Geojournal of Tourism and Geosites
ISSN 2065-0817, E-ISSN 2065-1198

Year XVIII, vol. 60, no. 2spl, 2025, p.1179-1187
DOI 10.30892/gtg.602spl15-1491

ATTENTIVE ITEM2VEC MACHINE LEARNING METHOD FOR RECOMMENDING TOURIST DESTINATIONS IN INDONESIA

Aisha Kusuma PUTRI [1], Gumgum DARMAWAN [1*], Budhi HANDOKO [1]

[1]Universitas Padjadjaran, Department of Statistics, Faculty of Mathematics and Natural Sciences, Bandung, Indonesia; aisha23004@mail.unpad.ac.id (A.K.P.); gumgum@unpad.ac.id (G.D.); budhi.handoko@unpad.ac.id (B.H.)

Citation: Putri, A.K., Darmawan, G., & Handoko, B. (2025). Attentive item2vec machine learning method for recommending tourist destinations in Indonesia. *Geojournal of Tourism and Geosites*, 60(2spl), 1179–1187. https://doi.org/10.30892/gtg.602spl15-1491

Abstract: In today's digital era, recommendation systems have been used by various age groups. One of the recommendation systems that many people often use is the YouTube application, and users usually get video recommendations related to previously watched videos. This is evidence that this recommendation system is close and helps human life today. This makes researchers compete to develop better recommendation systems. One of the recommendation systems that is considered to have good performance is the Attentive Item2Vec (AI2V) method. The AI2V method is a development of the Item2Vec (I2V) method, which combines collaborative filtering with a neural network to get a recommendation. The advantages of AI2V are that it pays attention to the entire sequence of items (videos, images, etc) consumed by the user, the complex relationships between all items in the user's sequence, and can be applied in various recommendation systems. In this study, the AI2V method was applied to tourist assessment data on several tourist destinations in Indonesia (Jakarta, Yogyakarta, Bandung, Semarang and Surabaya). The data consists of 10,000 assessments given by 300 tourists. The AI2V methods include data preparation, data splitting, Attentive Context-Target Representation, Multi-Attentive User Representation, AI2V similarity function, Top 5 Recommendations, and performance evaluations. Based on the analysis used, AI2V produces the Top 5 Recommendations for tourist destinations, namely Keraton Surabaya, Desa Wisata Gamplong, Sanghyang Heuleut, Jogja Bay Pirates Adventure Waterpark, and Jogja Exotarium. The accuracy level with the Mean Percentage Ranking (MPR) is 0.48976, meaning that the recommended results for all tourists are considered reasonably good performance. The parameters used in the study include a learning rate (α) of 0.5, a window size of 500, and a max epoch of 50. The parameters that have been determined and the process run are expected to provide accurate recommendations and match the interests of tourists.

Keywords: attentive item2vec, destination, item2vec, tourism, tourist, recommendation system

* * * * * *

INTRODUCTION

Machine learning is a part of artificial intelligence that can process big data to make fast and accurate decisions in various aspects of life (Hou et al., 2020; Injadat et al., 2021). Machine learning has been studied since the late 1950s and has recently been increasingly developed by researchers (Badillo et al., 2020). One of the machine learning methods widely used by the public and companies today is the recommendation system. Large companies often use recommendation systems that offer goods and services, such as Google, Netflix, YouTube, Spotify, TikTok, etc (Bartlett et al., 2023; Jannach et al., 2022; Sánchez-Corcuera et al., 2024). These companies use recommendation systems to make it easier for customers to choose items (for example, goods, images, videos, sounds, and so on) (Ahmed et al., 2022; Geetha & Renuka, 2024; Lee et al., 2020; Tareq et al., 2020). The recommendation system has been introduced since the 1990s, and along with the development of science, researchers have studied the use of machine learning in recommendation systems (Bhareti et al., 2020). Recommendation systems play a major role in promotion. Therefore, companies rely on this system to support sales success (Stalidis et al., 2023). Before recommendations were developed using machine learning in recommendation systems, recommendations initially relied only on basic statistics to determine recommendations (Shan et al., 2019).

Recommendations are made by calculating the average rating score of an item and recommending items based on the highest average rating score (Gupta & Kant, 2020). Then, the recommendations continue to develop and can be calculated using correlation analysis (Wang et al., 2023). Recommendations using this simple approach are less commonly used because they cannot capture complex user behavior patterns in the data (Cui et al., 2020; Li et al., 2021; Tahmasbi et al., 2021). In this case, it is complex, including cost factors, location, etc.

In addition, correlation analysis requires a sufficient sample size to produce accurate estimates, while user data is not always available with a sufficient sample size (Andrade, 2020). Therefore, in getting recommendations, it is highly recommended that machine learning or so-called recommendation systems be used to provide more effective and appropriate recommendations (Alfaifi, 2024). The recommendation system has advantages, including allowing the system to continue to adapt to changes in user preferences and allowing it to make relevant recommendations. Machine learning allows the system to capture relationships that simple recommendations cannot explain.

[*] Corresponding author

There are two recommendation system types, content-based and collaborative filtering (Alamdari et al., 2020; Sharma et al., 2022). Among the two recommendation systems, collaborative filtering is a popular and widely used recommendation system (Gupta et al., 2020). Collaborative filtering (CF) is a recommendation system based on explicit feedback, such as the rating score given by the user (Chen et al., 2020; Patoulia et al., 2022). Neural networks in CF are used to predict the value of each item as a number (He et al., 2015). In predicting the value of each item, a weighting technique is used with optimization to obtain a vector representation of numbers for the appropriate item. One combination of CF with a neural network in a recommendation system is the Item2Vec (I2V) method. The I2V method was designed by Barkan and Koenigstein in 2016. In previous studies, I2V had an accuracy of 68% (Barkan & Koenigstein, 2016) and an accuracy of 61% (Putri & Suliadi, 2023). This accuracy is considered a poor performance, so in 2020, Barkan et al. developed I2V into Attentive Item2Vec (AI2V). Like I2V, AI2V also represents items in a vector of numbers to capture various behavioural patterns in the item history and get items that might be recommended. AI2V has outperformed previous methods, including I2V (Barkan et al., 2020). The advantages of AI2V compared to I2V are that AI2V pays attention to the entire sequence of items consumed by the user, pays attention to the relationship of all more complex items using the attention mechanism and can be applied in various types of recommendation systems (e.g., product recommendations, music content and others). Previous studies related to AI2V include Barkan et al. (2020) about AI2V, then Gaiger et al. (2023) about AI2V++. In addition, there is research from Putri & Suliadi (2023) who studied the Item2Vec (I2V).

