Design a Recommendation System in Real Estate Investment based on Context Approach

Tinh T. Nguyen[1,2], Sang Vu[1,2,*], Truc Nguyen[1,2], Vuong T. Pham[2,3,4] and Hien D. Nguyen[1,2,a]

[1]University of Information Technology, Ho Chi Minh City, Vietnam
[2]Vietnam National University, Ho Chi Minh City, Vietnam
[3]University of Science, Ho Chi Minh City, Vietnam
[4]Institute of Data Science and Artificial Intelligence, Sai Gon University, Ho Chi Minh City, Vietnam

Keywords: Recommendation System, Consultation, Many-Cold-Start-Users, Nearest-Neighbors Approach, Intelligent System.

Abstract: The real estate investment industry has experienced a significant increase in user participation over the years, with individuals keen on registering concurrent interests in both recent and prior projects. This growing trend necessitates the development of an approach that can recommend real estate items in a simultaneous manner. However, the presence of unrequired memberships and stop-by behaviors has introduced several challenges, resulting in numerous cold-start scenarios for new users. This study proposes a recommendation system tailored specifically for real estate, designed to offer warm-start item recommendations of cold-start users using a content-based approach and a session-based recommendation system. Herein, a real estate recommendation system designed to offer satisfactory initial recommendations for items, even in scenarios where many users are encountering the cold-start problem. The session-based recommendation system is adapted and made use of pre-existing methods to effectively handle sequential and contextual data for the encoded attribute prediction of the next-interacted item. Then, the nearest-neighbors method is employed weighted cosine similarity to identify conforming candidates. The results demonstrate the effectiveness of efficiently integrating the information and the difficulty in performing well in item recommendations simultaneously.

1 INTRODUCTION

Among the crucial challenges in any e-commerce is fmaintaining the existing users while attracting new ones and recommendation systems play a crucial role in enhancing the user experience and improving the overall functionality of a real estate investment platform. The common recommendation system uses historical records as prior knowledge to choose candidates and performs most effectively with adequate records (Roy and Duta, 2022, Nguyen et al, 2020). However, the recommendation task becomes complex due to no clear data of new users which inevitably leads to more problems for the recommendation/consultation system. Therefore, addressing this cold-start problem is essential for enhancing the overall functionality and usability of real estate investment platforms.

Real estate recommendation systems further complicate this by requiring consideration of both recent and prior projects, reflecting user interest concurrency. Additionally, specific attributes like location, developer brand, and living space significantly influence user search and purchase behaviors (Chia et al., 2016). Consequently, an effective recommendation approach requires the ability to simultaneously incorporate these attributes into real estate item representations, addressing the item cold-start problem within this specific context.

Item cold-start presents a significant challenge for recommender systems, particularly in domains with frequent additions like real estate. This issue arises when insufficient user interaction data exists

[a] https://orcid.org/0000-0002-8527-0602
[*] Corresponding author.

Nguyen, T., Vu, S., Nguyen, T., Pham, V. and Nguyen, H.
Design a Recommendation System in Real Estate Investment Based on Context Approach.
DOI: 10.5220/0012210800003598
In *Proceedings of the 15th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2023) - Volume 2: KEOD*, pages 255-263
ISBN: 978-989-758-671-2; ISSN: 2184-3228

for new items, hindering accurate recommendations (Wei et al., 2017). These cold-start items remain invisible to users due to record deficiencies or listing omissions. The integration of context information is demonstrably beneficial for enhancing recommendation system performance (Adomavicius & Tuzhilin, 2011). This is particularly true in the realm of real estate recommendations, where user preferences are demonstrably influenced by surrounding contextual factors. Using context information is an approach to achieve better real estate recommendations. The context information is also represented by ontology (Bouihi and Bahaj, 2018, Nguyen et al., 2021) and graph information (Yan et al., 2019, Nguyen et al., 2023).

This study presents a real estate recommendation methodology devised to address the item cold-start predicament, particularly within scenarios involving cold-start users. It entails the adaptation of a session-based recommendation framework, leveraging preexisting mechanisms to effectively manage sequential and contextual data pertinent to the prediction of encoded attributes for the subsequent interacted item. Additionally, a nearest-neighbors approach is incorporated to ascertain suitable candidate items. Empirical findings demonstrate that the proposed approach not only compares favorably with various applied mechanisms but also outperforms baseline methodologies utilizing the top-n recommendation paradigm, as evidenced by experimentation conducted on a real estate dataset.

