ISBN: 978-981-94-1156-6
2024 the 14th International Workshop on Computer Science and Engineering (WCSE 2024)
doi: 10.18178/wcse.2024.06.014

Predicting Consumer Actions in Digital Banking with Time-Sensitive User Behavior Analysis

Yusuf Subaşı [1], Yıldız Karadayı [1], Ilgın Şafak [1 +] and Mehmet S. Aktaş [2]

[1] Fibabanka R&D Center, Istanbul, Turkey

[2] Yıldız Technical University, Computer Engineering Department, Istanbul, Turkey

Abstract. Digital banking provides customers access to a wide range of banking services with rich graphical user interfaces to conduct banking activities efficiently and effectively. Many digital banking sites analyze end-users' navigational patterns in order to extract information that can be used to increase customer loyalty, and to predict their next action. This can be accomplished by utilizing machine learning techniques. However, data with high dimensions pose computational challenges to machine learning, as well as an increased risk of overfitting, making it difficult for meaningful patterns to be extracted. Embedding techniques can help mitigate these issues by transforming complex, high-dimensional data into a manageable, low-dimensional space, which makes it possible for machine learning algorithms to perform effectively. Embedding methods that utilize graph structure-based embedding approaches are required to capture and model user behavior to provide better predictions. In this study, graph structure-based embedding approaches are proposed as a means of representing user navigational patterns during browsing. A prototype implementation of the proposed embedding approach is provided in order to facilitate the validation of the approach. Experimental results suggest that the proposed approach has the potential to capture the navigational behavior of the user.

Keywords: graph-based embedding approach, struc2vec, user navigational behavior, digital banking, user behavior prediction

1. Introduction

Digital banking has witnessed an unprecedented surge in popularity in recent years, revolutionizing the way individuals and businesses manage their finances through unprecedented convenience, accessibility, and an array of innovative features. Understanding user navigational behavior within digital banking platforms, such as mobile apps and online banking websites, is crucial for financial institutions. Navigational behavior in the context of digital banking includes actions such as browsing through account statements, searching for specific transactions, exploring investment options, or navigating through financial reports. Machine learning (ML) methods could be utilized to find patterns in clickstream data and make predictions or decisions without explicit programming. Data with high dimensions, however, presents computational challenges to ML, as well as an increased risk of overfitting. These issues can be mitigated using embedding techniques, through which complex, high-dimensional data can be transformed into a manageable, low-dimensional space, enabling ML algorithms to perform effectively. Existing embedding methods include word embedding, image embedding and graph embedding [1], [2]. Graph based embedding approaches in the literature include vertex embedding and graph embedding [1]. Vertex embedding methods include Deepwalk [3], node2vec [4], and Structural Deep Network Embedding [5]. Graph2vec [6] and struct2vec [7] are examples of graph embedding methods. There exist several other embedding approaches for representing graphs in the literature [8-9]. Some of these embedding approaches are as follows: IsoMap [10], Laplacian Eigenmaps [11], GraRep [12], Hope [13], Graph Factorization [14], and Structure Preserving [15]. This study differs from these approaches in that it examines the role of time in successful graph representation. To this end, aging-based graph representation methods are explored, where the graph embedding approach, struc2vec, is extended. Additionally, within the scope of literature, we note the application of word2vec and graph-based embedding methods for modeling data in

+ Corresponding author. Tel.: 90 (212) 381 3003; fax: 90 (212) 381 8576.
E-mail address: ilgin.safak@fibabanka.com.tr.

various domains [16-21]. In the context of this research, we employ embedding approaches for modeling user behaviors in the banking domain.

