm' and 'gradient descent with momentum algorithm'.

L2 norm regularization has been applied on the hidden layer weights with a regularization rate (0.1) in order to avoid the overfitting issue. The key phases of the proposed method are showed in Algorithm 1.

3.3 MODEL EVALUATION

The evaluation of the predicted ratings is performed using Root Mean Squared Error (RMSE), which compares them with the actual testing data to assess their accuracy. RMSE has a direct relationship with the MMSE score and is used as a metric for evaluation [33].

$$RMSE = \sqrt{MMSE} \tag{7}$$

4. RESULTS AND DISCUSSION

The outcomes obtained from the experiments conducted on the suggested model (cluster-based DAE) are computed by taking the average RMSE of all the seven clusters' models, where the number of clusters are selected based on the Silhouette score. The models' evaluation results are shown in Table 1. Since, we could not find other research works using the same dataset with same evaluation matrices, the proposed model is compared with other models such as AE and DAE without clustering to evaluate its performance. The comparison of RMSE scores and the models' parameters are illustrated in Table 2. All the models use 80% of the dataset for training and 20% for testing. Adam optimizer with (0.0001) learning rate is used in all models. They all use the same activation functions in the hidden layer and the output layer which are "sigmoid" and "linear" activation functions respectively. DAE and cluster-based DAE models use drop-out noise with a ratio of (0.5) with the input data. A smaller regularization rate (0.1) is used in the hidden layer's weights of Cluster-based DAE since it has a smaller number of input nodes.

Algorithm 1. The procedure of the proposed model

Input: Interactions data;
Output: Predictions, RMSE;

$k \leftarrow$ number of clusters +1
Set counter i
While $i < k$ do
  Find all ratings of cluster i
  Generate item-user rating matrix r_i
  Split rating matrix into train and test sets
  Build AE model with:
    input/output nodes = number of users in cluster i
  For each item in the train set do
    $\tilde{r}_i \leftarrow$ dropout_noise (r_i, 0.5)
    Compute z using (4)
    Compute $\hat{r}_i$ using (5)
    Compute MMSE using (6)
    Update parameters using Adam optimizer
  End for
  Generate predictions
  Compute RMSE using the test set and (7)
  i ← i+1
  End while
Compute average RMSE

Table 1. RMSE of each cluster's model

Cluster	#Items	# Users	RMSE
0	3970	685	0. 2197
1	3969	653	0.2282
2	2123	316	0.1534
3	2849	395	0.1541
4	3043	381	0.1354
5	2939	416	0.1761
6	4405	1009	0.2821
			Average: 0.1927

Table 2. Comparing the proposed method with other non-clustering methods

Model	#Hidden Nodes	Regularization Rate	RMSE
AE	256	0.5	0.4358
DAE	256	0.5	0.4354
Proposed model	128	0.1	**0.1927**

The model proposed in the study was coded using Python 3.9.12 and relied on several packages such as, Pandas 1.4.2, NumPy 1.22.4, , Scikit-learn 1.1.1, Tensorflow 2.9.1, Matplotlib 3.5.2, and Keras 2.9.0.

5. CONCLUSION

This paper proposes a food recommender system based on denoising autoencoder and k-means algorithm to cluster users into groups based on their personal dietary behavior. Each cluster's members cooperate to train a denoising autoencoder model to learn the latent space of corrupted version of item-user interaction matrix and avoid involving users with different dietary behaviors in this process. The proposed model showed a great success in reducing the RMSE score of the predicted rates within each cluster, where the average was (0.1927). It outperformed the other models that used autoencoder and denoising autoencoder without clustering where their RMSE scores were (0.4358) and (0.4354) respectively.

FUNDING

None

ACKNOWLEDGEMENT

None

CONFLICTS OF INTEREST

The author declares no conflict of interest.