2 PREMILARIES

2.1 The Recommendation System with Attention Mechanism

The recommendation system functions by analyzing the click sequence, denoted as $[c_1, c_2, \ldots, c_k]$, and forecasting the subsequent click c_{k+1}, where c_j is the j^{th} event of the session. Following this analysis, the system generates scores for each item, ultimately recommending only the top-n highest-scored items to the user. Leveraging this framework, the profile learner predicts the user profile based on these sequential patterns.

In a recommendation system, a click sequence serves as implicit feedback, derived from user behavior. Such feedback is indirect, necessitating thoughtful handling owing to its highly noisy nature and the absence of negative feedback (Hu et al., 2008). Determining whether users like or dislike a clicked item, or whether a click is a miss click, proves impossible. To address this challenge, the profile learner employs an attention mechanism to manage noise and discern the sequence's intent, assigning varying levels of importance to individual clicks.

Li et al. (2017) introduced the encoder segment of the Neural Attentive Recommendation Machine (NARM), a session-based recommendation system featuring an attention mechanism. The encoder component of NARM comprises two encoders: the global encoder and the local encoder. The global encoder encapsulates the entirety of user behavior within the click sequence, culminating in the final hidden state of the Recurrent Neural Network (RNN). This final hidden state serves as a comprehensive representation of the user's interaction history.

$$e_l = \sum_{j=1}^{n} w_j . s_j \tag{1}$$

$$e_g = s_q \tag{2}$$

In which:

- e_l and e_g are the output of the local and golobal encoders (resp.), and
- s_j is the hidden state of RNN at time step j, and s_q is the last hidden state of RNN, and
- w_j is the weighted factor, they are defined as the sum of weighted hidden states from every time step as follows (3):

$$w_j = \frac{\exp(score(s_q, s_j))}{\sum_{i=1}^{n} \exp(score(s_q, s_i))} \tag{3}$$

$$score(s_q, s_j) = C.\sigma\left(A.s_q + B.s_j\right) \tag{4}$$

where, σ is an activation function, C is a weighting vector, and A and B are the learned weights of s_q and s_j (res.). Consequently, the outputs from the local and global encoders are combined and applied to the layers that follow.

2.2 The Recommendation System based on Context Information

Moreover, recommenders based on content characteristics leverage item specifics, including both structured data like metadata and less structured forms like descriptions, to suggest relevant items to users (Lops et al., 2011). The profile learner aims to anticipate the user profile by examining attributes of interacted items, which are structured similarly to item profiles. Following this, the filtering component utilizes a matching algorithm to identify relevant items solely based on their attributes.

Contextual information proves beneficial in enhancing the recommendation task (Adomavicius and Tuzhilin, 2011). To address challenges related to increasing input dimensions, particularly when it involves additional hidden units in the model, the profile learner adopts the Localized Context technique in (Beutel et al., 2018). This method enables efficient integration of contextual features, thereby enhancing the effectiveness of the recommendation system. It operates by multiplying all embedded context features in hidden states by their elements as follows:

$$s_j := \left(1+\sum_{i=1}^{n_j}\alpha_i\right)*s_j \tag{5}$$

where, s_j is the hidden state of RNN at time step j ($1 \leq j \leq n$), and α_i is the embedded context feature ($1 \leq i \leq n_j$, with n_j is the number of features at state s_j).

The embedding layer of each context feature is initialized by a 0-mean Gaussian distribution to ensure that the multiplicative term has a mean of 1 (Polohakul et al., 2021). This initialization causes the multiplicative term to act like an attention mechanism in the hidden state. The element-wise product is performed both before and after passing through the RNN. These multiplications are considered as prefusion and post fusion, consecutively.

3 CONTENT-BASED APPROACH FOR RECOMMENDATION SYSTEM IN REAL ESTATE

Using the content-based approach having a profile learner and a filtering component, the item cold-start problem has been solved. This recommendation system in real estate stands out with its use of an attention mechanism within a session-based framework. This allows it to prioritize specific details within each user's browsing history, leading to more accurate profile creation and relevant item suggestions based on nearest-neighbor comparisons.

