On the Use of Predictive Deep Learning Approaches in the Frequency and Uniqueness-Based Representation of Sequential Browsing Events

Hakan Hakvar[1], Cansu Cavuldak[1], Oğulcan Söyler[1], Yusuf Subaşı[1], Yıldız Karadayı[1], Ilgın Şafak[1(✉)], Nail Taşgetiren[2], and Mehmet S. Aktaş[3]

[1] Fibabanka R&D Center, Istanbul, Türkiye
{hakan.hakvar,cansu.cavuldak,ogulcan.soyler,yusuf.subasi,
yildiz.karadayi,ilgin.safak}@fibabanka.com.tr
[2] Hepsiburada R&D Center, Istanbul, Türkiye
nail.tasgetiren@hepsiburada.com
[3] Yıldız Technical University, Computer Engineering Department, Istanbul, Türkiye
aktas@yildiz.edu.tr

Abstract. To gain a more comprehensive understanding of how users navigate, this study investigates the potential of predictive deep learning techniques. To properly describe user's browsing events such page visits, this paper first provides an overview of two embedding methodologies based on natural language processing and graphical representations. Secondly, this study presents two aggregation methods, one based on frequency and the other on uniqueness, that can be used to depict the possible sequence of events that may occur during a user's surfing session. With an emphasis on online banking, this research summarizes data representation approaches to model customers' digital footprints and learn about their banking product purchase intents. When we analyze a customer's digital footprint in the digital banking domain during the early stages of the application's lifecycle, we find that there are insufficient digital user footprints to model and comprehend users' behavior. Consequently, new approaches are needed to increase the quantity of session recordings kept by customers' digital footprints. We consider the possibility of increasing the quantity of digital footprints of bank customers by employing predictive deep learning techniques. To enhance the quantity of user navigational behaviors, we present a predictive technique based on long short-term memories. According to the findings, the newly generated customer sequences facilitate testing the suggested method, and accurately represent the initial client session data.

Keywords: Customer Embedding · Customer Behavior · Digital Banking · Graph Based Embedding · Representation Learning · Word Embedding

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
O. Gervasi et al. (Eds.): ICCSA 2024, LNCS 14814, pp. 83–100, 2024.
https://doi.org/10.1007/978-3-031-64608-9_6

1 Introduction

In the ever-changing realm of digital banking, the analysis and thorough comprehension of users' digital footprints become crucial undertakings with significant consequences. A user's digital footprint is comprised of a wide range of interactions, including the complex series of events that occur when he or she uses banking software. These footprints, consisting of patterns of movement, records of transactions, and changes over time, are extremely significant sources of information. Accurate modeling and interpreting these digital footprints enable us to get valuable insights into user habits, preferences, and intentions. We can then create strong predictive models and customize tailored services to enhance overall efficiency, security, and user experience, as well as to determine future directions in digital banking, and to promote a mutually beneficial relationship between financial institutions and their customers.

As part of the complex network of online traces, we examine specifically the surfing behavior of consumers on digital banking websites. These patterns, which represent how users interact with the online banking interface, contain a significant amount of important information that is crucial for understanding user habits and preferences. Modeling and comprehending these navigational sequences are of utmost importance, since they directly mirror the intentions of users, the paths they take to complete transactions, and their levels of involvement. Valuable information can be extracted by analyzing browsing behaviors to improve user experience, optimize service offerings, and strengthen security measures. This helps determining the future of personalized, efficient, and secure digital financial transactions.

The depiction of users' browsing behaviors, especially in the context of consecutive occurrences, is a complex and diverse problem. A major challenge stems from the intrinsic intricacy and variety of users each with distinct and changing navigational preferences. The ever-changing characteristics of digital interactions exacerbate the difficulty, as users connect with financial services on various devices and in different settings. Furthermore, large amounts of consecutive occurrences produced during user sessions present difficulties in terms of scalability, computational requirements and efficient representation. The complexities arising from temporal dependencies and contextual nuances in event sequences introduce an additional level of intricacy, requiring novel approaches to capture and simulate the dynamic nature of user interactions. Developing strong representation approaches is crucial in order to accurately capture the varied and ever-changing sequential browsing behaviors of users in digital banking settings.

