

Tensorized Neural Network for Multi-Aspect Rating-Based Recommendation

Nurkhairizan Khairudin, Nurfadhlina Mohd Sharef, Shahrul Azman Mohd Noah, Norwati Mustapha

Abstract— Generating personalized recommendations is one of the most crucial aspects in Recommender System research area. Most of the researches only focus on the accuracy of recommendation using collaborative filtering that relies on a single overall rating that represents the overall preferences. However, the user may have a different emphasis on different specific aspects that affect the users' final rating decisions. Therefore, we present a neural network model that utilize multi-aspects ratings using Tensor Factorization to improve the accuracy of personalization, as well as optimizing the dynamic weights of the aspect. To measure the estimated weights for the aspects, we employ the Higher Order Singular Value Decomposition (HOSVD) technique called CANDECOMP/PARAFAC (CP) decomposition that allows for multi-faceted data processing. We then develop the Neural Network with backpropagation error to train the model with different parameter settings and limited computational time. We also use a non-linear activation function in each hidden layer in various settings. The experimental result measured using MAE shows that the proposed model has significantly outperformed the baseline approach in terms of the prediction accuracy. Based on the observation, the performance of rating prediction has been improved by employing the Tensorized Neural Network model and can overcome the problem of local optimum convergence for multi-aspect rating recommendation.

Index Terms: Collaborative filtering, multi-aspect, recommendation system, tensor factorization.

I. INTRODUCTION

Rapid and variety development of IT applications nowadays leads to the increasingly outrageous amounts of collected and stored data that brings to the era of big data. One of the popular application systems that contribute to the importance of this area is Recommender systems (RSs). Recently, RSs are no longer depending on the rating matrix only for the purpose of producing an accurate personalized recommendation, but they try to make use of various kind of available data on the web. This include the users' demography [1], [2] users and items features [3], [4], tags and time [5], social relation [6], [7] contextual features [8]-[10], and geographic information [11], [12] which are somehow incorporated into the recommendation models for the purpose of improving the performance of the recommendation.

Revised Version Manuscript Received on September 16, 2019.

Nurkhairizan Khairudin, Faculty of Computer and Mathematical Science, Universiti Teknologi MARA, Selangor, Malaysia.

Nurfadhlina Mohd Sharef, Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, Selangor, Malaysia.

Shahrul Azman Mohd Noah, Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, Selangor, Malaysia.

Norwati Mustapha, Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, Selangor, Malaysia.

Multi-criteria ratings RS aims at leveraging historical ratings with users' and items' additional information such as reviews and contextual features to predict user preferences for a personalized recommendation. For review-based recommendations, most existing approaches make a prediction on unknown ratings by implementing the sentiment/polarity on reviews as a whole such as [13]-[15]. Furthermore, most existing approaches in multi-criteria RS are focusing on utilizing contextual information which available explicitly (e.g.: age, gender, time and etc.) or implicitly (e.g., extracted context from reviews and item descriptions). It is difficult to capture the complex correlations among these contextual features because of the limitation of contextual expression. Whereas, aspects extracted from reviews can represent more condition preferences.

Popular recommendation approach such as Matrix Factorization can model the relationship of 2 dimensions only. To solve the multi-aspect rating problem, Tensor Factorization is employed to uncover relationships among users, items and aspects. TF has the ability to directly incorporate more dimensions in addition to rating information which is not possible through the application of MF [16]. However, TF is one of the multivariate linear models where all dimensions are factorized in a linear way [17], whereas the real interactions between dimensions are in a non-linear way. Furthermore, as more dimensions are added, the number of model parameters grows exponentially, resulted in computationally intensive and very time-consuming.

Recently, neural network approaches with review content have been applied to recommender system to overcome the non-linear problem. However, with the high dimensionality of data, most problems are non-convex where the preference learning is stuck to the local optimal solution. Therefore, there is no guarantee for the model to get to a global optimal solution. Therefore, this research proposes a weighted Tensorized Neural Network model for personalized RS which incorporates aspects ratings retrieved from reviews expressed by the user for an item.

The remainder of the paper is organized as follows. Section II reviews some of the related works. Section III explains the proposed recommendation algorithm. Next, experimental settings have been detailed out in Section IV. Results are presented in Section V and finally, section VI concludes this paper.