In this study, AI2V is applied to the tourism sector to generate destination recommendations. The application of AI2V in the tourism sector is a relatively new approach, considering that previously this method was more widely used in recommendation systems in the entertainment sector, such as movies and music. The adaptation of AI2V for tourism recommendation systems aims to capture tourist preferences for destinations, which have different characteristics compared to entertainment items. By expanding the application of AI2V to the tourism sector, this study opens up new opportunities for the development of AI2V-based recommendation systems in various application areas. The data used is the assessment data of tourist destinations in Indonesia. Based on this assessment, AI2V will recommend the five best tourist destinations (Top 5 Recommendations) based on previous tourist destination visit records. In the field of tourism, this research provides a significant contribution to the development of tourist destinations in Indonesia, especially in Jakarta, Yogyakarta, Bandung, Semarang and Surabaya. So far, Bali has been the most well-known destination, both by domestic and foreign tourists. Therefore, efforts are needed to promote other tourist destinations in Indonesia so that Bali is not the only one in the spotlight. This research is expected to be one of the strategic steps to increase the popularity and competitiveness of other tourist destinations. So the study limits the scope to the data of tourist destination assessment in Indonesia for the cities of Jakarta, Yogyakarta, Bandung, Semarang and Surabaya. In addition, this study focuses on the application of the Attentive Item2Vec (AI2V) method. And finally, the results of this study are in the form of general recommendations for tourists.

MATERIALS AND METHODOLOGY

This study applies a combination of neural networks with collaborative filtering, namely Attentive Item2Vec (AI2V), to represent tourist behaviour patterns so that tourist destination recommendations can be found for tourists. AI2V is a method that studies dynamically changing user patterns with various potential new recommendations (Barkan et al., 2020). The data in this study are assessment data from tourists at several tourist destinations in Indonesia (Jakarta, Yogyakarta, Bandung, Semarang and Surabaya). This data was obtained from the Kaggle site uploaded by the Getloc Team in July 2021 (Prabowo et al., 2021). The data consists of 10,000 assessments given by 300 tourists. The evaluation was carried out by tourists using a rating score of 1 (very dislike), 2 (dislike), 3 (moderate), 4 (like), and 5 (very like). The main instrument in this study is the AI2V method, which is applied to the assessment data of tourist destinations in Indonesia. Figure 1 is a flowchart of the AI2V method to analyse tourist destination assessment data in Indonesia (Barkan et al., 2020).

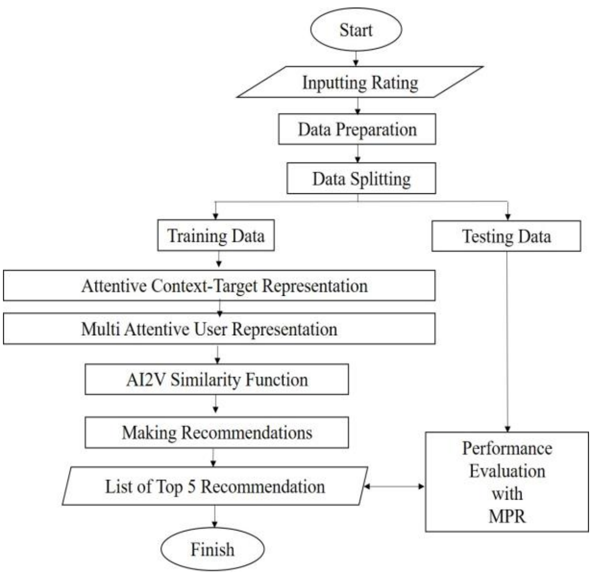

Figure 1. Flowchart of the Attentive Item2Vec (AI2V)

The process of AI2V starts from the “Start”. Then, “Inputting Rating”, where the rating is a list of users and the assessment given to an item. Next, the data is processed in the “Data Preparation” stage to ensure the data is ready to use. After that, “Data Splitting” is carried out to divide the data into “Training Data” and “Testing Data”. Training Data is used to form “Attentive Context-Target Representation” in studying the relationship between context and target items. Next, the “Multi-Attentive User Representation” stage is used to build a more in-depth user representation. Then, the system applies the “AI2V Similarity Function” to calculate the similarity between items or users.

Based on these results, the system makes recommendations through the “Making Recommendations” and produces a “List of Top 5 Recommendations”. On the other hand, “Testing Data” is used to perform “Performance Evaluation with MPR” to evaluate the performance of the Top 5 Recommendations. After going through all the stages, the process ends with “Finish” with the results being Top 5 Recommendations and MPR scores.

o support data processing in this study, the Python language is used on Google Colab based on the GitHub link by the following Kerengaiger account https://github.com/kerengaiger/ai2v/tree/master/models.

DATA ANALYSIS

This section describes the stages of the AI2V analysis process, complete with explanations and accompanying equations. This description refers to the flow shown in Figure 1. The analysis consists of seven subsections and includes a total of 22 equations. The entire series of seven stages produces outputs in the form of recommendations and MPR scores.

1. Data Preparation

Data preparation is transforming original data into data with a more suitable structure for analysis using the method used (Hameed & Naumann, 2020). In the AI2V data preparation stage, the assessment is used with rating scores of 4 (like) and 5 (very like) (Barkan et al., 2020). Next, filtering all tourists with fewer than three tourist destinations (Barkan et al., 2020). This is done so that the data prepared has sufficient information and can capture tourist behaviour patterns. The data structure for applying AI2V is shown in Table 1.

Table 1. Data Structure for Attentive Item2Vec (AI2V)

Tourist	Destination Code
1	5,101,258,20,393,208,405,41,336,67,246,265,307
2	2,322,85,78,437,407,208,437,384,426,18
⋮	⋮
150	4,220,320,28,333,311,51,332,411,83,14,371,292,81,239,427,28,341
⋮	⋮
300	69,200,285,343,144,193,435,346,255,416,108,103,64,279

2. Data Splitting

Data splitting is separating a data set into training and testing data. In AI2V, the proportion of training and testing data is determined by taking the last destination in each user list to be used as test data, and the rest is training data (Barkan et al., 2020).