Existing literature suggests that embedding methods that use graph data to define user navigation behaviors are effective. Current studies affirm the effectiveness of embedding techniques in capturing user navigation behaviors, particularly in the context of digital banking [8], [9], [1], [2]. These methodologies, however, exhibit a critical limitation: they ignore the temporal dynamics and frequency of user interactions, which are imperative for a comprehensive understanding of the user experience. This study addresses several key issues related to modeling and predicting user behavior in digital banking using graph-based identification methods. Specifically, the following questions were explored:

1. How can a business process be developed and used to predict future user behavior based on digital banking interaction data? How can embedding methods be utilized to implement such a business process?
2. How will prioritizing frequently visited nodes within user navigation browsing data help develop more accurate models of overall user navigational behavior?
3. How will prioritizing recent session data within user navigation browsing data help develop more accurate models of overall user navigational behavior?
4. How can the performance of a business process utilizing graph-based data definition methods be evaluated to predict user behavior by modeling click- stream data from digital banking data sets?

In this paper, we introduce a novel time-sensitive embedding methodology specifically designed for interpreting digital banking users' browsing behaviors. By focusing on recent activities, we capture the most relevant insights for current financial communications, aiding institutions in narrating a story that resonates with market realities. Our approach not only represents user activities more accurately but also predicts their subsequent actions, allowing for preemptive decision-making by financial institutions.

The contributions of this paper include the following:

- The graph embedding approach, struc2vec method, is extended by incorporating the time and frequency of web pages visits in order to model customers' behaviors using ML algorithms. Real digital banking data related to digital banking service usage statuses, including foreign exchange (FX) market usage, was utilized.
- The embedded data is used in predicting bank customers' usage of FX products with the XGBoost, Isolation Forest and Random Forest ML algorithms.
- A new business process that accurately defines and models user behavior in digital banking for the purpose of predicting future user behavior based on transactions and interactions with digital banking is developed and validated.

This paper is organized as follows. In Section 2, the proposed methodology is described in detail. Section 3 provides information on the prototype implementation and the experimental study. Finally, Section 4 summarizes the key findings of the study, draws conclusions, and presents recommendations for future research.

2. Graph-Based Embedding Approaches to Modeling Click-stream Data for Predicting Customer Behavior

This section presents a business workflow designed to model click-stream data using several graph-based embedding approaches for prediction purposes. The proposed business workflow predicts whether users will access an FX service. Fig.1 illustrates how we apply data pre-processing phases to click-stream data using different embedding techniques. The proposed business workflow modules are described. Fig. 2 illustrates how we leverage pre-processed ML-ready data sets to enable prediction tasks. The discussion of the proposed workflow addresses Research Question 1, outlined in the Introduction.

- Description of the Attributes of the Data Set: The proposed business workflow takes a data set from click-stream data collected from digital banking sites. A description of the fields of the dataset and system components are provided below.

- Customer Number: Contains an identifier for each user. Ensures that customer data stays separate and tracked, protecting user privacy.
- Sessions: Each time a customer uses the online banking system, a separate session is recorded, with a unique session number assigned to each session.
- Session Page Sequence: Describes the order in which buyers view pages during each session.
- Data Pre-processing Module: Responsible for applying pre-processing techniques on raw data to improve data quality and clean data, such as the removal of outliers and imputation of missing values. In terms of the number of page visits, the data set includes the most visited 'transition and system pages', such as 'LOGIN', 'HOMEPAGE', and 'LOGOUT'.
- Group by Customer Module: This module groups together customer session data per customer and calculates page duration between page visits. Tables 1 and 2 illustrate an example output of this module.
- Embedding Module: Generates a vector presentation of each page based on customer number-page tuples using the embedding algorithm, struc2vec.
- Weight Calculation Module: Different weights are assigned based on customer navigational behaviors. This module takes an input data format il- lustrated in Table 2 (Data State-2), and calculates the weights corresponding to each Page Visit. Various weight calculation methods based on customer navigational data are utilized, such as the total number of visits and the duration of each page.
- Customer-Based Feature Vector Construction Module: Generates embedding data sets used as inputs for ML algorithms. This module takes three types of input data; the first is Data State-1, as illustrated in Table 1; the second is page visit embedding vectors; and the third is weight data based on the weight calculation method described above. Produces a customer-based embedding vector associated with a label from session- specific vectors. The customer embedding vectors are determined using two approaches. In the first approach, all sessions are assumed to be equally important, so an average is taken across all sessions of a particular user. In the second approach, a weighted average approach is used to determine the customer embedding vector.
- Data Splitter Module: Divides customer-based vector and label column combined data into two data sets: the training data set and the test data set (see Fig. 2). ML models are constructed/trained using training data, and the model is evaluated using test data.
- Model Construction Module: Isolation Forest, Random Forest, and XG-Boost ML algorithms are used for predicting customer behavior.
- Prediction/Success Evaluation Module: Test data created in the "Data Splitter Module" is used as output accuracy, recall, precision, and F1 score metrics.