II. RELATED WORKS

Recently, Collaborative Filtering (CF) that relies on overall ratings has been widely accepted approach due to the ability to generate recommendations based on the ratings of like-minded users. However, sometimes the overall rating alone is not enough to precisely understand users' interest. A user may show his/her overall preferences on an item through the overall ratings but at the same time, they may have satisfaction level variation in certain aspects. Existing works on the aspect-based review can be divided into the expressed aspects emphasis through the textual based review [18] and the numerical representation of the aspects [19], [20]. However, the textual based aspects emphasis cannot compute the representation of aspects weights interdependencies towards the overall rating.

When incorporating other information such as aspects ratings in the predictions, the dimensionality of input will increase, and this situation reflects on the problem of multi-dimensionality that lead to the computational complexity. There are researches such as [21]-[23] which applied dimensionality reduction to overcome this problem. In [21] they apply Latent Semantic Indexing (LSI) to identify the important dimension while [22] implement Higher Order Value Decomposition (HOSVD) to reduce the dimensionality. In [23] proposed a neural network model for the multi-aspect-based recommendation which only selects strongly correlated aspects to be used as the auxiliary aspect rating to the overall ratings. However, they only use the correlation values to determine the strength of the aspects to the overall ratings.

Instead of dimensional reduction-based approach, we propose a weighted-based multi-aspect rating parameters that are applicable to be used to enhance the predictive accuracy of recommendation

III. RECOMMENDATION ALGORITHM

In this section, we discuss the implementation of the proposed framework for weighted Tensor Factorized Neural Network for Multi-Aspect rating prediction, wTF-NNMA. We start with the notation symbols used in the explanation. U, V and K representing the set of users, items and aspect respectively. These components are used to build m-order tensor, X. We use n to indicate the aspect ratings K_n , where n holds the number of aspect ratings. The weighted aspect ratings are represented by S which is used to calculate the weighted TF to indicate the relation between users, items and aspects.

The recommendation algorithm consists of three parts started by building the m-order tensor, X from raw multi-aspect rating dataset as described in Section IV-A. Algorithm 1 shows the steps for building m-order tensor, X. The output from this phase is used in the decomposition phases to calculate the weighted factor matrices and core tensor, λ of m-order tensor, X for K_{1-n} as shown in Algorithm 2. The weighted factor matrices with the core tensor, λ , is used to generate a single weight for every user, $wTF - u$ items, $wTF - v$ and aspects $wTF - k$. These matrices are fed to the neural network model with 5 different layer settings to measure the accuracy of the overall rating prediction. The steps for this phase can be referred to as Algorithm 3.

To build m-order tensor, X we first map the multi-aspect rating data to a separated file for each aspect. All unknown rating for all aspect is set to a default value of 0. This is important to make sure the calculation of weighted aspect values returns a meaningful value.

The main process of tensor decomposition is done by Algorithm 2 where all aspects converted to m-order tensor generate separated weighted matrices for each user, item and n-aspects.

Algorithm 1. Build m-order Tensor, X

- Input: Multi-aspect rating data
 Output: n-order tensor $X = \mathbb{R}^{k_1 \times k_2 \times \dots \times k_n}$
 Steps:
 i. Pivot mapping multi-aspect rating data to $U \times V \times K_n$
 ii. Fill up all missing ratings to 0
 iii. Metricize $U \times V \times K_n$
 iv. Return Tensor X

Algorithm 2: weighted-TF CP Decomposition of n-order tensor, X

- Input: n-order tensor $X = \mathbb{R}^{k_1 \times k_2 \times \dots \times k_n}$
 Output: weighted factor matrices of m-order tensor, X for K_{1-n}
 Steps:
 i. Initialized $K^1, K^2 \dots K^n$
 ii. Initialized $i = 1$
 iii. While not converged || i less than $n+1$ do
 iv. $K_i \leftarrow X_{(i)} (K_{(i+2)} \odot K_{(i+1)}) (K_{(i+2)}^T K_{(i+2)} \odot K_{(i+1)}^T K_{(i+1)})$
 v. Normalized column vectors of K_i
 vi. End while
 vii. Return $K_1, K_2 \dots K_n$ and core tensor, λ

Algorithm 3: Tensorized NN Training for rating prediction

- Input: $U \times V \times K_1 \dots \times K_n$
 Output: Tensorized Neural Network Model
 Steps:
 i. Initialized $U \times V \times K_1 \dots \times K_n$
 ii. Calculate $S = \sum_{r=1}^R (U V^T) \cdot (V \cdot K^T)$
 iii. Generate $wTF - u, wTF - v, wTF - k$
 iv. Training NN for
 Layer-64
 Layer-64-64
 Layer-128-64
 Layer-128-128
 Layer-128-64-32
 v. Calculate prediction accuracy

Before the weighted matrices are fed to the neural, network model, the interaction between users, items and aspect is calculated using formula (1) as follows:

$$S = \sum_{r=1}^R (U V^T) \cdot (V \cdot K^T) \tag{1}$$

where S represented the relation between users, U, item V and Aspect, K.