3. Attentive Context-Target Representation

Attentive context-target representation is a vector representation for context destinations and target destinations attentively. The target destination is the destination for which recommendations will be sought, while the context destination is the historical destination that is on the same list as the target destination. For example, in Table 1, the first tourist, if the target destination (l_j) is a destination with code 307, then the context destination (l_m) is a destination with code 5 to destination code 265. Equation (1) is used to model attentive context-target representation (Barkan et al., 2020) .

$$\boldsymbol{a_{j-1}} = \sum_{m=1}^{L} c_{jm} \boldsymbol{Bc}\, \boldsymbol{u}_{l_m} \quad (1)$$

We will obtain $\boldsymbol{a_{j-1}} = (a_{j-1}^{1}, \dots, a_{j-1}^{N})$. a_{j-1} shows the historical representation of each u tourist, denoted by $\boldsymbol{\alpha_{j-1}^{u}}$. $\boldsymbol{u_{l_m}}$ is the input data context formed using one-hot encoding. One-hot encoding is a vector of size $L \times 1$ containing the values 1 and 0. L measures the number of tourist destinations for tourists u in the training data. In one-hot encoding, the position of the number 1 is determined based on the position of the context destination (l_m) on the tourist list.

For example, $\boldsymbol{u_{l_m}}$ for the first tourist, if the destination m is in the first position on the list, then the number 1 is placed in the first row. Here is $\boldsymbol{u_{l_1}}$ for the first tourist.

$$\boldsymbol{u_{l_1}} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Where $\boldsymbol{u}_{l_m} = \{xc_1, xc_2, \dots, xc_m, \dots, xt_L\}$, with xc_m is the value of the m-th context layer input, and xt_L is the destination target. Meanwhile, $\boldsymbol{Bc}$ does a linear transformation obtain a learnable linear mapping. Weight is a linear transformation that the neural network must learn. So, $\boldsymbol{Bc}$ does the neural network learn the weight during training. Based on this, determining $\boldsymbol{Bc}$ involves the neural network process below.

a. Feedforward

The feedforward process consists of input, hidden, and output layers. In this stage, the input layer is namely xc_m, the value of the m-th context layer input. The hidden layer calculation is done using equation (2) (Wadi, 2021).

$$h_m^{Bc} = \sum_{p=1}^{L}\sum_{m=1}^{L}\left(Bc_{pm} \times xc_m\right) \tag{2}$$

Bc_{pm} is the elements in the p-th row and m-th column of the weight matrix $\boldsymbol{Bc}$, while xc_m is the value of the m-th context layer input. The weight matrix $\boldsymbol{Bc}$ is first obtained from initialization with random values 0 to 0.05 with a matrix size of $L \times L$, L is the row size of $\boldsymbol{u_{l_m}}$. $\boldsymbol{u_{l_m}}$ is the input data context formed using one-hot encoding. One-hot encoding is a vector of size $L \times 1$ containing the values 1 and 0. If the calculation is done in matrix form, equation (2) becomes equation (3) (Wadi, 2021).

$$\boldsymbol{h^{Bc}}_{(\boldsymbol{L\times 1})} = \boldsymbol{Bc} \times \boldsymbol{u_{l_m}} \tag{3}$$

After obtaining the hidden layer, the output layer is calculated using equation (4) (Wadi, 2021).

$$\widehat{y_m^{Bc}} = \sum_{q=1}^{L}\sum_{m=1}^{L}\left({Bc'}_{qm} \times h_m^{Bc}\right) \tag{4}$$

Where ${Bc'}_{qm}$ is the element in the q-th row and m-th column of the weight matrix $\boldsymbol{Bc'}$, while h_m^{Bc} is the value of the m-th hidden layer context. The weight matrix $\boldsymbol{Bc'}$ is first obtained from initialization with a value of 0 to 0.05 with a matrix size of $L \times L$. If in matrix form, equation (4) becomes equation (5) (Wadi, 2021).

$$\widehat{\boldsymbol{y^{Bc}}}_{(\boldsymbol{w\times 1})} = \boldsymbol{f}(\boldsymbol{Bc'} \times \boldsymbol{h^{Bc}}) \tag{5}$$

f is the softmax activation function calculated using equation (6) (Wadi, 2021).

$$\boldsymbol{f}(\boldsymbol{Bc'} \times \boldsymbol{h^{Bc}}) = \frac{\boldsymbol{e}^{(\boldsymbol{Bc'}\times \boldsymbol{h^{Bc}})_{\boldsymbol{m}}}}{\sum_{\boldsymbol{m=1}}^{\boldsymbol{L}} \boldsymbol{e}^{(\boldsymbol{Bc'}\times \boldsymbol{h^{Bc}})_{\boldsymbol{m}}}} \tag{6}$$

Next, the Binary Cross-Entropy (BCE) loss function is calculated using equation (7). The BCE calculation determines whether the weights are optimal and whether the training process should stop or undergo backpropagation.

$$E^{Bc} = -\frac{1}{L}\sum_{m}^{L}\left[\, y_m^{Bc}\, log\left(\widehat{y_m^{Bc}}\right) + (1 - y_m^{Bc}) log(1 - \widehat{y_m^{Bc}})\right] \tag{7}$$

With y_m^{Bc} is the target value where t is given a value of 0 because of the destination context, while $\widehat{y_m^{Bc}}$ is the softmax activation value of the i-th context layer output. The criteria are if the E^{Bc} value is less than or equal to the error tolerance (α), then the training process stops. At the same time, if the E^{Bc} value is greater than the error tolerance (α), then the backpropagation process is needed.

b. Backpropagation

In the backpropagation process, weight calculations are carried out to obtain optimal weights and a minimum error rate. First, the weight calculation connects the output layer with the hidden layer, called the second weight ($\boldsymbol{Bc'}$), using equation (8) (Wadi, 2021).

$${Bc'}_{qm}(new) = {Bc'}_{qm} + \Delta {Bc'}_m \tag{8}$$

If the calculation is carried out in matrix form, equation (8) becomes equation (9) (Wadi, 2021).

$$\boldsymbol{Bc'}(\boldsymbol{new})_{(\boldsymbol{L\times L})} = \boldsymbol{Bc'} + \Delta\boldsymbol{Bc'} \tag{9}$$

is the element in the q-th row and m-th column of the weight matrix Bc', while $\Delta {Bc'}_m$ is the weight correction that connects the output of the m-th context layer with the m-th hidden context layer obtained from equation (10) (Wadi, 2021).