Table 1: Raw data format example

(Data state-1)

Customer No.	Session No.	Page Sequences	Page Duration
1	2	1, 5, 2	5, 10, 13
1	3	2, 5, 3	2, 4, 7

Table 2: Aggregated data format example

(Data state-2)

Customer No.	Page Sequences	Page Duration
1	1, 5, 2, 3	5, 14, 15, 7

During the pre-processing stage of this module, we implemented a set of stringent rules. This ensured consistency, accuracy, and a meaningful interpretation of the session data. These rules address common discrepancies and gaps in the data, establishing a uniform standard for subsequent analysis stages:

- Alignment of Page Sequences and Duration: To maintain data integrity, a check is performed to ensure that the page sequence length aligns with the page duration sequence length. In instances where a

mismatch occurs, “null” values are systematically added to reconcile the length difference. This process ensures that each page visit has a corresponding duration value, thereby preserving the continuity and completeness of the data records.

- Adjustment of Zero-Valued Duration: Sessions occasionally record a ’zero’ page duration, which can skew analysis and interpretations. To mitigate this, any duration logged as ’0’ is recalculated to reflect the mean duration of the entire session. This adjustment provides a more accurate representation of the time spent during the session, eliminating bias introduced by zero values.
- Handling of Null Values: “Null” entries challenge data analysis. To address this, any “null” value identified within the data set is replaced with the session’s mean value. This approach contributes to a more robust data set by minimizing the impact of missing or incomplete records.
- Sequence Length Consideration: Following post-processing of the data set, only records with sequence lengths of at least three are included. By applying this rule, we ensure that the data set accurately reflects the navigation patterns of users in a manner that provides meaningful insights for analysis. This eliminates instances where limited data might compromise the accuracy of behavioral predictions.

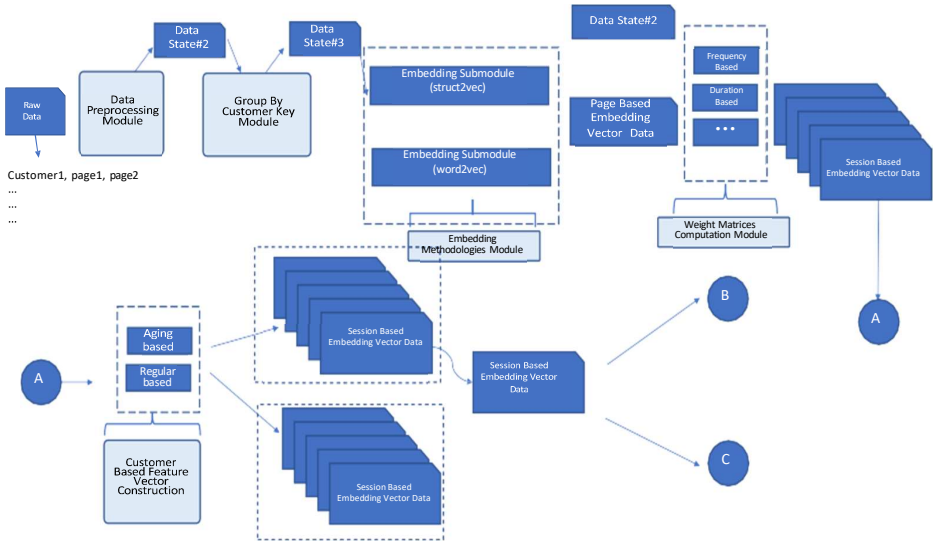

Fig. 1: Proposed e-business workflow for data-pre-processing steps to prepare ML-ready data set

We utilize four different weight calculation methods, explained below.