IV. EXPERIMENTAL SETTINGS

In this research, we present a tensor-based Neural Network framework which incorporates aspects ratings retrieved from reviews expressed by the user for an item. This section describes the experimental setup and results are presented.

A. Datasets

The dataset used to evaluate the proposed model have been widely used in related research for an aspect-based recommendation. This dataset is provided by [24] and originated from the real world dataset crawled from TripAdvisor (<http://www.tripAdvisor.com>) dated from 2002 to 2012. In this dataset, each user preference information on hotels was provided on eight predefined aspects in addition to the overall rating. The original data has a high sparsity rate. Therefore, to reduce the computational complexity and to produce a meaningful computation, we filter out an invalid user (e.g. Anonymous) and the user who provide less than 10 ratings and reviews. The result from data selection produces a much more condensed data which involve 5 586 users, 8 047 items and 86 369 ratings. The detail properties of the dataset are shown in Table 1.

B. Evaluation metrics

In this experiment, we examine the effectiveness of auxiliary information such as aspects from reviews influence user preferences on items. Therefore, we use Mean Absolute Error (MAE) and Mean Squared Error (MSE) error metrics to measure the difference between the predicted ratings from the model and the true ratings given by users. The definitions of these metrics are shown in (2) and (3) respectively.

$$MAE(U, V, K|k) = \frac{1}{|N_t|} \sum_{(u,v) \in N_t} |r_{uv} - \hat{r}_{uv(k)}| \quad (2)$$

$$MSE(U, V, K|k) = \frac{1}{|N_t|} \sum_{(u,v) \in N_t} (\hat{r}_{uv(k)} - r_{uv})^2 \quad (3)$$

where N_t is the number of ratings in the test set, $\hat{r}_{uv(k)}$ denotes the aspect-based rating prediction for user u and item v and r_{uv} represents the true rating values provided in the dataset.

The concern of this research is to precisely predict the ratings rather than shortlisting of interesting items to the users. Therefore, both MAE and MSE are employed to measure how much the predicted rating differs from the true rating. A lower value indicates a more accurate and better performance of the predictions.

C. Experimental settings

We conducted a series of the experimental procedure with various configuration setting of hyperparameter, input mapping and different distribution of training-test data.

Hyperparameter and input mapping

For the purpose of hyperparameter optimization, we evaluate the proposed framework using a different combination of hyperparameter and input mapping. We initialize 2 hyperparameters which are activation function, α ($\alpha = \text{'relu' | 'tanh'}$) and embedding size, λ ($\lambda = 10 | 50$). For the input mapping, ϕ we implement 'flatten' and 'reshape' mapping represented as 0 and 1 respectively.

Model hidden layers

We also evaluate the result of the proposed model using different layers with different regularization, δ settings. Table 2 summarizes the layer and regularization setting used in this research.

Training procedure

After hyperparameter tuning and optimization, we then select the best model to be compared with training-test in the ratio of 80:20 for training and test data. We also use 10-fold cross-validation with at least 50 epochs while monitoring the validation pattern and performance.

Baseline

For the baseline method, we compare the result of the proposed framework with a Neural Network with Multi-Aspect rating prediction, NNMA [25].

Table 1: Properties of the dataset

Attribute	Original	Filter
Number of users	781403	5586
Number of items	12746	8047
Number of ratings	1621956	86369
Range of ratings	1-5	1-5
Density of rating matrix	1.6%	19.21%

Table 2: Layer configuration

Model Layer	Regularization
Layer-64	$\delta_1 = 0.02$
Layer-64-64	$\delta_1 = 0.2, \delta_2 = 0.4$
Layer-128-64	$\delta_1 = 0.2, \delta_2 = 0.2$
Layers-128-128	$\delta_1 = 0.2, \delta_2 = 0.4$
Layers-128-64-32	$\delta_1 = 0.2, \delta_2 = 0.4, \delta_3 = 0.4$

V. RESULTS AND DISCUSSION

In this section, we present the result of the experiment conducted including the comparison with the baseline method.