$$\Delta {Bc'}_m = \alpha \delta_m^{Bc} h_m^{Bc} \tag{10}$$

Where $\alpha > 0$ does the researcher determine the learning rate and h_m^{Bc} is the hidden layer context value obtained in the feedforward section. At the same time, δ_m^{Bc} is the correction factor from the m-th output layer context. δ_m^{Bc} is obtained from the calculation with equation (11) (Wadi, 2021).

$$\delta_m^{Bc} = (t_m^{Bc} - \widehat{y_m^{Bc}})\widehat{y_m^{Bc}}(1 - \widehat{y_m^{Bc}}) \tag{11}$$

With t_m^{Bc} is the target output on the output layer context, which in the target destination is given a value of 1 and the destination context is given a value of 0. At the same time, $\widehat{y_m^{Bc}}$ is the output layer context value obtained in the feedforward section. After the calculation of the $\boldsymbol{Bc}$ weight updates and the $\boldsymbol{Bc}(\boldsymbol{new})$ weight is obtained, the next step is to calculate the $\boldsymbol{Bc}$ weight using equation (12) (Wadi, 2021).

$$Bc_{pm}(new) = Bc_{pm} + \Delta Bc_m \tag{12}$$

Bc_{pm} is an element in the p-th row and m-th column of the weight matrix $\boldsymbol{Bc}$. At the same time, ΔBc_m is a weight correction that connects the m-th hidden layer context with the m-th input layer context. If the calculation is done in matrix form, equation (12) becomes equation (13) (Wadi, 2021).

$$\boldsymbol{Bc}(\boldsymbol{new})_{(\boldsymbol{L\times L})} = \boldsymbol{Bc} + \Delta\boldsymbol{Bc} \tag{13}$$

The value of ΔBc_m is obtained from the calculation of equation (14) (Wadi, 2021).

$$\Delta Bc_m = \alpha \delta h_m^{Bc} xc_m \tag{14}$$

With $m = 1, \ldots, L$. While $\alpha > 0$ does the researcher determine the learning rate and xc_m is the input of the m-th context layer. δh_m^{Bc} is the correction factor of the m-th context layer hidden obtained from equation (15) (Wadi, 2021).

$$\delta h_m^{Bc} = \delta o_m^{Bc} h_m^{Bc}(1 - h_m^{Bc}) \tag{15}$$

Where h_m^{Bc} is the value of the m-th hidden layer obtained in the feedforward section, at the same time, $\delta o_m^{Bc}$is the correction factor signal from the context layer output to the context hidden layer. δo_m^{Bc} is obtained from the calculation with equation (16) (Wadi, 2021).

$$\delta o_m^{Bc} = \sum_{m=1}^{L} Bc'_{qm}\, \delta_m^{Bc} \tag{16}$$

Bc'_{qm} is the element in the q-th row and m-th column of the weight matrix Bc', while δ_m^{Bc} is the correction factor of the m-th context layer output. After the $\boldsymbol{Bc}$ weight update calculation is performed and the $\boldsymbol{Bc(new)}$ weight is obtained, the next step is to perform the feedforward calculation again until it produces E^{Bc} that meets the researcher's criteria. While c_{jm} is the attention weight to emphasize further relevant destinations (target destinations), attention weights are calculated using equation (17) (Barkan et al., 2020).

$$c_{jm} = \frac{exp(h\left(\boldsymbol{Ac}\,\boldsymbol{u}_{l_m}, \boldsymbol{At}\,\boldsymbol{v}_{l_j}\right))}{\sum_{k=1}^{m-1} exp(h\left(\boldsymbol{Ac}\,\boldsymbol{u}_{l_k}, \boldsymbol{At}\,\boldsymbol{v}_{l_j}\right))} \tag{17}$$

With $m = 1, 2, \ldots, j$ and $k = 2, 3, \ldots, m-1$. Where $\boldsymbol{At}$ and $\boldsymbol{Ac}$ are each learnable linear mapping just like $\boldsymbol{Bc}$. So, $\boldsymbol{At}$ and $\boldsymbol{Ac}$ does the neural network learn the weights during the training process. Based on this, the process of determining $\boldsymbol{At}$ and $\boldsymbol{Ac}$ involves the neural network process. The stages carried out are the same as $\boldsymbol{Bc}$, with the difference that $\boldsymbol{At}$ and $\boldsymbol{Ac}$ are weight matrices measuring $w \times L$. The size of w, or called the window size, can be worth 50 to 1000, and there is no definite rule in determining w; the more significant the w, the better the results obtained, but it will slow down the processing time (Af'idah et al., 2021). The number of w is a characteristic of a destination, so if w is worth 50, then the number 50 is a number that explains or defines a destination. Barkan & Koenigstein (2016) used w of 100 for data measuring 2,500 to 20,000. Meanwhile, the size of L is the row size of $\boldsymbol{v}_{l_j}$ or $\boldsymbol{v}_{l_m}$. For the feedforward process, the size of the $\boldsymbol{At'}$ and $\boldsymbol{Ac'}$ matrix is $w \times w$.

4. Multi-Attentive User Representation

Multi-attentive user is a comprehensive representation of all tourists based on target destinations previously determined using equation (18) (Barkan et al., 2020).

$$\boldsymbol{z_{j-1}} = \boldsymbol{S}\boldsymbol{w}_{j-1} \tag{18}$$

Where $\boldsymbol{w}_{j-1} = [\left(\boldsymbol{a}_{j-1}^1\right)^T, \ldots, \left(\boldsymbol{a}_{j-1}^N\right)^T]^T$ and $\boldsymbol{S}$ are learnable linear mappings. Like $\boldsymbol{Bc}$, $\boldsymbol{At}$, and $\boldsymbol{Ac}$, $\boldsymbol{S}$ is obtained from the neural network process. The steps taken are the same as $\boldsymbol{Bc}$, $\boldsymbol{At}$, and $\boldsymbol{Ac}$, with the difference that $\boldsymbol{S}$ is a weight matrix measuring $L \times LN$. L is the row size of $\boldsymbol{v}_{l_j}$ or $\boldsymbol{v}_{l_m}$, while N is the size of the number of tourists.