Various methodologies, including natural language processing (NLP)-based, and graph-based techniques, show potential in accurately capturing the complexities of user behaviors in digital banking. Even so, there is a significant lack of research that examines the effectiveness of various methods for recognizing and describing user behaviors in the literature. While each strategy has its own advantages, little research has been conducted to explore how well they perform in accurately interpreting the intricate details of successive user interactions. Hence it is difficult to gain a comprehensive understanding of their strengths

and limitations, which hinders progress in optimizing and improving event modeling techniques.

Different methodologies may be used for combining representations of event sequences, especially in the context of user browsing sessions. This study uses frequency and uniqueness-based aggregation approaches to create embedding vectors that accurately capture the complex dynamics of user interactions. This enables the creation of a standardized and efficient embedding methodology specifically designed for event sequences that occur during user browsing sessions. We present strategies that combine different ways of representing events, creating a coherent and informative framework that can capture the complex patterns observed in the behaviors of digital banking users.

Most digital banking apps lack the user data necessary to model and understand customer behavior based on the examination of their digital footprints. This study provides a comprehensive analysis of how data representation is utilized to model users' digital behaviors and determine their level of interest in banking products. To this end, we update and extend our previous work [6], which is a machine learning (ML) pipeline that we implemented for predicting customer purchase behavior for a particular banking product, i.e., FX. We evaluate the prediction performance of the updated ML pipeline.

We also enhance frequency and inverse frequency-based weighting methodologies. In the frequency based weighting approach in [6], a frequency weight is calculated based on the weights of the pages visited by each customer. However, in this study, the optimized frequency-based weighting is achieved by normalizing the number of times a page appears in a session by the total number of pages in the session. Inverse frequency weighting reduces the importance given to the most visited pages by taking the inverse of the frequency of visits to the page for each customer. In contrast with the approach followed in [6], the inverse frequency-based weighting in this study involves taking the logarithm of the term for inverse frequency-based weighting in [6]. This logarithmic scaling is used to ensure that highly frequent web pages do not dominate the inverse page frequency score.

Furthermore, this study extends [6] by introducing a predictive deep-learning based methodology in enhancing banks' ability to gather more data on customer's digital footprints. We use a Long Short-Term Memory (LSTM) based predictive model to gain deeper insights into user behavior on the website. To facilitate testing of this method, we introduce an evaluation methodology that can measure how newly generated session sequences successfully represent the original user's browsing behavior. The ultimate goal is to create an optimized ML business process that can learn from the temporal sequences, recorded during user's browsing sessions, and that can predict user's next actions, such as purchasing actions. By conducting a methodical analysis, we aim to uncover the potential of these cutting-edge methodologies for collecting and analyzing the patterns of user interactions in the digital banking domain. This will ultimately enhance our understanding of user browsing behaviors.

Therefore, we address the following issues in this study:

1. How can we design a methodology that can synthetically generate customer session data using predictive deep learning techniques?

2. How can we understand whether the generated session sequences successfully represent the original user's browsing behavior?
3. How can we evaluate the effectiveness of the optimized version of the ML pipeline capability for predicting customers' financial behavior? This includes their likelihood of purchasing specific financial products or engaging in certain transactions.

The remainder of this article is organized as follows: Sect. 2 reviews the literature. Section 3 presents the methodology used in our research. Section 4 discusses the implementation of our framework and data set used in experimental work. Section 5 discusses the experimental work conducted. Conclusions and future work are presented in Sect. 6.

2 Fundamental Concepts and Literature Survey

To better predict customers' time-based activities in digital banking applications, compact and dynamic representations of customers are required. The use of embedding techniques provides a solution to this problem by digitizing customer transactions in banking applications and transforming their behavior into numerical representations that can be used to accurately predict their future behavior [2]. This section discusses the fundamental concepts of embedding mechanisms and reviews previous studies analyzing embedding approaches to representing user navigational behavior.