REFERENCES

[1] A. A. Neamah and A. S. El-Ameer, "Design and Evaluation of a Course Recommender System Using Content-Based Approach," *2018 International Conference on Advanced Science and Engineering (ICOASE)*, 2018.
[2] Q. Zhang, J. Lu, and Y. Jin, "Artificial intelligence in recommender systems," *Complex & Intelligent Systems*, vol. 7, no. 1, pp. 439–457, 2021.
[3] S. Zhang, L. Yao, A. Sun, and Y. Tay, "Deep learning based recommender system: A survey and new perspectives," *ACM Computing Surveys (CSUR)*, vol. 52, no. 1, pp. 1–38, 2019.
[4] G. Zhang, Y. Liu, and X. Jin, "A survey of autoencoder-based recommender systems," *Front Comput Sci*, vol. 14, no. 2, pp. 430–450, 2020.
[5] Y. Koren, S. Rendle, and R. Bell, "Advances in collaborative filtering," Recommender systems handbook," 2022. 91-142.
[6] M. H. Mohamed, M. H. Khafagy, and M. H. Ibrahim, "Recommender systems challenges and solutions survey," *2019 International Conference on Innovative Trends in Computer Engineering (ITCE)*, pp. 149–155, 2019.
[7] A. A. Al-Asadi and M. N. Jasim, "Deep Learning-Based Rate Prediction Model for Recommender System Using Clustering Techniques."
[8] R. Rashidi, K. Khamforoosh, and A. Sheikhahmadi, "Proposing improved meta-heuristic algorithms for clustering and separating users in the recommender systems," *" Electronic Commerce Research*, vol. 22, no. 2, pp. 623–648, 2022.
[9] J. Chai, H. Zeng, A. Li, and E. W. T. Ngai, "Deep learning in computer vision: A critical review of emerging techniques and application scenarios," *Machine Learning with Applications*, vol. 6, pp. 100134–100134, 2021.
[10] Y. Peng, "Cross-media analysis and reasoning: advances and directions," *Frontiers of Information Technology & Electronic Engineering*, vol. 18, no. 1, pp. 44–57, 2017.
[11] I. A. S. Jabbar, R. S. Alhamdani, and M. N. Abdullah, "Analyzing Restricted Boltzmann Machine Neural Network for Building Recommender Systems," *2019 2nd International Conference on Engineering Technology and its Applications (IICETA)*, pp. 133–137, 2019.
[12] P. S. Huang, X. He, J. Gao, L. Deng, A. Acero, and L. Heck, "Learning deep structured semantic models for web search using clickthrough data," *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pp. 2333–2338, 2013.
[13] H. Wang, N. Wang, and D. Y. Yeung, "Collaborative deep learning for recommender systems," *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1235–1244, 2015.
[14] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T. S. Chua, "Neural collaborative filtering," *Proceedings of the 26th international conference on world wide web*, pp. 173–182, 2017.
[15] J. Wei, J. He, K. Chen, Y. Zhou, and Z. Tang, "Collaborative filtering and deep learning based recommendation system for cold start items," *Expert Syst Appl*, vol. 69, pp. 29–39, 2017.
[16] S. Sedhain, A. K. Menon, S. Sanner, and L. Xie, "Autorec: Autoencoders meet collaborative filtering," *Proceedings of the 24th international conference on World Wide Web*, pp. 111–112, 2015.
[17] S. Li, J. Kawale, and Y. Fu, "Deep collaborative filtering via marginalized denoising auto-encoder," *Proceedings of the 24th ACM international on conference on information and knowledge management*, pp. 811–820, 2015.
[18] Y. Pan, F. He, and H. Yu, "Learning social representations with deep autoencoder for recommender system," *World Wide Web*, vol. 23, no. 4, pp. 2259–2279, 2020.
[19] Y. Ouyang, W. Liu, W. Rong, and Z. Xiong, "Autoencoder-based collaborative filtering," *International conference on neural information processing*, pp. 284–291, 2014.
[20] X. Dong, L. Yu, Z. Wu, Y. Sun, L. Yuan, and F. Zhang, "A hybrid collaborative filtering model with deep structure for recommender systems," *Proceedings of the AAAI Conference on artificial intelligence*, 2017.
[21] Y. Pan, F. He, and H. Yu, "A novel enhanced collaborative autoencoder with knowledge distillation for top-N recommender systems," *" Neurocomputing*, vol. 332, pp. 137–148, 2019.
[22] B. P. Majumder, S. Li, J. Ni, and J. Mcauley, "Generating Personalized Recipes from Historical User Preferences," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 5975–5981, Association for Computational Linguistics, 2019.
[23] S. B. Belhaouari, "Unsupervised outlier detection in multidimensional data," *J Big Data*, vol. 8, no. 1, pp. 1–27, 2021.

[24] D. C. Li, W. K. Huang, and Y. S. Lin, "New Product Short-Term Demands Forecasting with Boxplot-Based Fractional Grey Prediction Model," *Applied Sciences*, vol. 12, no. 10, pp. 5131–5131, 2022.
[25] X. Ran, X. Zhou, M. Lei, W. Tepsan, and W. Deng, "A novel k-means clustering algorithm with a noise algorithm for capturing urban hotspots," *Applied Sciences*, vol. 11, no. 23, pp. 11202–11202, 2021.
[26] T. Puraram, P. Chaovalit, A. Peethong, P. Tiyanunti, S. Charoensiriwath, and W. Kimpan, "Thai food recommendation system using hybrid of particle swarm optimization and K-means algorithm," *ACM International Conference Proceeding Series*, pp. 90–95, 2021.
[27] P. J. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *J Comput Appl Math*, vol. 20, pp. 90125–90132, 1987.
[28] M. A. Kramer, "Nonlinear principal component analysis using autoassociative neural networks," *" AIChE journal*, vol. 37, no. 2, pp. 233–243, 1991.
[29] M. Yu, T. Quan, Q. Peng, X. Yu, and L. Liu, "A model-based collaborate filtering algorithm based on stacked AutoEncoder," *Neural Comput Appl*, vol. 34, no. 4, pp. 2503–2511, 2022.
[30] P. Vincent, H. Larochelle, Y. Bengio, and P. A. Manzagol, "Extracting and composing robust features with denoising autoencoders," *Proceedings of the 25th international conference on Machine learning*, pp. 1096–1103, 2008.
[31] Y. Xiong and R. Zuo, "Robust feature extraction for geochemical anomaly recognition using a stacked convolutional denoising autoencoder," *Math Geosci*, vol. 54, no. 3, pp. 623–644, 2022.
[32] J. Zhao, L. Wang, D. Xiang, and B. Johanson, "Collaborative Deep Denoising Autoencoder Framework for Recommendations," 2019.
[33] O. Kuchaiev and B. Ginsburg, "Training deep autoencoders for collaborative filtering," 2017.