5. AI2V Similarity Function

After calculating the Attentive context-target representation and Multi-Attentive User Representation, the next step is calculating the AI2V Similarity Function using equation (19) (Barkan et al., 2020).

$$o\left(z_{j-1}, v_{l_j}\right) = \psi_o\left(z_{j-1}, Bt, v_{l_j}\right) + b_{l_j} \tag{19}$$

Where ψ_o is the neural scoring function calculated in equation (20) (Barkan et al., 2020). b_{l_j} is the target bias term for the item l_j. The bias is set to a small value, such as 0.1 (Goodfellow et al., 2016).

$$\psi_o\left(z_{j-1}, B_t v_{l_j}\right) = \boldsymbol{z_{j-1}}^{\boldsymbol{T}} \times \boldsymbol{Bt} \times \boldsymbol{v_{l_j}} \tag{20}$$

Equation (20) is a hidden-layer neural network with scalar output. $\boldsymbol{z_{j-1}}^{\boldsymbol{T}}$ is a transpose of a multi-attentive user. $\boldsymbol{Bt}$ is a learnable linear mapping obtained from the neural network. The steps taken are the same as $\boldsymbol{Bc}$, the difference is the input, namely $\boldsymbol{v_{l_j}}$.

6. Making Recommendations

Recommendations are obtained by sorting the AI2V similarity function values in equation (20). The similarity values generated between $\boldsymbol{v_{l_j}}$ and $\boldsymbol{u_{l_m}}$are then sorted from the largest to the smallest. The greater the similarity value, the greater the relationship. The destination paired with the five highest similarity values will enter the Top 5 Recommendations of tourist destinations in Indonesia. Then, to complete the information from Table 2 we will find out the names and cities for each destination code.

Table 2. Description of Destination

Destination Code	Destination name	City
1	Monumen Nasional	Jakarta
2	Kota Tua	Jakarta
⋮	⋮	⋮
436	Taman Flora Bratang	Surabaya

7. Performance Evaluation

Testing data is used to evaluate the performance of the AI2V model. Recommendation performance can be evaluated using the Mean Percentage Ranking (MPR). MPR is the average percentile ranking for each sample with user satisfaction based on the recommendation list (Ginzburg et al., 2021). The calculation begins with the calculation of the percentile ranking for each tourist. If the destination from the testing data (l_t) is found in the recommendation list, then the calculation is carried out using Equation (21) (Gaiger et al., 2023). For the same destination for different tourists, the same order is given.

$$PR_t = \left(\frac{Position\ l_t}{T}\right) \tag{21}$$

With $t = 1,2, \dots, T$, in this case T is the number of tourists in the testing data. While the position (l_t) is the order of destinations in the overall recommendation list and N is the number of tourists. After calculating the PR for each tourist in the testing data (t), the next step is to calculate the overall MPR using Equation (22) (Gaiger et al., 2023).

$$MPR = \frac{\sum_{t=1}^{T} PR_t}{T} \tag{22}$$

Where $\sum_{t=1}^{T} PR_t$ is the sum of the percentile ranks for each t. The criteria for a good MPR value are those below 0.5 (Gaiger et al., 2023).

RESULTS AND DISCUSSIONS

This section displays all the results of the AI2V process. As mentioned earlier, this study uses the Python language and Google Colab. Before the AI2V method analysis section, a descriptive analysis was first carried out to see the distribution of tourist destination data. Based on Figure 2, the tourist destination with the highest frequency of visits is Keraton Surabaya, Sanghyang Heleut, Bukit Jamur, Desa Wisata Gamplong, Bukit Jamur, Jogja Bay Pirates Adventure Waterpark, Jogja Exotarium, Geofest Watu Payung Turunan, Taman Sungai Mudal and Kampung Batu Malakasari.

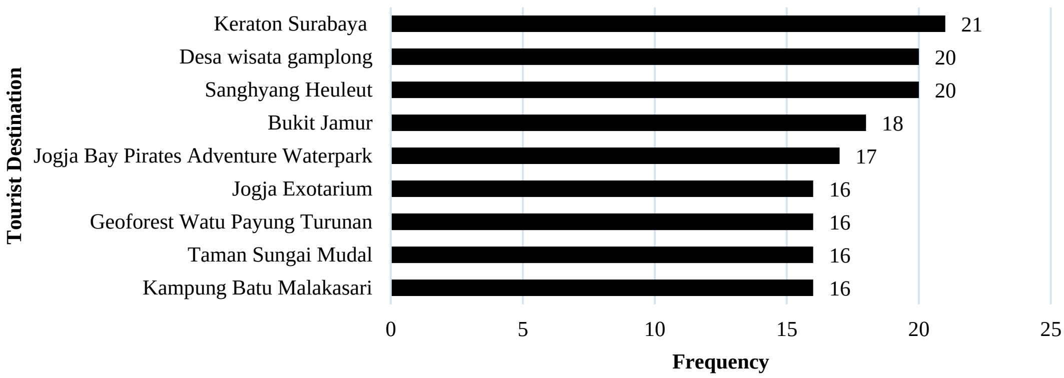

Figure 2. 10 Tourist Destinations with the Most Visiting Frequency

The results will later validate the results obtained using Python. In AI2V analysis, selecting optimal parameters such as window size, max epoch, and learning rate is very important to produce maximum model performance.