1. Customer Frequency Based Weight Calculation Method: Page visit counts are calculated using all-session data. The data is then normalized by dividing the specific page visit count by the number of total visits:

$$\text{CFPW} = \frac{\text{CP}}{\text{CAP}} \tag{1}$$

where CFPW is the Customer Frequency Page Weight, CP is the Customer-Specific Page Visit Count and CAP is the Customer All Page Visits. By giving priority to user behavior that has occurred frequently in the past, user behavior may be modeled more accurately.

2. Customer Inverse Frequency Based Weight Calculation Method: For punishing the visit of a frequently used page, the inverse of frequency-based weight is used:

$$\text{CPIFW} = \frac{1}{\text{CFPW}\sum(1/\text{CFPW})} \tag{2}$$

where CPIFW is the Customer Inverse Frequency Page Weight and CFPW is the Customer Frequency Page Weight.

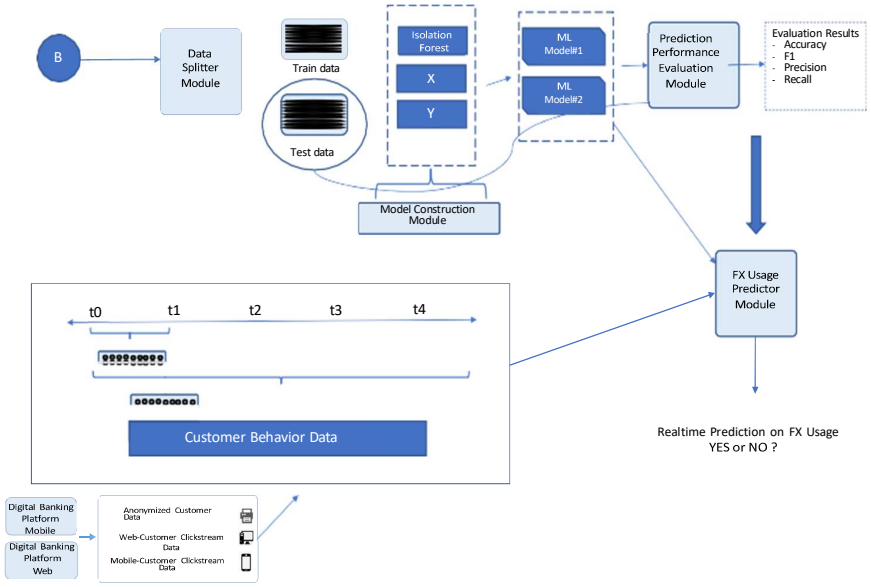

Fig. 2: Prediction of customer behavior using ML models with embedded data as input

3. Customer Page Duration Based Weight Calculation Method: For punishing the visit of a frequently used page, the inverse of frequency-based weight is used:

$$\text{CPDW} = \frac{\text{CPD}}{\text{CAPD}} \tag{3}$$

where CPDW is the Customer Page Duration Based Weight, CPD is the Customer Specific Page Visit Duration Total, and CAPD is the Customer All Page Visits Durations Total.

4. Session Page Duration Based Weight Calculation Method: In this method, weights are calculated through related (single) session duration:

$$\text{SPDW} = \frac{\text{SPD}}{\text{SD}} \tag{4}$$

where SPDW is the Session Page Duration Based Weight, SPD is the Session Page Visit Duration Total, and SD is the Session Duration Total.