2.1 Embedding Approaches

Digital analytics and user behavior modeling require accurate representation of events in user browsing activity. One thus needs techniques to accurately represent customers' navigational footprints both over time and on a frequency basis. To this end, one may follow three separate embedding approaches, each based on core notions that support their effectiveness. Embeddings based on NLP utilize linguistic patterns to encode textual information, offering a nuanced comprehension of user interactions. Graph-based embeddings utilize network topologies to capture links and dependencies in browsing activity, providing a comprehensive understanding of user interaction. Autoencoder-based embeddings, in contrast, utilize neural network structures to acquire condensed representations of data, enabling effective storage and retrieval of significant insights. These three embedding approaches are crucial to investigate the core concepts necessary for the analysis and interpretation of user interactions in the digital environment. Embedding techniques have evolved over time into sub-fields such as word embedding and graph embedding.

NLP based Embedding Approaches: NLP-based embedding provides a vector representation of words in a corpus, which captures both semantic and syntactic information [33]. This technique enables similarity analysis, speech

labeling, clustering, and syntactic parsing in NLP applications [4]. Word embedding has facilitated numerous analyses and predictions in supervised, semi-supervised, and unsupervised ML algorithms by converting words to vectors [33]. The Word2Vec model consists of two approaches: Continuous Bag of Words (CBoW) and Skip-gram [34]. CBoW predicts a specific word by analyzing its neighboring words. Words are converted into one-hot encoded vectors and fed into a neural network, with the predicted word being output as the result. Skip-gram, on the other hand, predicts neighboring words around a specific word, which is then converted into a one-hot encoded vector and used as input to a neural network, with the neighboring words predicted as output [34]-[10].

Graph-Based Embedding Approaches: As opposed to word embedding, graph embedding uses graphs to represent nodes as vectors in a low-dimensional space. The graph-based embedding is a powerful technique to solve problems related to social networks, community detection, node classification [35] and clustering [30]. This embedding approach uses computationally intensive eigenvalue decomposition techniques [27]. The graph embedding approach is a powerful technique that solves problems related to social networks and community detection. This approach helps to create vectorized features for graph attributes, such as nodes, edges, and subgraphs [14]. Some common graph embedding methods are Node2vec [5], DeepWalk [20], Struc2vec [26], Graph Convolutional Networks (GCNs) [11], GraphSAGE [7], LINE [29], and TransE [1]. The Node2vec algorithm generates random walks on a graph and uses a skip-gram model to learn node embeddings. In the same way, DeepWalk generates random walks on a graph, but instead of a skip-gram model, it uses a neural network for learning node embeddings. Struc2Vec is a powerful method for graph embedding that is particularly effective at capturing the structural information of a graph. GNCs use convolutional neural networks (CNNs) to learn node embeddings by aggregating the features of neighboring nodes. GraphSAGE extends the GCN idea by using a differentiable aggregator function to combine neighboring nodes' features. LINE preserves local and global network structures by optimizing nodes' first- and second-order proximity, while TransE is a method for embedding directed graphs that utilizes a translation-based scoring function to learn embeddings.

Among these graph embedding methods, we are particularly interested in DeepWalk, a popular and effective graph embedding technique. This technique maintains higher-order proximity by maximizing the likelihood of monitoring the last node and subsequent nodes in each random walk [28]. It works in two stages: In the first stage, random walks are generated in the graph to optimize embedded representations of nodes by randomly wandering near neighbors [13]. In the second stage, the Word2Vec Skip-gram algorithm is applied to every created walk to vectorize the nodes. When DeepWalk is applied to successful word embedding, the nodes are set so that the occurrence frequencies of pairs in short random walks are preserved [27]. There are different implementations of DeepWalk graph representations, such as Hierarchical Softmax, which uses the Huffman binary tree as an alternative representation of the vocabulary [9].

DeepWalk has been used successfully in many applications, such as community detection, link prediction, and recommendation systems [27,31].

Autoencoder-Based Embedding Approaches: Autoencoder-based embeddings are notable in the field of analysis, data representation and feature learning since they can effectively capture nuanced patterns and inherent structures in large datasets. Autoencoders are unsupervised learning methods that utilize neural network architecture to encode input data into a compact representation known as an embedding. The encoder network facilitates the encoding process, which prioritizes the extraction of important elements while eliminating unnecessary information. The decoder network can reconstruct the input data from its compressed representation. Autoencoders intrinsically acquire the ability to accurately reproduce the original input, enabling them to effectively capture the fundamental characteristics of the data. This makes autoencoders a versatile tool that may be used for various tasks such as reducing dimensionality, detecting anomalies, and extracting features.