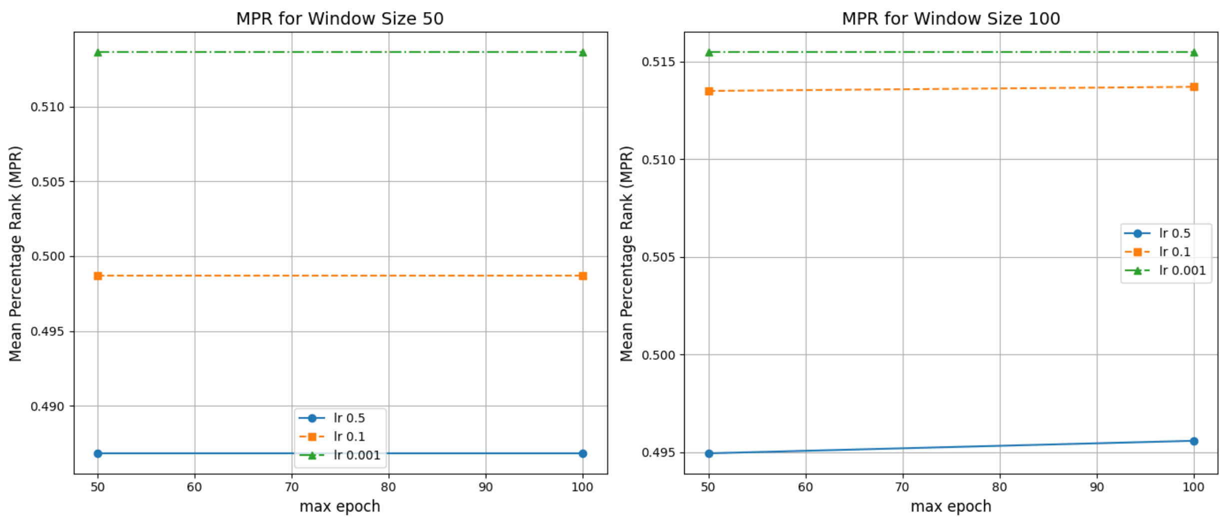

Figure 3. Mean Percentage Rank (MPR) For Window Size 50 and 100

First, comparing the Mean Percentage Rank (MPR) values at two window sizes ($w = 50{,}100$), which are shown in Figure 3. The first graph (left) shows the results with $w = 50$, while the second graph (right) shows the results with $w = 100$. At $w = 50$, all variations in learning rate ($\alpha = 0.5, 0.1, 0.001$) produce relatively stable MPR values even though the number of epochs increases. Meanwhile, at $w = 100$, there is a more significant increase in MPR, especially at $\alpha = 0.5$ as the number of epochs increases. Based on these results, it can be concluded that increasing max epoch tends to keep the MPR value constant or slightly increase. However, if observed further, max epoch 50 produces the lowest MPR value compared to larger max epochs. Given that the purpose of this study is to obtain the lowest possible MPR value, a max epoch of 50 is chosen as the most optimal parameter.

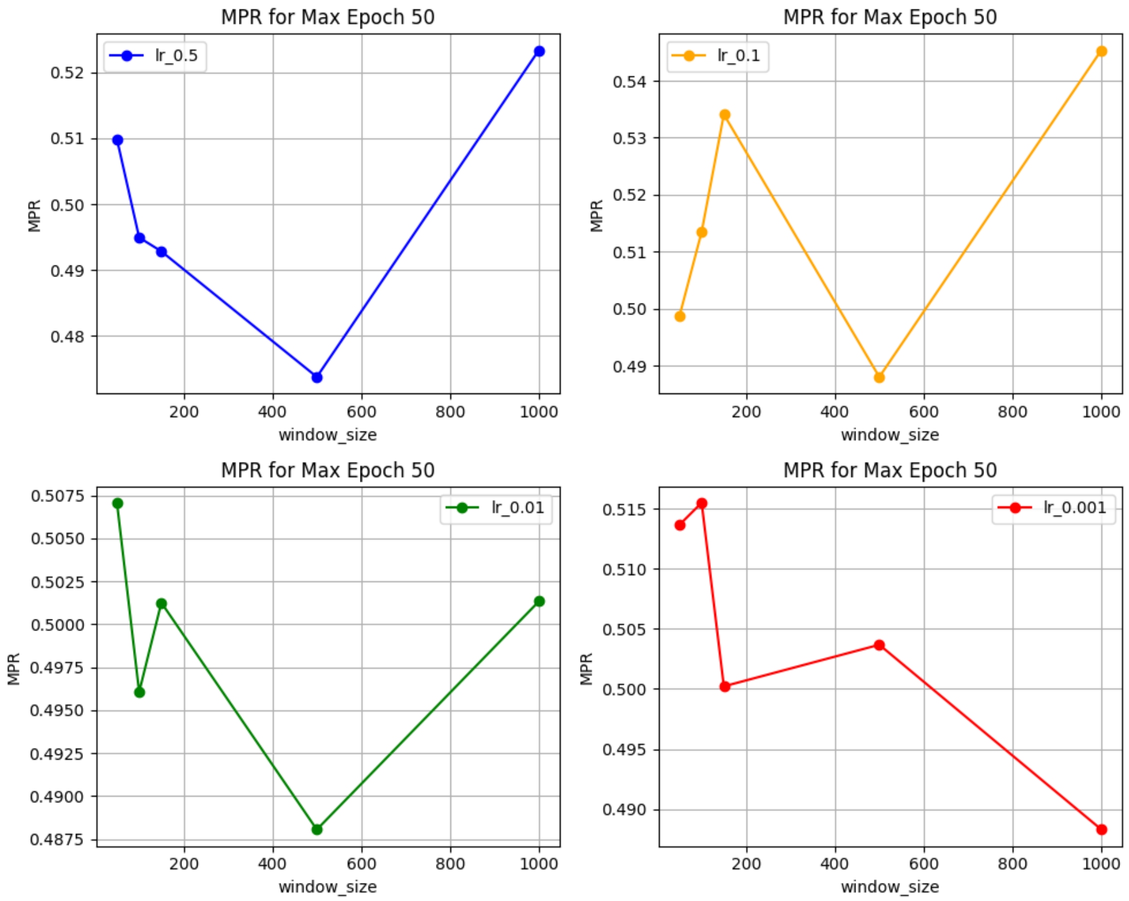

Figure 4. Mean Percentage Rank (MPR) For Max Epoch 50

Based on Figure 4, the effect of window size (w variations on the MPR value at max epoch 50 with four different levels of $\alpha = 0.5, 0.1, 0.01, 0.001$. At $\alpha = 0.5$ (top left), the MPR reaches its lowest point at $w = 500$ with an MPR below 0.48, it then increases gradually until it reaches its peak at $w = 1{,}000$ with an MPR above 0.52. A similar pattern is seen at $\alpha = 0.1$ (top right), where MPR reaches its lowest point when $w = 500$ with MPR below 0.49 and then increases until $w = 1{,}0000$. While at $\alpha = 0.01$ (bottom left), MPR reaches its lowest value at $w = 500$ with MPR below 0.49, then it increases slowly until $w = 1{,}000$. Meanwhile, $\alpha = 0.001$ (bottom right) shows a more stable downward trend, with the lowest MPR at $w < 200$ with MPR of 0.5. Overall, it can be concluded that MPR tends to reach its lowest point when $w = 500$, then increases again as the window size increases. Based on this, $w = 500$ is optimal because it has the smallest MPR value for $\alpha = 0.5, 0.1, 0.01$. In addition, $\alpha = 0.5$ was also chosen in this analysis because it has the smallest MPR value among the other α, namely below 0.48.