Aging-Based Customer Embedding Vector Calculation Method: Gives priority to user behavior data (session data) that occurred more recently, where the most recent n sessions are weighted by c. The Customer Embedding Vector is calculated using the formula below.

$$\text{CEV}(i) = \frac{\sum_k \alpha_k \text{Session Data}(k)}{S_T(i)} \tag{5}$$

where CEV(i) is the Customer Embedding Vector of customer i, α_k is the weight of the k^{th} Session Data and $S_T(i)$ is total number of sessions that belong to customer i.

The weighted sum of all session data is determined by

$$\sum_k \alpha_k \text{Session Data}(k) = c(S_{t-n+1}, S_{t-n+2}, \cdots, S_t) + (1-c)(S_{t-r}, S_{t-r+1}, \cdots, S_{t-1}) \tag{6}$$

where S_t is the t^{th} session data, n is the number of sessions in recent session data, c is the most recent session aging factor and takes values between ($0 < c < 1$), and r is the total number of customer sessions - n. The value n is chosen based on an empirical study for modeling customer navigational behavior. The session aging factor, c, represents the weight of the recent session data, where a value close to 0 indicates that past behavior is valued more than recent behavior. If the value is close to 1, this suggests that the customer's recent behavior is more important than the customer's past behavior.

3. Prototype Implementation and Evaluation

In this section, the details of the prototype implementation and evaluation of the proposed business workflow, as well as the data set used in this study are described.

- Input Dataset: Real-life digital banking customer behavior data is used in this work. The data set contains approximately 3M sessions with 35K unique customers. The FX product's usage is used as a label column. If a customer has a transaction in the FX platform, the customer would be labeled as 1; otherwise, it would be labeled as 0. For 35K customers, the 1:0 ratio is approximately %15 to %85.
- Prototype implementation: In order to implement the proposed work- flow, several libraries are used. The ML algorithms used are XGBoost algorithm version 1.0.2, Isolation Forest and Random Forest algorithms from the ensemble package of Scikit-learn version 0.24.2.
- Evaluation strategy: Experimental studies are conducted to investigate the usefulness of different embedding approached, where different embedding approaches are utilized to create ML-ready data sets. For each ML-ready data set, a variety of ML algorithms are applied to explore prediction success. The train-test split method in scikit-learn is used with a 30-70% split ratio. The accuracy and F1 score of each prediction model are calculated.

The results indicate that embedding approaches that prioritize customers' frequent behaviors provide reliable prediction results. Similarly, reliable prediction results were observed for embedding approaches that prioritize customer visits duration during the session. Furthermore, the results indicate that prioritizing the most recent customer's behaviors, for a predefined most recent session number, when calculating the overall customer behaviors embedding vector, does not lead to significant positive prediction results. These results provide answers to the aforementioned research questions outlined in the Introduction.

Table 3: Embedding methodologies used in the proposed workflow and their explanations

Weight Calculation Method	**Description**
Method 1: Customer Duration Weighted Matrix	Customer-based embedding vector is calculated based on the page duration weights matrix in which weights are calculated through the customer-based page duration.
Method 2: No Weight Matrix	Calculation without the use of weights, this calculation relies solely on page vectors and the sequence of session pages. After summarizing the session vector based on pages visited, it is normalized by dividing the total page visits in the session.
Method 3: Customer Frequency Weighted Matrix	Customer-based embedding vector is calculated based on frequency-based weights matrix. Customer-based page frequencies are calculated using a customer's all-session data.
Method 4: Customer Frequency Inverse Weighted Matrix	Customer-based embedding vector is calculated based on inverse frequency-based weights matrix. It is calculated by taking the inverse of the frequency-based weight to punish frequent visits to a page.
Method 5: Session Duration Weighted Matrix	Customer-based embedding vector is calculated based on the page duration weights matrix in which weights are calculated through related (single) session duration.
Method 6: struc2vec Direct	Plain graph embedding without any weight matrix. Struc2vec algorithm gives vectors for each node (page). Since customer numbers and pages are given as input, customer embedding vectors are also created.