For the preliminary implementation, we compare the performance of the proposed framework based on a different combination of hyperparameter setting as described in Section IV-C.

A. Hyperparameter and input mapping

When we execute different input mapping, ϕ which are flattened and reshape. Both mappings produce the same pattern of learning where NNMA shows overfitting and stuck at the local optimum. wTF-NNMA shows the same problem at the beginning of the training process but it starts to stabilize throughout the process clearly can be seen after for about 15 epochs. From this observation, the result shows that, in the training process, wTF-NNMA can achieve global optimum solution.

We observed the effect of number of embedding size of 10 and 50. The larger epochs shows more consistent results as shown in Fig. 1 and Fig. 2 for both NNMA and wTF-NNMA. During the training, the MAE reduced significantly when epochs achieve 8 for NNMA and at 10 for wTF-NNMA. However, the validation result for NNMA shows overfitting and it stuck at the local optimum. While for wTF-NNMA, the



overfitting problem also occurs at the beginning of the validation process, but it starts stabilizing to the global optimum solution after 12 epochs. From this observation, wTF-NNMA has training and validation consistency.

B. Model Hidden Layers

Table 3 shows the rating prediction result for the proposed model wTF-NNMA and the baseline method, NNMA. This result verifies that wTF-NNMA outperformed NNMA for all layers with different regularization settings. For different layer configurations, we observed small changes of MAE for NNMA. However, for overall performance of wTF-NNMA, it seems that the different mentioned layer settings do not significantly effect the MAE. This result shows that the complexity of model layers do affect NNMA but not significantly affect wTF-NNMA.

C. Overall performance

To explain in detail about the overall performance of both NNMA and wTF-NNMA, we observed the result from the training and validation processes for both model. Fig. 1 and Fig. 2 shows the training and validation MAE pattern for NNMA and wTF-NNMA respectively. From the result, both models show a good training pattern, however for the validation, NNMA shows overfitting with the local optimum for the whole validation processes with 50 epochs. wTF-NNMA also shows an overfitting problem at the early cycle of the validation. However, the MAE keep on deteriorating and start to stabilized after 12 epochs and the overfitting problem disappears. This result shows that wTF-NNMA is much better in order to produce a consistent result as compared to NNMA which stuck to a lot of local optimum.

Fig. 3 and 4 show the comparison between training and validation pattern for both NNMA and wTF-NNMA. During the training, NNMA shows better performance than wTF-NNMA with the lower MAE. However, in the validation process, NNMA show overfitting with the local optimum problem. wTF-NNMA start the convergence of global optimum at 12 epochs. This result shows the performance consistency of wTF-NNMA.

Table 3: MAE for model with different layer settings

Model	NNMA		wTF-NNMA	
Metric	MSE	MAE	MSE	MAE
Layer-64	2.3803	1.4030	1.1022	0.7785
Layer-64-64	2.9906	1.5858	1.1022	0.7783
Layer-128-64	3.0421	1.5952	1.0973	0.7855
Layers-128-128	3.0866	1.6124	1.1023	0.7793
Layers-128-64-32	3.0384	1.5920	1.1014	0.7795