2.2 Other Embedding Methods

Other embedding methodologies have been proposed and surveyed in the literature [31]. Pattern-based embedding approaches [16,17] have been widely used to analyze customers' navigational behavior. A number of studies have employed word-based embedding techniques [3,15,18,32] to understand customer behavior and generate graphical customer interface testing scripts. There are, however, few comparative analyses of embedding methods in digital footprint studies. The embedding methods that take time-based datasets into account are insufficient.

3 Methodology

In this section, we discuss the contributions of this study in two fold. Firstly, we give an overview of a ML process that can analyze and model customer's browsing behavior data. Secondly, we introduce a methodology that synthetically generates customer user sequences in order to improve the prediction capability of the ML process.

3.1 An Overview of the ML Business Process

The customer embedding framework proposed in this study involves six stages, including data pre-processing, embedding, post-processing, training and prediction.

1. **Pre-processing stage:** Raw data, that refers to customers' digital footprints obtained from mobile application platforms, is cleaned, and irrelevant features are removed. Features selected are Customer Number, Page Sequence, Page Durations, Session Start Date, FX Event Flag (see Table 1 for a description of the features). Pages that customers visit by default, such as login and logout, and pages that are rarely visited, are eliminated. The time duration spent by a customer on each page is calculated.

Table 1. Browsing data on mobile banking application pages of bank customers

Feature	Explanation
Customer Number	Unique customer number
Page Sequence	Chronologically ordered page numbers the customer browsed in the mobile app during the session
Page Durations	How long has the customer been on a specific page (in seconds)
Session Start Date	The date and time the customer created the session via logging into the mobile app
FX Event Flag	Whether the customer executes the FX transaction during the session (1 = True, 0 = False)

2. **Embedding stage:** A suitable embedding technique is applied to the pre-processed data to represent customer behavior in a high-dimensional vector space. The word embedding method employed in this study is Word2Vec (CBoW). An extension to the DeepWalk algorithm is used as a graph embedding method to capture mobile banking customers' tendencies towards specific financial products. The proposed extension incorporates a new metric, namely the frequency with which customers visit the pages of a mobile banking application. To achieve this, we introduce frequency-based weighting approach at the post-processing stage.
3. **Post-processing stage:** The embeddings are further processed to eliminate noise and enhance embedding quality. This involves generating customer embedding vectors using visit frequency-based weighting techniques. Event frequency-based weighting: This approach, that involves calculating an event frequency value for each page based on the frequency of customer C visits, aims to represent customers' digital footprints. The event frequency weighted vector for a customer is obtained using the following equation:
   Event frequency-based weighting: Involves calculating an event frequency value for each page based on the frequency of customer visits. The event frequency weighted vector for a customer is obtained using the following equation:

$$NF_C = \frac{VF_C}{\sum_{i=1}^{K} f_C(i)}. \tag{1}$$

   Here, NF_C is the matrix consisting of customer C's normalized event frequency values of all sessions, VF_C is customer C's event frequency matrix of all sessions, K is the total number of sessions, and $f_C(i)$ is the total frequency of all pages visited by customer C in session i. The embedded session vectors are multiplied by this matrix to obtain the normalized event frequency based weighted session vectors. In this approach, the most frequently visited pages are given greater importance.