3.1 Precdiction of User Profile

The encoded qualities of the next item that is interacted with make up the prediction of a user profile, also known as a profile learner. It makes use of the context data and click sequence.

This system analyzes a user's actions, represented as a sequence of clicks [c_1, c_2, ..., c_n] with additional context [$cont_1$, $cont_2$, ..., $cont_n$]. The profile learner then makes educated guesses about what the user might do next [f_1, f_2, ..., f_m]. These predictions are like compact descriptions of future actions, expressed either as one-hot or binary encoding depending on the possible number of classes. Each f_j prediction is either a multiclass or binary classification depending on ground truth encoding.

For instance, in real estate ventures, the count of bedrooms serves as a feature. This is exemplified by three potential categories: one, two, or three bedrooms. Consequently, predicting this feature entails a multiclass classification endeavor. Furthermore, it evolves into a multilabel classification task when consolidating these forecasts into feature predictions. Specifically, we forecast all potential categories for each feature of the subsequent interaction, utilizing them to shape a user profile.

The architecture of the session-based recommendation system has been adjusted to align more closely with its objectives. Additionally, the attention mechanism from the encoder part of NARM has been adopted with the LC in the profile learner to efficiently deal with sequential and context information. Consequently, this enhancement is expected to yield improved user profile prediction outcomes, facilitating both warm-start and cold-start item recommendations. In the profile learner, we input an embedded item identifier, numerical features, embedded categorical features, and embedded context features of a click sequence to predict the user profile (see Figure 1).

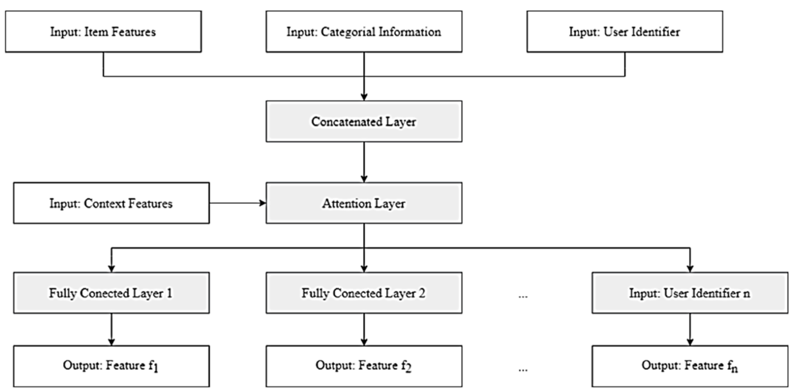

Figure 1: The structure of the proposeed profile leaner.

In the attention layer, both global and local encoders are utilized to the structure found in the encoder section of NARM. However, instead of relying on the original hidden states, we employ postfusion products. This substitution for the influence of context information when determining the attention score. The output of the attention layer is a combined vector derived from both the local and global encoders, which is then passed to the fully connected layers for computing class scores across all features. Each fully connected layer is tasked with predicting one encoded feature, thus its number of units corresponds to the number of classes for that feature. For multiclass classification, the activation function typically employed is either softmax or sigmoid, while for binary classification.

3.2 Filtering Components

The key role of the filtering component is to identify items aligned with a user's predicted profile. The nearest-neighbor approach is achieved to employ. This method involves scoring all items using weighted cosine similarity**,** which takes into account the numerical values of all features in the user profile. Notably, this metric is well-suited for high-dimensional vectors, such as those representing user and item profiles in our system.

For similarity computations, user and item profiles need to be represented in the same format. Therefore, a concatenated vector is utilized to encompass all encoded features. Because multiclass and multilabel encoded features can impact the similarity score differently due to varying value sums within the vector, the weighted cosine similarity is used by its flexibility in assigning specific weights to each component. The weighted cosine similarity between two vectors a = (a_1, …, a_n) and b = (b_1,…b_n) is defined as follows (Xia et al., 2015):

$$Similarity = \frac{\sum_{i=1}^{n} m_i.a_i.b_i}{\sqrt{\sum_{i=1}^{n} m_i.a_i^2} \times \sqrt{\sum_{i=1}^{n} m_i.b_i^2}} \quad (6)$$