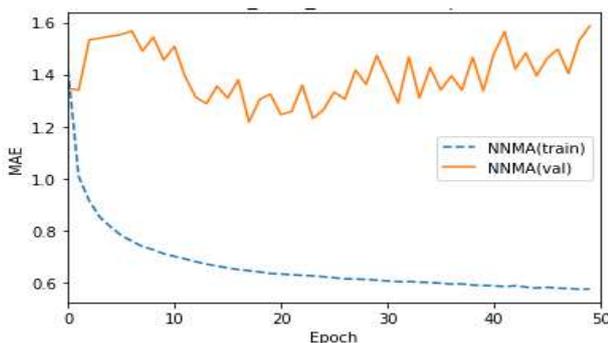


Fig. 1: Training and validation MAE for NNMA

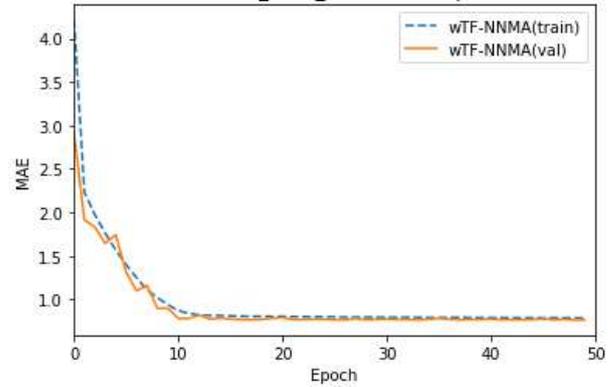


Fig. 2: Training and validation MAE for wTF-NNMA

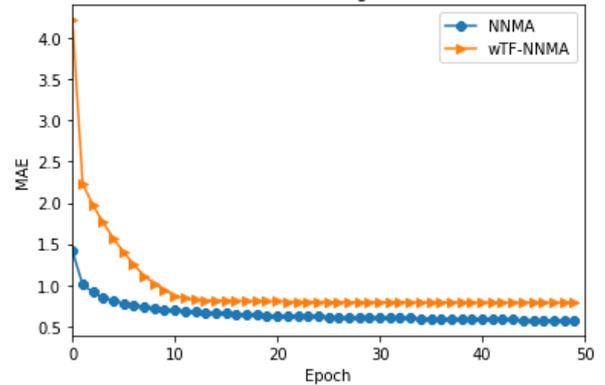


Fig. 3: Training MAE for NNMA and wTF-NNMA

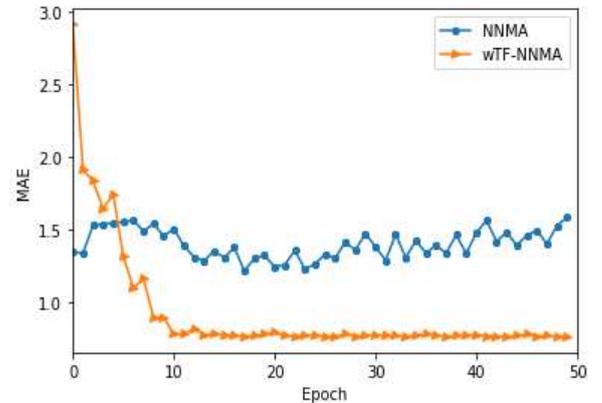


Fig. 4 Validation MAE for NNMA and wTF-NNMA

VI.CONCLUSION

We proposed a framework design and implementation of multi-aspect-based recommendation using weighted-based tensorized neural network approach called wTF-NNMA. We design the overall architecture of the recommendation with three steps recommendation algorithm. The performance and consistency of rating prediction are shown to improve by employing the Tensorized Neural Network framework and can overcome the problem of local optimum convergence for multi-aspect rating recommendation. For the future works, we plan to investigate on the other domain or dataset and to explore on the other popular approaches such as recurrent and convolutional neural network which has been proven an outstanding achievement in other research fields.



VII.ACKNOWLEDGMENT

This research has been supported by Research Grant: GP-IPS/2018/9621000 from Universiti Putra Malaysia.