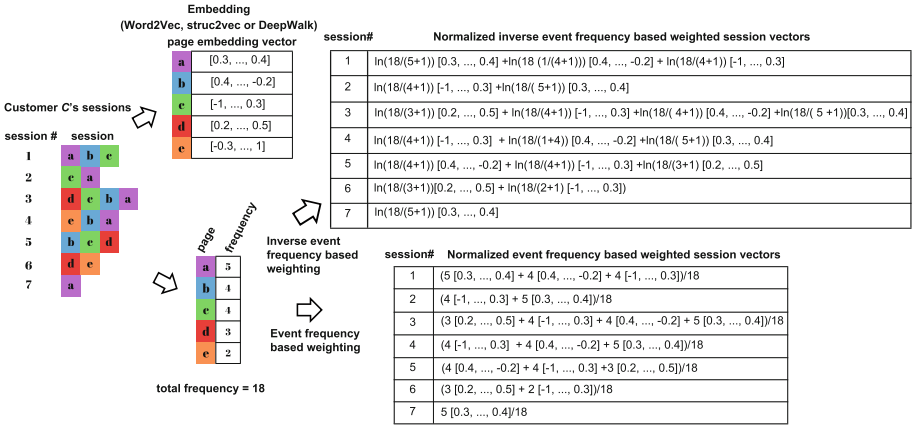

Fig. 1. An example of frequency and inverse frequency-based weighting mechanism

Let us consider the example with seven sessions as illustrated in Fig. 1. According to the session data, the customer visits five pages, namely a, b, c, d and e. The first session includes pages a, b and c, the second session includes pages c and a, and so on. The frequency of visits to each page is calculated, and the normalized event frequency-based weighting matrix is obtained using Eq. 1. Each page is represented with an embedding vector by using one of the embedding methods, Word2Vec or Deepwalk. Lastly, the embedding vectors are multiplied by the normalized frequency based weighting vector to obtain the normalized event frequency based weighted session matrix. In this example, the total page visit frequency count of all sessions is 18, which is used as the normalization value. Note that the same customer session data and embedding vectors are used in frequency, inverse frequency and event time duration based weighting methods.

Inverse event frequency-based weighting: In this weighting approach, more frequently visited pages are given less weight. This is obtained by taking the inverse and then the natural logarithm of Eq. 1, and adding 1 to the denominator in order to scale and prevent division by zero. The inverse frequency weighted matrix for a customer is thus given by:

$$INF_C = \ln\left(\frac{\sum_{i=1}^{K} f_C(i)}{VF_C + 1}\right). \tag{2}$$

Here, INF_C is the matrix consisting of the inverse of the frequency values of pages visited by customer C in all sessions. The embedded session vectors are multiplied by this matrix to obtain the normalized inverse event frequency based weighted session vectors. Overall, our proposed approach aims to represent customers' digital footprints. It also enables more accurate predictions of their behavior on mobile banking applications. This methodology answers the first research question mentioned in the Introduction section.