where, m_i is the weight for each component of vectors ($1 \le i \le n$). Each weight value m_i is defined as follows:

$$m_i = \begin{cases} 1/n & \text{, if } a_i \text{ and } b_i \text{ are components} \\ & \text{of multilabel features} \\ 1 & \text{, otherwise} \end{cases} \quad (7)$$

where, n is the number of classes of the corresponding multilabel feature. The structure of our filtering component is shown in Figure 3. This module was employed to compute the similarity score between all item profiles and the anticipated user profile, utilizing equations (6) and (7) as the similarity metric. This investigation devises an approach for recommending the top-n items with the highhest similarity score to the user. The structure of the proposed filtering component is shown as Figure 2.

3.3 Real-Estate Pricing System

A real estate pricing system is a valuable tool or model employed to gauge the value or worth of properties within the real estate market. By leveraging an array of factors and data points, it offers valuable insights into property valuation, empowering buyers, sellers, and real estate professionals to make well-informed decisions. A standard real estate pricing system encompasses the following components:

Data Collection: A real estate valuation system gathers an extensive array of information associated with the properties under examination. Moreover, it might incorporate supplementary sources of data, including demographic details, economic metrics, and plans for infrastructure development.

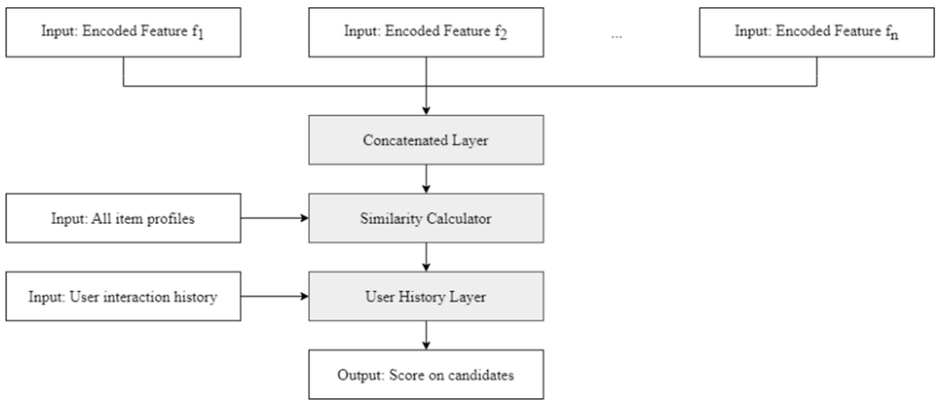

Figure 2: The structure of the proposed filtering component

Feature Selection: After the data collection phase, the pricing system proceeds to pinpoint the most pertinent features that exert a substantial influence on property prices. This endeavor employs statistical analysis and machine learning methodologies to ascertain which variables exhibit the most robust correlation with property values.

Model Development: The pricing system employs a variety of statistical and machine learning models to construct a prediction model. Frequently utilized models encompass multiple linear regression, random forests, or neural networks. The selection of model depends on the complexity of the data and the precision necessary for estimating prices.

3.4 Regression Tree

Regression tree is a machine learning algorithm used for solving regression problems (Loh, 2011). Operating as a variant of decision trees, each internal node within a regression tree assesses a specific feature or attribute, while every leaf node corresponds to a prediction or an estimated value for the target variable.

Constructing a regression tree entails recursively dividing the feature space according to the values of various input features. The objective is to generate homogeneous regions or subsets of the data that exhibit similar characteristics concerning the target variable. This segmentation is achieved by identifying the feature and its corresponding threshold that yields the optimal split, typically evaluated using metrics such as mean squared error (MSE) or mean absolute error (MAE).

3.5 Linear Regression

For the dataset of house prices and their corresponding sizes, a linear regression model is built to predict the price of a house based on its size, number of bedrooms, number of toilets, number of floors as Table 1.

Table 1: Some information of a house.

Order	Price	Area	Bedrooms	Toilets	Floors
1	10	126	2	2	1
2	15	152	2	3	1
3	17	116	1	2	3
4	11	133	3	2	1

The linear relationship between the area, number of bedrooms/toilets/floors of the house (variables x_1, x_2, x_3, x_4) and its price (dependent variable y) was found. It can be represented by the equation:

$$y' = f(x) = w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4 + c \quad (8)$$

where, $w = \{w_1, w_2, w_3, w_4\}$ is the slope (regression coefficient) and c is the intercept (y-intercept).