REFERENCES

1. A. Tejada-Lorente, C. Porcel, J. Bernabe-Moreno, and E. Herrera-Viedma, "REFORE: A recommender system for researchers based on bibliometrics," *Appl. Soft Comput. J.*, 30, 2015, pp. 778–791.
2. F. Yang, "A hybrid recommendation algorithm – based intelligent business recommendation system system," *J. Discret. Math. Sci. Cryptogr.*, 21(6), 2018, pp. 1317–1322.
3. R. M. D. Addio, R. S. Marinho, and M. G. Manzato, "Combining different metadata views for better recommendation accuracy," *Inf. Syst.*, 83, 2019, pp. 1–12.
4. B. Purkaystha, T. Datta, and S. Islam, "Rating prediction for recommendation: Constructing user profiles and item characteristics using backpropagation," *Appl. Soft Comput. J.*, 75, 2019, pp. 310–322.
5. L. Luo, H. Xie, Y. Rao, and F. L. Wang, "Personalized recommendation by matrix co-factorization with tags and time information," *Expert Syst. Appl.*, 119, 2018, pp. 311–321.
6. S. M. Taheri and A. Movaghar, "Extracting implicit social relation for social recommendation techniques in user rating prediction," *26th Int. Conf. World Wide Web Companion*, 2017, pp. 1343–1351.
7. H. Song, Q. Pei, Y. Xiao, Z. Li, and Y. Wang, "A novel recommendation model based on trust relations and item ratings in social networks," *Int. Conf. Netw. Netw. Appl.*, 2017, pp. 17–23.
8. T. M. Phuong, D. T. Lien, and N. D. Phuong, "Graph-based context-aware collaborative filtering," *Expert Syst. Appl.*, 126, 2019, pp. 9–19.
9. M. H. Aghdam, "Context-aware recommender systems using hierarchical hidden Markov model," *Physica A*, 518, 2018, pp. 89–98.
10. J. I. N. Xie, F. Zhu, M. Huang, N. Xiong, and S. Huang, "Unsupervised learning of paragraph embeddings for context-aware recommendation," *IEEE Access*, 7, 2019, pp. 43100–43109.
11. I. Andr, A. Suniga, J. Donini, C. V. Sundermann, S. O. Rezende, and M. Aur, "Transforming geo-referenced data in contextual information for context-aware recommender systems," *IEEE/WIC/ACM Int. Conf. Web Intell.*, 2018, pp. 528–533.
12. P. Kefalas and Y. Manolopoulos, "A time-aware spatio-textual recommender system," *Expert Syst. Appl.*, 78, 2017, pp. 396–406.
13. D. Yu, Y. Mu, and Y. Jin, "Rating prediction using review texts with underlying sentiments," *Information Processing Letters*, 117, 2017, pp. 10–18.
14. X. Ma, X. Lei, G. Zhao, and X. Qian, "Rating prediction by exploring user's preference and sentiment rating prediction by exploring user's preference and sentiment," *Multimedia Tools and Applications*, 77, 2018, pp. 6425–6444.
15. N. Jing, T. Jiang, J. Du, and V. Sugumaran, "Personalized recommendation based on customer preference mining and sentiment assessment from a Chinese e-commerce website," *Electron. Commer. Res.*, 18(1), 2018, pp. 159–179.
16. A. Taneja and A. Arora, "Modeling user preferences using neural networks and tensor factorization model," *Int. J. Inf. Manage.*, 45, 2019, pp. 132–148.
17. Y. Ding, S. Li, W. Yu, J. Wang, and M. Liu, "A unified neural model for review-based rating prediction by leveraging multi-criteria ratings and review text," *Cluster Comput.*, 2018, pp. 1-9.
18. D. Yu, Y. Mu, and Y. Jin, "Rating prediction using review texts with underlying sentiments," *Inf. Process. Lett.*, 117, 2017, pp. 10–18.
19. C. Yang, X. Yu, Y. Liu, Y. Nie, and Y. Wang, "Collaborative filtering with weighted opinion aspects," *Neurocomputing*, 210, 2016, pp. 185–196.
20. Y. Nie, Y. Liu, and X. Yu, "Weighted aspect-based collaborative filtering," *37th International ACM SIGIR conference on Research and Development in Information Retrieval*, 2014, pp. 1071–1074.
21. P. Symeonidis, "Content-based dimensionality reduction for recommender systems," in *Data Analysis, Machine Learning and Applications. Studies in Classification, Data Analysis, and Knowledge Organization*, C. Preisach, H. Burkhardt, L. Schmidt-Thieme, and R. Decker, Eds. Berlin: Springer, 2008, pp. 619–626.
22. M. Nilashi, A. Ahani, M. D. Esfahani, E. Yadegaridehkordi, S. Samad, O. Ibrahim, N. M. Sharef, and E. Akbari, "Preference learning for eco-friendly hotels recommendation: A multi-criteria collaborative filtering approach," *J. Clean. Prod.*, 215, 2019, pp. 767-783.
23. N. Khairudin, N. M. Sharef, S. A. M. Noah, and N. Mustapha, "Embedded learning for leveraging multi-aspect in rating prediction of personalized recommendation," *Malaysian J. Comput. Sci.*, 2018: Special Issue December 2018: "Information Retrieval and Knowledge Management Special Issue Publication", 2018, pp. 31–47.
24. H. Wang, Y. Lu, and C. Zhai, "Latent aspect rating analysis without aspect keyword supervision," *17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2011, pp. 618–626.
25. N. Khairudin, N. M. Sharef, N. Mustapha, and S. A. M. Noah, "Enhancing multi-aspect collaborative filtering for personalized recommendation," *4th Int. Conf. Inf. Retr. Knowl. Manag.*, 2018, pp. 91–96.