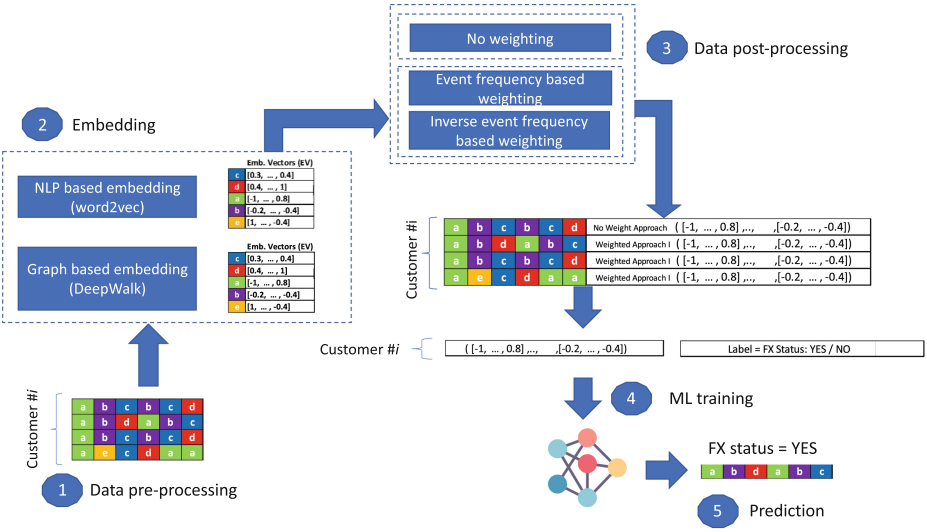

Fig. 2. An example flow using the customer embedding framework

An example flow using the customer embedding framework is illustrated in Fig. 2. In the example shown in Fig. 1, each page's visit frequency is obtained using the session data, and each page is represented with an embedding vector. The embedding vectors are multiplied by the normalized inverse frequency matrix obtained by using Eq. 2 to determine the normalized inverse frequency-based session vectors. After generating session embedding vectors using the DeepWalk algorithm and applying frequency-based weighting techniques, customer-based vectors are calculated to represent each customer with an embedding vector. The customer embedding vectors are generated by taking the arithmetic mean of the session embedding vectors. This represents the customer's overall navigational behavior. The average of all session embedding vectors for each customer results in a n dimensional customer-based vector, where the so-called framework parameter n shows the embedding vector's dimension. The customer-based vectors thus obtained are used as input for clustering algorithms to segment customers based on their navigational behavior.

4. **Training phase:** During this stage, the labeled embedding datasets are partitioned into a training dataset (80%) and a test dataset (20%). ML algorithms are trained on the training dataset to create ML models using customer embedding vectors to predict customer behavior. These models are then validated using the test dataset.
   The performance of the framework was evaluated by using the ML algorithm called as Isolation Forest that detects anomalies by assuming that instances that deviate from the central data cluster are abnormal. It uses random sampling to create binary trees and aggregates them to build a model for a given

dataset. This study utilized an unbalanced dataset due to a small number of customers conducting FX transactions. As a result, Isolation Forest was selected as the ML algorithm for testing the proposed framework [12].

5. **Prediction stage:** Resulting embeddings are utilized to predict customer behavior and preferences. The generated ML model is employed to forecast new users' navigational behavior.

3.2 A Novel Methodology for Generating User Sequences

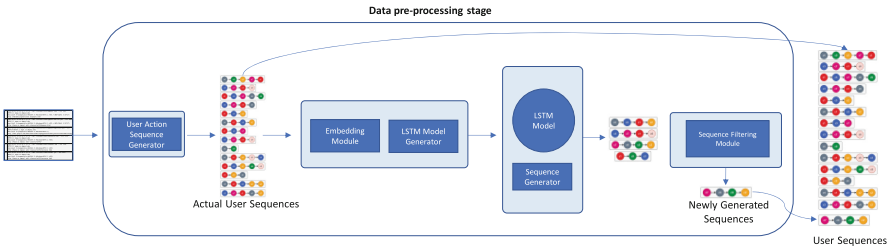

Fig. 3. Predictive Deep Learning Based Methodology for Generating User Sequences

When working with small labeled datasets in supervised ML, one of the techniques used is data augmentation. This involves generating new training samples from existing ones by applying various transformations. A predictive deep learning approach is used in this study to create hidden navigational behaviors of customers. The proposed approach is illustrated in Fig. 3. The details of the proposed data augmentation methodology is given below:

1. **User Action Sequence Generator:** The customer's navigational behaviors are reflected by user sequences. These sequences are then transformed by embedding so that they can be applied as input for an LSTM algorithm. LSTM-based predictive model is used for learning certain patterns from data, so that when presented with some prompt, it can produce an output in the same way as the pattern that was learned. Next, this model is used to generate new sequences. These newly generated sequences are checked against original user sequences. A final output is generated by filtering out only those sequences that differ from the actual user actions and that are feasible considering the user interfaces of digital banking mobile applications. As a result, the original and newly generated user action sequences are obtained.
2. **Embedding Module:** This module is responsible for converting the web page labels to integers. Each unique web page label is converted a distinct integer. With this approach, user sequences are represented as sequence of integers.

3. **Predictive Deep Learning Model Generator:** LSTM [8] is a type of recurrent neural network that is widely used in deep learning. It is particularly effective at capturing long-term dependencies, making it a suitable tool for predicting sequences.
4. **Sequence Generator:** This module is responsible for generating new sequences based on the trained LSTM model.
5. **Sequence Filtering Module:** This module is responsible for eliminating sequences that match to one of the actual sequences. This way, new and distinct sequences are generated.