The linear regression model offers a simplified depiction of the connection between house size and price by assuming a constant slope, implying that the price rises linearly with the house's size. Nevertheless, it is important to recognize that linear regression might not effectively capture intricate nonlinear relationships or interactions among variables.

4 EXPERIMENTAL RESULTS

4.1 Dataset and Data Preparation

The dataset is collected from website batdongsan.com.vn[1], which is a huge website in real estate in Vietnam. The data was comprised in 2020. It included 7,800,552 interactions involving 4,424 users and 5,674 items. Subsequently, these metadata were processed to form the item profile for each respective item, thereby ensuring that each interaction is contextualized with its relevant featThe item profiles derived from their metadata serve as the ground truth for predicting user profiles and are involved in calculating similarity scores. Features with a substantial number of unique values or missing data exceeding half of the dataset are excluded from consideration as they lack effectiveness in interpretation. Moreover, data imputation techniques are applied to address the remaining missing values, utilizing information from other features such as real estate type, location, and price level.

After processing, numerical values are transformed into discrete classes based on their feature characteristics. An upper bound is defined for each class, grouping values together beyond that point. This strategy simplifies analysis but introduces a trade-off by merging some potential variations into larger categories.

Following the classification process, the assigned class for one possible class is encoded using either one-hot or binary encoding, while many possible classes are represented through binary encoding. Consequently, item profiles comprising 11 encoded features were generated for 5,674 items. The corresponding information is presented in Table 2.

[1] https://batdongsan.com.vn/

Table 2: Description, type, and post-processed information of the encoded attributes within the item profiles.

Name	Type	isRequire	Description
ID	string		Unique item ID
Code	number	True	Display code of item
Title	string	True	Title of item
Type	string	True	Item Type
Project	string		Project ID containing item
Pictures	string	True	Pictures
Detail	object	True	Description of item
Author	object		User ID of author
Contact	Object	True	Display contact info
dateUpload	Date		Updated item date
dateEnd	Date		Scheduled end date

Table 3: Description and participation of context features in prefusion and postfusion.

Feature name	Description
Click	Number of clicks with the item
user_device	Type of user system (Desktop and Mobile)
fetched_province	Province ID requested with View-API
fetched_country	Country ID requested with View-API
user_agent	Information of user's operating system (browser info, Windows, Android,…)
prev_page	Previous fetched page
await_time	Time await since last View-API called

The dataset was split into training and testing sets, each containing sequences of context features. These features provided context for a specific prefusion process, likely related to user interactions. Seven features were used in total (Table 3): number of clicks (*click*), type of users' device (*user_device*), province and country (*fetch_province* and *fetch_country*), information of operation system (*user_agent*), previous fetched page (*prev_page*), and time elapsed since the previous interaction (*await_time*).

4.2 Evaluation

The performance of the system is assessed using the top-n recommendation task. This task focuses on how well the recommended items align with users' actual clicks. Two key metrics are used, Recall@K and Mean Reciprocal Rank@K (MRR@K), where K represents the number of recommended items.

Recall@K is a metric used to evaluate the performance of a recommendation model by measuring how many relevant items from the actual set of items are included in the top-K recommendation list provided by the model.

$$\text{Recall@K} = \frac{n_{hit}}{N} \tag{9}$$

where, n_{hit} is the number of cases having the actual click and N is the number of all cases.

MRR@K is a metric used to evaluate the ranking performance of a recommendation model. It calculates the average reciprocal rank of the first relevant item within the top-K recommendation list provided by the model.

$$\text{MRR@K} = \frac{1}{N}\sum_{c \in Clicks} \frac{1}{rank(c)} \tag{10}$$

where, c is the actual click, *Clicks* is a set of cases having the actual click, N is the number of all cases.

The model performance is evaluated in terms of two aspects: warm-start and cold-start item recommendations. Test cases are assigned to each perspective using the type of their actual click item. New items are defined when they only appear in the testing set and the top 100 most recently introduced items in the training set.