Table 3. Top 5 Recommendation Tourist Destinations in Indonesia

No	Destination Code	Destination Name
1	416	Keraton Surabaya
2	300	Sanghyang Heuleut
3	134	Desa Wisata Gamplong
4	183	Jogja Bay Pirates Adventure Waterpark
5	138	Jogja Exotarium

Top 5 recommendations in Table 3 from the AI2V method. The Top recommendations include Keraton Surabaya, Sanghyang Heuleut, Desa Wisata Gamplong, Jogja Bay Pirates Adventure Waterpark and Jogja Exotarium. The five destinations were designated Top recommendations because they have the largest similarity score. These destinations are expected to be recommendations for tourists who want to travel to a place. The MPR accuracy value is 0.48976, this value shows that the recommendations are pretty good and can provide recommendations that follow the interests of tourists.

CONCLUSIONS

Based on the results and discussion, the Top 5 Recommendations for tourist destinations in Indonesia are obtained along with the MPR value to measure the recommendations' accuracy. The parameters used in the study include a learning rate (α) of 0.5, a window size of 500, and a max epoch of 50. The parameters that have been determined and the process run are expected to provide accurate recommendations and match the interests of tourists.

The Top 5 Recommendations generated in this study are Keraton Surabaya, Sanghyang Heuleut, Desa Wisata Gamplong, Jogja Bay Pirates Adventure Waterpark and Jogja Exotarium. It is hoped that implementing AI2V can help many tourists who often feel confused when choosing tourist destinations in Indonesia, especially Jakarta, Yogyakarta, Bandung, Semarang, and Surabaya. In addition, these recommendations also provide strategic advantages for travel agents in offering tourist destinations that are more in line with the desires of tourists to increase sales opportunities and customer satisfaction. The level of accuracy using the Mean Percentage Ranking (MPR) of the Top 5 AI2V recommendations generated is 0.48976. This value is below 0.5, meaning that the recommendation results for all tourists are assessed as quite good performance and by tourist interests. Suggestions for further research are adding other data and variables that can influence the recommendation results, such as user domicile, economic level, and other factors.

In addition, using tourist behavioral data, such as search patterns, activity preferences, or duration of visits to certain tourist destinations, can be an interesting approach. This combination is expected to produce a more relevant, accurate recommendation system that meets the specific needs of each tourist.

This research is expected to contribute to increasing insight, knowledge, and information for researchers and academics regarding recommendation systems using the Attentive Item2Vec (AI2V) method.

Author Contributions: Conceptualization, A.K.P and G.D and B.H.; methodology, A.K.P and G.D and B.H.; software, A.K.P.; validation, A.K.P and G.D and B.H.; formal analysis, A.K.P and G.D and B.H.; investigation, A.K.P and G.D and B.H.; data curation, A.K.P and G.D and B.H.; writing - original draft preparation, A.K.P.; writing - review and editing, A.K.P and G.D and B.H; visualization, A.K.P.; supervision, A.K.P and G.D and B.H.; project administration, A.K.P and G.D and B.H. All authors have read and agreed to the published version of the manuscript.

Funding: 1/UN6.RKT/Kep/HK/2025

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study may be obtained on request from the corresponding author.

Acknowledgements: The author expresses gratitude to Universitas Padjadjaran for the support and funding provided for the publication of this thesis, with the thesis number "T.302/AIS/PROG/STAT/2025" This assistance is highly valuable in facilitating the completion of this research and contributing to the author's academic work.

Conflicts of Interest: The authors declare no conflict of interest.

REFERENCES

Af'idah, D. I., Dairoh, Handayani, S. F., & Pratiwi, R. W. (2021). Pengaruh Parameter Word2Vec terhadap Performa Deep Learning pada Klasifikasi Sentimen. *Jurnal Informatika: Jurnal Pengembangan IT (JPIT)*, *6*(3). https://doi.org/https://doi.org/10.30591/jpit.v6i3.3016

Ahmed, M., Ansari, M. D., Singh, N., Gunjan, V. K., Krishna, B. S., & Khan, M. (2022). Rating-Based Recommender System Based on Textual Reviews Using IoT Smart Devices. *Mobile Information Systems*, *2022*, 1–18. https://doi.org/10.1155/2022/2854741

Alamdari, P. M., Navimipour, N. J., Hosseinzadeh, M., Safaei, A. A., & Darwesh, A. (2020). A Systematic Study on the Recommender Systems in the E-Commerce. *IEEE Access*, *8*, 115694–115716. https://doi.org/10.1109/ACCESS.2020.3002803

Alfaifi, Y. H. (2024). Recommender Systems Applications: Data Sources, Features, and Challenges. *Information*, *15*(10), 660. https://doi.org/10.3390/info15100660

Andrade, C. (2020). Sample Size and Its Importance in Research. *Indian Journal of Psychological Medicine*, *42*(1), 102–103. https://doi.org/10.4103/IJPSYM.IJPSYM_504_19

Badillo, S., Banfai, B., Birzele, F., Davydov, I. I., Hutchinson, L., Kam-Thong, T., Siebourg-Polster, J., Steiert, B., & Zhang, J. D. (2020). An Introduction to Machine Learning. *Clinical Pharmacology & Therapeutics*, *107*(4), 871–885. https://doi.org/10.1002/cpt.1796

Barkan, O., Caciularu, A., Katz, O., & Koenigstein, N. (2020). Attentive Item2Vec: Neural Attentive User Representations. https://doi.org/10.48550

Barkan, O., & Koenigstein, N. (2016). Item2Vec: Neural Item Embedding for Collaborative Filtering. https://doi.org/https://doi.org/10.48550

Bartlett, M., Morreale, F., & Prabhakar, G. (2023). Analysing Privacy Policies and Terms of Use to understand algorithmic recommendations: the case studies of Tinder and Spotify. *Journal of the Royal Society of New Zealand*, *53*(1), 119–132. https://doi.org/10.1080/03036758.2022.2064517

Bhareti, K., Perera, S., Jamal, S., Pallege, M. H., Akash, V., & Wiieweera, S. (2020). A Literature Review of Recommendation Systems. *2020 IEEE International Conference for Innovation in Technology (INOCON)*, 1–7. https://doi.org/10.1109/INOCON50539.2020.9298450

Chen, J., Lian, D., & Zheng, K. (2020). Collaborative Filtering with Ranking-Based Priors on Unknown Ratings. *IEEE Intelligent Systems*, *35*(5), 38–49. https://doi.org/10.1109/MIS.2020.3000012

Cui, Z., Xu, X., Xue, F., Cai, X., Cao, Y., Zhang, W., & Chen, J. (2020). Personalized Recommendation System Based on Collaborative Filtering for IoT Scenarios. *IEEE Transactions on Services Computing*, *13*(4), 685–695. https://doi.org/10.1109/TSC.2020.2964552

Gaiger, K., Barkan, O., Tsipory-Samuel, S., & Koenigstein, N. (2023). Not All Memories Created Equal: Dynamic User Representations for Collaborative Filtering. IEEE Access, 11, 34746–34763. https://doi.org/10.1109/ACCESS.2023.3263931

Geetha, M. P., & Renuka, D. K. (2024). Deep Learning Architecture Towards Consumer Buying Behaviour Prediction Using Multitask Learning Paradigm. *Journal of Intelligent & Fuzzy Systems*, *46*(1), 1341–1357. https://doi.org/10.3233/JIFS-231116

Ginzburg, D., Malkiel, I., Barkan, O., Caciularu, A., & Koenigstein, N. (2021). Self-Supervised Document Similarity Ranking via Contextualized Language Models and Hierarchical Inference. https://doi.org/10.18653/v1/2021.findings-acl.272

Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.