4 Proposed Framework Implementation

The proposed framework was implemented using Python-3 (version 3.6.13) [24]. To generate the embedding vectors, we utilized the Gensim (version 4.2.0) library, which provides implementations of Word2Vec algorithms such as CBoW and Skip-gram [21]. For converting sessions to graphs, we used the NetworkX (version 2.5.1) library [22]. Pandas (version 1.1.5) and Numpy (version 1.19.5) libraries were utilized for data processing operations [23,25]. We employed the scikit-learn (version 0.24.2) library for implementing the Isolation Forest model and metrics. To evaluate the framework's functionality, we conducted tests using clickstream data from Fibabanka's mobile banking application. The dataset consists of 2,971,999 rows with five attributes and 36,160 distinct customers. The data captures online activities of mobile banking customers, including page views and actions such as FX transactions, loan applications, and account openings.

Figure 4 depicts the histogram of the session lengths per customer in the dataset. It is observed that the majority of customers have sessions with less than 200 events.

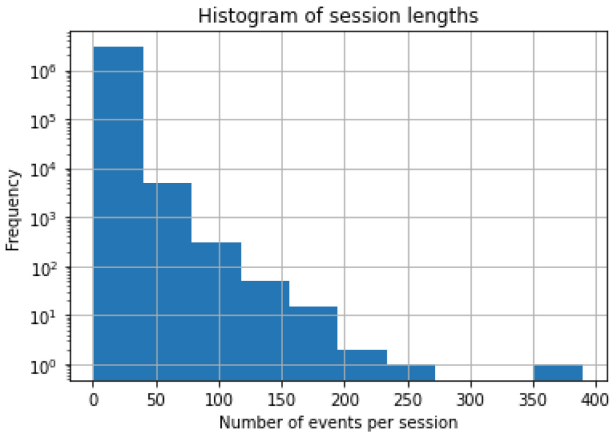

Fig. 4. Histogram of the number of events per session per customer

Table 2. FX rate in dataset

FX Status	Train	Test
FX=1	4.39	1.07
FX=0	24.53	6.15
FX=1 Rate	15.2%	14.9%

Table 2 shows the rate at which customers perform FX transactions in sessions in the dataset. Here, FX=1 means that an FX transaction occurs, and FX=0 implies that no FX transaction occurred in a customer session. 'FX=1 Rate' is the percentage of FX transactions in the test or train dataset.

Each session in the clickstream data includes the unique customer number, pages visited during the session, time spent on each page, date and time of the session, and whether the customer executed the target event, which in this case, is an FX transaction.

The study focused on customers who had 50 or more sessions during 76 d between April 1, 2022, and June 15, 2022. By analyzing the dataset, we predicted customer behavior over an extended period and comprehensively analyzed customer intent and behavior.

The clickstream data captured for each session includes two significant features with equal lengths: "Page Sequence" and "Page Durations". The "Page Sequence" shows the list of pages a customer navigated throughout a session in the order they were visited. For example, if a customer visited the Login page, Main Page, Loan Application, Calculate Loan, Main Page, Money Transfer, and Logout page in that order, the "Page Sequence" feature would be represented as [Login, Main Page, Loan Application, Calculate Loan, Main Page, Money Transfer, Logout]. The customer's overall navigational behavior can be better described by using the "Page Sequence" data to calculate the frequency of each page sequence visited by each customer. Frequency-based weighting enhances the framework's accuracy.

The "Page Durations" feature represents a list of time spent on each page visited by the customer during the session, in seconds. For instance, the time spent on each page of the "Page Sequence" above is represented by the "Page Durations" [0.5, 0.7, 1.1, 1.2, 0.5, 2.4, 0.2]. "Page Durations" data is used to compute customer-based time duration weighting for the embedding vectors in order to improve the framework's accuracy in predicting customers' intent, as the duration spent on a specific page indicates interest.

The proposed framework's data pre-processing stage involves several steps. To begin with, we remove some pages, such as login and logout pages, as well as pages rarely visited by customers, from the clickstream dataset pages. After eliminating these pages, the duration of each page in each session is calculated. If there are any missing page duration values, the average page duration of all sessions for the customer is calculated to replace the null values.