Table 3 provides a description of the customer embedding vector calculation methods used. The F1 scores (left) and accuracy (right) of the three ML models used are compared in Fig. 3 for the methods described in Table 3 using the proposed aged calculation. It is observed that Method 1, Customer Duration Weighted Matrix, provides the highest F1 score and accuracy performance for all ML models.

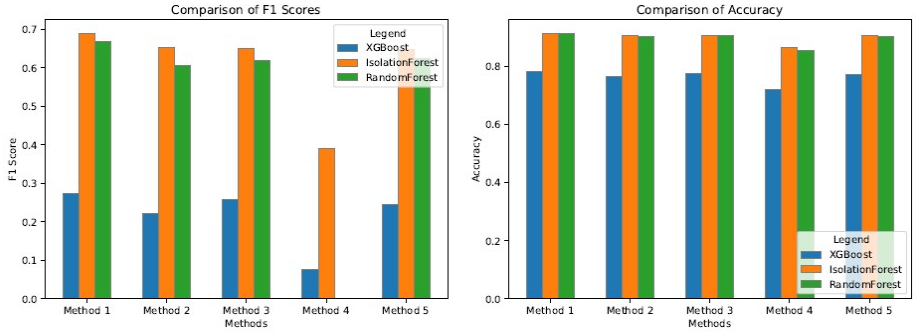

Fig. 3: Comparison of the F1 scores and accuracy of ML models for each method using the aging calculation

Experimental results indicate that the proposed business workflow for modeling digital banking customer behavior is more effective at predicting customers' next FX product transaction patterns. Specifically, for the isolation forest model, frequency-based weighting has improved performance more than other approaches. For the random forest model, customer duration-based and frequency-based weighting approaches work better than aging methods. In the XGBoost-based prediction model, the embedding representation of customers calculated using a session duration-based weighted matrix perform better on all metrics. Overall, the proposed customer behavioral embedding approaches, either duration-based or frequency-based, produce higher prediction results than the struct2vec embedding technique alone.

4. Conclusions and Future Work

This study proposes a business workflow that can process and analyze click- stream data obtained from digital banking sites. As a result of the proposed workflow, the system is able to model the browsing patterns of users in order to predict the next behavior, for example, the likelihood of customers utilizing a particular banking service. Various graph-based embedding approaches were employed to represent click-stream data sets. The representative graph- based embedding approach, struc2vec is extended to include time and frequency of web page views. Graph-based embedding methodologies are examined for representing customer navigation behavior changing with time. The paper also examines whether frequency-based graph embedding approaches provide better data modeling for understanding customer behavior. A prototype implementation and implementation details are provided in order to facilitate the testing of the proposed business workflow. An experimental study was conducted to evaluate the success of embedding approaches. Various embedding approaches were used during the data pre-processing in order to evaluate the prediction success of the proposed business workflow. The results indicate that frequency-based graph embedding improves the modeling and prediction of data. Future work includes comparing the effects of different embedding techniques, and incorporating supervised learning algorithms into the proposed business workflow in order to improve prediction accuracy.

5. Acknowledgement

This study was made possible by the provision of digital banking data sets and computational facilities provided by Fibabanka.

6. References

[1] Palash Goyal and Emilio Ferrara. "Graph embedding techniques, applications, and performance: A survey". In: *Knowledge-Based Systems* 151 (2018), pp. 78–94.

[2] H. Cai, V. W. Zheng, and K. Chang. "A comprehensive survey of graph embedding: problems techniques and application". In: IEEE Transactions on Knowledge and Data Engineering (2018).

[3] B. Perozzi, R. Al-Rfou, and S. Skiena. "DeepWalk: Online Learning of Social Representations". In: KDD'14: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Association for Computing Machinery, 2014, pp. 701–710.