4.3 Comparison with Other Methods

Baselines were classified based on their effectiveness in two key aspects: warm-start and cold-start item recommendations. The content-based approach (Lops et al., 2011) employs various profile learners and similar filtering components to address cold-start item baselines. Table 4 introduces baselines methods and context-approach, which are used to compare in this study:

Table 4: Baseline methods for comparison.

Order	Methods	Meaning
1	S-Pop	A sequential popularity predictor suggests items based on their ranking determined by the frequency of interactions within the current sequence.
2	Item-KNN (Sarwar et al., 2001)	This approach delivers personalized recommendations by identifying items closely related to your last interaction. It analyzes the unique characteristics of the interacted item and suggests similar items based on these characteristics, providing a seamless and relevant user experience.
3	NARM (Li et al., 2017)	A session-based recommendation system utilizing encoder-decoder GRU and featuring an attention mechanism.
4	STAMP (Liu et al, 2018)	A session-based recommendation system utilizing MLP and featuring an attention mechanism. Based on consecutive clicks, a recommendation list is generated.
5	CB (S-Pop) (Lops et al., 2011)	A popularity predictor sequence that forecasts the user profile based on the item profile of the most interacted item within the current sequence.

Table 5 is the results of warm-start item recommendation through 1,532 test cases. Moreover, to identify conforming candidates, the nearest-neighbors method is also combined with weighted cosine similarity.

In terms of Recall@K, the proposed approach is better than the warm-start itenm S-Pop and the context-approach CB (S-Pop). The proposed method struggles to match the Recall@K performance of Item-KNN, NARM, and STAMP. This stems from its exclusive use of item attributes for candidate selection. This approach can effectively recommend items aligned with predicted user profiles built from those attributes, but fails to capture the full spectrum of user interests that might extend beyond predefined categories.

In terms of MRR@K, the proposed approach is at the third place behind NARM and STAMP, consecutively. However, our approach is outperformance than the cold-start item as context-approach, CB (S-Pop), in both terms, Recall@K and MRR@K.

Personalizing recommendations is crucial in real estate, where user priorities vary greatly. The current approach treats all features as equally important, potentially missing the mark for individual needs. By introducing user-specific weights for each feature, we can unlock a personalized user experience, suggesting properties that truly match their unique preferences and significantly improve overall satisfaction.

5 CONCLUSION

In this paper, an approach for building a recommendation system in real-estate is proposed. In the case of numerous cold-start customers, this method resolves the item cold-start problem with respectable warm-start item recommendations. It adapts a session-based recommendation system and makes use of already in place methods to effectively handle sequential and contextual data for the encoded attribute prediction of the next interacted item. The experimental results demonstrate that this method is superior to baselines utilizing the top-*n* recommendation with the dataset from the real estate search engine as well as to other used methods.

Based on the idea, people in the same group should react similarly to similar items, the recommendation approach improves search results by using customer demographic data (Matuszelański and Kopczewska, 2022). In the future, the proposed method will be combine the knowledge base of real-estate investment (Nguyen et al., 2022) for recommending more accuracy based on customers' behaviors, which will be aimed at the demographic profile of customer.

Table 5: Comparison with other methods.

Method	Recall @5	MRR @5	Recall @10	MRR @10	Recall @20	MRR @20	Recall @30	MRR @30
S-Pop	21.44	12.47	20.12	12.52	22.62	12.24	24.12	13.14
Item – KNN	28.34	13.7	36.56	15.12	47.52	18.15	49.01	18.12
NARM	34.71	22.73	**41.33**	20.53	53.24	23.71	**56.23**	22.23
STAMP	**35.56**	**25.12**	40.32	**23.65**	**55.35**	**24.11**	55.21	**23.64**
CB (S-Pop)	17.96	13.45	19.13	10.21	24.82	10.56	24.12	11.09
Proposed approach	27.41	19.43*	34.23	18.51*	40.56	18.55*	43.04	18.67*

Bold means the highest value in the measurement.
The symbol "*" means Top 3 in the measurement between methods

ACKNOWLEDGMENT

This research was supported by The VNUHCM-University of Information Technology's Scientific Research Support Fund.

REFERENCES

Adomavicius, G., Tuzhilin, A. 2011. Context-aware recommender systems. In *Recommender Systems Handbook*. Ricci, F. et al. (Eds). Springer.