Gupta, M., Thakkar, A., Aashish, Gupta, V., & Rathore, D. P. S. (2020). Movie Recommender System Using Collaborative Filtering. *2020 International Conference on Electronics and Sustainable Communication Systems (ICESC)*, 415–420. https://doi.org/10.1109/ICESC48915.2020.9155879

Gupta, S., & Kant, V. (2020). An Aggregation Approach to Multi-Criteria Recommender System Using Genetic Programming. *Evolving Systems*, *11*(1), 29–44. https://doi.org/10.1007/s12530-019-09296-3

Hameed, M., & Naumann, F. (2020). Data Preparation. *ACM SIGMOD Record*, *49*(3), 18–29. https://doi.org/10.1145/3444831.3444835

He, X., Chen, T., Kan, M. Y., & Chen, X. (2015). TriRank. *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, 1661–1670. https://doi.org/10.1145/2806416.2806504

Hou, R., Kong, Y. Q., Cai, B., & Liu, H. (2020). Unstructured Big Data Analysis Algorithm and Simulation of Internet of Things Based on Machine Learning. *Neural Computing and Applications*, *32*(10), 5399–5407. https://doi.org/10.1007/s00521-019-04682-z

Injadat, M., Moubayed, A., Nassif, A. B., & Shami, A. (2021). Machine Learning Towards Intelligent Systems: Applications, Challenges, and Opportunities. https://doi.org/10.1007/s10462-020-09948-w

Jannach, D., Pu, P., Ricci, F., & Zanker, M. (2022). Recommender systems: Trends and frontiers. In *AI Magazine* (Vol. 43, Issue 2, pp. 145–150). John Wiley and Sons Inc. https://doi.org/10.1002/aaai.12050

Lee, H. I., Choi, I. Y., Moon, H. S., & Kim, J. K. (2020). A Multi-Period Product Recommender System in Online Food Market Based on Recurrent Neural Networks. *Sustainability (Switzerland)*, *12*(3). https://doi.org/10.3390/su12030969

Li, Y., Wang, R., Nan, G., Li, D., & Li, M. (2021). A Personalized Paper Recommendation Method Considering Diverse User Preferences. *Decision Support Systems*, *146*. https://doi.org/10.1016/j.dss.2021.113546

Patoulia, A. A., Kiourtis, A., Mavrogiorgou, A., & Kyriazis, D. (2022). A Comparative Study of Collaborative Filtering in Product Recommendation. *Emerging Science Journal*, *7*(1), 1–15. https://doi.org/10.28991/ESJ-2023-07-01-01

Prabowo, A., Ula, S. M., Syalsabila, A., Prabowo, F. S., Jati, D. K., & Mulana, L. (2021). *Indonesia Tourism Destination.* https://www.kaggle.com/datasets/aprabowo/indonesia-tourism-destination/data

Putri, A. K., & Suliadi. (2023). Rekomendasi Destinasi Wisata di Indonesia Menggunakan Metode Item2Vec. *Jurnal Riset Statistika*, 11–18. https://doi.org/10.29313/jrs.v3i1.1770

Sánchez-Corcuera, R., Casado-Mansilla, D., Borges, C. E., & López-de-Ipiña, D. (2024). Persuasion-Based Recommender System Ensambling Matrix Factorisation and Active Learning Models. *Personal and Ubiquitous Computing*, *28*(1), 247–257. https://doi.org/10.1007/s00779-020-01382-7

Shan, Z. P., Lei, Y. Q., Zhang, D. F., & Zhou, J. (2019). NASM: Nonlinearly Attentive Similarity Model for Recommendation System via Locally Attentive Embedding. *IEEE Access*, *7*, 70689–70700. https://doi.org/10.1109/ACCESS.2019.2916938

Sharma, S., Rana, V., & Malhotra, M. (2022). Automatic Recommendation System Based on Hybrid Filtering Algorithm. *Education and Information Technologies*, *27*(2), 1523–1538. https://doi.org/10.1007/s10639-021-10643-8

Stalidis, G., Karaveli, I., Diamantaras, K., Delianidi, M., Christantonis, K., Tektonidis, D., Katsalis, A., & Salampasis, M. (2023). Recommendation Systems for E-Shopping: Review of Techniques for Retail and Sustainable Marketing. In *Sustainability (Switzerland)* (Vol. 15, Issue 23). Multidisciplinary Digital Publishing Institute (MDPI). https://doi.org/10.3390/su152316151

Tahmasbi, H., Jalali, M., & Shakeri, H. (2021). Modeling User Preference Dynamics with Coupled Tensor Factorization for Social Media recommendation. *Journal of Ambient Intelligence and Humanized Computing*, *12*(10), 9693–9712. https://doi.org/10.1007/s12652-020-02714-4

Tareq, S. U., Noor, M. H., & Bepery, C. (2020). Framework of Dynamic Recommendation System for E-shopping. *International Journal of Information Technology*, *12*(1), 135–140. https://doi.org/10.1007/s41870-019-00388-6

Wadi, H. (2021). *Implementasi Jaringan Syaraf Tiruan Backpropagation Menggunakan MATLAB GUI* (3rd ed.).

Wang, L., Mistry, S., Hasan, A. A., Hassan, A. O., Islam, Y., & Junior Osei, F. A. (2023). Implementation of a Collaborative Recommendation System Based on Multi-Clustering. *Mathematics*, *11*(6). https://doi.org/10.3390/math11061346

Article history: Received: 11.02.2025 Revised: 15.05.2025 Accepted: 06.06.2025 Available online: 30.06.2025