[4] A. Grover and J. Leskovec. "node2vec: Scalable Feature Learning for Net- works". In: KDD'16: Proceedings of the

22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Association for Computing Machinery, 2016, pp. 701–710.

[5] D. Wang, P. Cui, and W. Zhu. “Structural Deep Network Embedding”. In: KDD’16: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Association for Computing Machinery, 2016, pp. 1225–1234.

[6] Annamalai Narayanan et al. graph2vec: Learning Distributed Representations of Graphs. 2017. arXiv: 1707.05005 [cs.AI].

[7] Daniel R. Figueiredo Leonardo F.R. Ribeiro Pedro H.P. Savarese. “struc2vec: Learning Node Representations from Structural Identity”. In: KDD’17: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Association for Computing Machinery, 2017, pp. 385–394.

[8] N. Taşgetiren and M.S. Aktaş. “Mining Web User Behavior: A Systematic Mapping Study”. In: International Conference on Computational Science and Its Applications. 2022, pp. 667–683.

[9] P. Cui et al. “A survey on network embedding”. In: IEEE Transactions on Knowledge and Data Engineering (2018).

[10] J. B. Tenenbaum, V. de Silva, and J. C. Langford, “A global geometric framework for nonlinear dimensionality reduction,” Science, vol. 290, no. 5500, pp. 2319–2323, 2000.

[11] M. Belkin and P. Niyogi, “Laplacian eigenmaps for dimensionality reduction and data representation,” Neural Computation, vol. 15, no. 6, p. 1373–1396, 2003.

[12] S. Cao, W. Lu, and Q. Xu, “Grarep: Learning graph representations with global structural information,” in CIKM ’15: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management. Association for Computing Machinery, 2015, p. 891–900.

[13] M. Ou, P. Cui, J. Pei, Z. Zhang, and W. Zhu, “Symmetric transitivity preserving graph embedding,” in KDD’16 Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Association for Computing Machinery, 2016, p.1105–1114.

[14] Ahmed, N. Shervashidze, S. Narayanamurthy, V. Josifovski, and A. J. Smola, “Distributed large-scale natural graph factorization,” in Proceedings of the 22nd International Conference on World Wide Web, ser. WWW’13. New York, NY, USA: Association for Computing Machinery, 2013, p. 37–48.

[15] B. Shaw and T. Jebara, “Structure preserving embedding,” in Proceedings of the 26th Annual International Conference on Machine Learning, ser. ICML ’09. New York, NY, USA: Association for Computing Machinery, 2009, p. 937–944.

[16] Y. Uygun, et al., “On the large-scale graph data processing for user interface testing in big data science projects,” in 2020, IEEE International Conference on Big Data (Big Data), IEEE Computer Society, 2020, pp. 2049–2056.

[17] E. Olmezogullari and M. S. Aktas, “Pattern2vec: Representation of clickstream data sequences for learning user navigational behavior,” Concurrency and Computation: Practice and Experience, vol. 34, no. 9, p. e6546, 2022.

[18] E. Olmezogullari and M. S. Aktas, “Representation of click-stream datasequences for learning user navigational behavior by using embeddings,” in 2020 IEEE International Conference on Big Data (Big Data), IEEE Computer Society, 2020, pp. 3173–3179.

[19] M. Oz, et al. “On the Use of Generative Deep Learning Approaches for Generating Hidden Test Scripts”, International Journal of Software Engineering and Knowledge Engineering, Vol 31, Issue 10, p1447, 2021.

[20] I. Erdem, et al. “Test Script Generation Based on Hidden Markov Models Learning From User Browsing Behaviors”, 2021 IEEE International Conference on Big Data (Big Data), IEEE Computer Society, 2021.

[21] H. Vardar, et al. “Time-Sensitive Embedding for Understanding Customer Navigational Behavior in Mobile Banking”, International Conference on Computing, Intelligence and Data Analytics, 257-270, 2022.