Beutel, A., Covington, P., Jain, S., et al. 2018. Latent cross: Making use of context in recurrent recommender systems. In *WSDM 2018, 11th Int. Conf. Web Search Data Mining, 2018*, pp. 46–54. ACM.

Bouihi, B., Bahaj, M. 2018. Ontology and rule-based recommender system for e-learning applications. *International Journal of Emerging Technologies in Learning*, 14(15), 4.

Chia, J., Harun, A., Kassim, A., 2016. Understanding factors that influence house purchase intention among consumers in kota kinabalu: An application of buyer behavior model theory. *J. Technol. Manage. Bus.* 3(2).

Hidasi, B., et al. 2016. Session-based recommendations with recurrent neural networks. *In ICLR 2016, 4th International Conference on Learning Representations,* May 2016.

Hu, Y., Koren, Y., Volinsky, C. 2008. Collaborative filtering for implicit feedback datasets. In *ICDM 2008, 8th IEEE Int. Conf. Data Mining, Dec. 2008*, pp. 263–272. IEEE.

Li, J., Ren, P., Chen, Z., et al. 2017. Neural attentive session-based recommendation. In *Proc. ACM Conf. Inf. Knowl. Manage.*, *Nov. 2017*. ACM

Liu, Q., Zeng, Y., Mokhosi, R., et al. 2018. STAMP: Short-term attention/memory priority model for session-based recommendation. In S*IGKDD 2018, 24th Int. Conf. Knowl. Discovery Data Mining*, *Jul. 2018*, pp. 1831–1839. ACM.

Loh, W.Y. 2011. Classification and regression trees. *Wiley interdisciplinary reviews: data mining and knowledge discovery*, *1*(1), 14-23.

Lops, P., Gemmis, M., Semeraro, G. 2011. Content-based recommender systems: State of the art and trends. In *Recommender Systems Handbook*. Ricci, F. et al. (Eds). Springer.

Matuszelański, K., Kopczewska, K. 2022. Customer Churn in Retail E-Commerce Business: Spatial and Machine Learning Approach. *Journal of Theoretical and Applied Electronic Commerce Research 17*(1), 165-198.

Nguyen, H., Tran, D., Do, H., et al. 2020. Design an intelligent system to automatically tutor the method for solving problems. *International Journal of Integrated Engineering*, *12*(7), 211-223.

Nguyen, H. D., Tran, T. V., Pham, X. T., et al. 2021. Ontology-based integration of knowledge base for building an intelligent searching chatbot. *Sensors & Materials*, 33(9), 3101 – 3123.

Nguyen, H. T., Nguyen, H. D., Do, N. V., et al. 2022. Knowledge Representation of Expert System in Real-Estate Investment Combining Collected Data. In *New Trends in Intelligent Software Methodologies, Tools and Techniques* (pp. 571-583). IOS Press.

Nguyen, H. D., Truong, D., Vu, S., et al. 2023. Knowledge Management for Information Querying System in Education via the Combination of Rela-Ops Model and Knowledge Graph. *Journal of Cases on Information Technology (JCIT)*, 25(1), 1-17.

Polohakul, J., Chuangsuwanich, E., Suchato, A., Punyabukkana, P. 2021. Real estate recommendation approach for solving the item cold-start problem. *IEEE Access*, *9*, 68139-68150.

Roy, D., Dutta, M. 2022. A systematic review and research perspective on recommender systems. *Journal of Big Data*, *9*(1), 59.

Sarwar, B., Karypis, G., Konstan, et al. 2001. Item-based collaborative filtering recommendation algorithms. In *WWW 2001, 10th Int. Conf. World Wide Web,* 2001, pp. 285–295.

Wei, J., He, J., Chen, K. et al., 2017. Collaborative filtering and deep learning based recommendation system for cold start items. *Expert Syst. Appl.* 69, 29–39.

Xia, P., Zhang, L., Li, F. 2015. Learning similarity with cosine similarity ensemble. *Information sciences*, *307*, 39-52.

Yan, Y., Zhang, Q., Ni, B., et al. 2019. Learning context graph for person search. In *CVPR 2019, the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2158-2167. IEEE.

Zhang, S., Yao, L., Sun, A., Tay, Y. 2019. Deep learning based recommender system: A survey and new perspectives. *ACM Comput. Surv*. 52(1), 5