In the embedding stage of the framework, we employed the embedding methodologies, Word2Vec CBoW and DeepWalk, to generate session-based representations in order to capture customer behavior and intent effectively. To further enhance the generated embedding vectors, we utilized novel frequency-based weighting techniques in the post-processing stage.

We analyzed data from previous mobile sessions to predict whether customers would buy or sell FX in their next mobile session. To accomplish this, we used the "FX Event Flag" attribute as the target label for our dataset. We experimented with different empirically-selected embedding vector sizes of 20, 50, and 100. Due to the relatively small size of our training dataset, which contains only 36K distinct entities, we selected smaller vector sizes.

5 Experimental Work

We conducted two different types of experiments in order to determine how practical the study's frameworks were. The first examines how well synthetic user sequences are generated, and the second examines how well optimized ML pipelines are implemented. The second research question posed in the Introduction is addressed by the first experimental study, and the third research question by the second.

First Experimental Study employed an Autoencoder LSTM architecture to generate novel yet plausible synthetic sessions. Aiming to ensure the novelty of the generated sessions, we eliminated those that were exact duplicates of existing sessions and selected only the technically feasible sessions from the remainder. We then measured the similarity between the generated sessions and the existing ones using the Bilingual Evaluation Understudy (BLEU) score metric [19]. A BLEU score is a value between zero and one that determines how similar the machine-translated text is to a collection of high-quality reference translations. A BLEU score close to unity indicates a high degree of similarity between the machine-generated text and the reference translation. The newly generated session set and the eliminated session set were compared with the existing sessions. The sessions with the highest BLEU scores were identified as the most suitable for adding new sessions (see Table 3).

As for the second Experimental Study, we investigated the prediction performance of the optimized/updated ML pipeline and compared the results against our previous study. For our experiments, we utilized two base embedding methodologies, namely Word2Vec and DeepWalk. These methodologies are applied directly to customers' navigational behavior data to create benchmark datasets. These datasets are then used in the training and prediction stages without transformations and weighting methods introduced in our proposed framework. The models are trained on unbalanced datasets in order to predict customers who execute FX transactions. The results of experiments conducted to evaluate the performance of the proposed optimized ML pipeline in predicting customers who execute FX transactions are presented in Table 4 for an embedding vector size of 100 and prediction using the Isolation Forest ML algorithm.

Table 3. BLEU Score of generated sequence and most similar sequence

No.	Generated Hidden Sequence	Most Similar Sequence	BLEU Score
1	PRODUCTS ACCOUNT DETAIL, ALL DOCUMENTS, PRODUCTS ACCOUNT ACTIVITIES, TRANSACTIONS, FIBA FX	PRODUCTS ACCOUNTS, PRODUCTS ACCOUNT DETAIL, PRODUCTS ACCOUNT ACTIVITIES, HOME PAGE, FIBA FX	0,75984
2	LOGIN, HOME PAGE, PPPPSCREENTSX, PPPPSCREENTSX, TRANSFER ORDER DETAIL	LOGIN, HOME PAGE, PRODUCTS ACCOUNTS, TRANSACTIONS, FIBA FX	0,70711
3	HOME PAGE, NOTIFICATIONS, HOME PAGE, TRANSACTIONS, FIBA FX	HOME PAGE, NOTIFICATIONS, PRODUCTS ACCOUNTS, TRANSACTIONS, FIBA FX	0,99871

The results indicate that the optimized ML pipeline (Isolation Forest pipeline) prediction performance is improved as in the followings: a) for frequency based weighting mechanism and word2vec embedding approach, F1 Score is improved approximately by 20%; b) for inverse frequency based weighting mechanism and word2vec embedding approach, F1 score is improved approximately by 31%; c) for frequency based weighting mechanism and deepWalk embedding approach, F1 Score is improved approximately by 15%; d) for inverse frequency based weighting mechanism and word2vec embedding approach, F1 score is improved approximately by 6%;

Table 4 show the F1 score, Precision, Recall and Accuracy for different weighting options of the optimized Isolation Forest ML pipeline. The F1 score provides a balanced view of Precision and Recall, making it an effective means of evaluating the performance of models in imbalanced datasets. This is valid in this case since positive cases are rare. The proposed framework's prediction outputs are compared with base models' plain embeddings, i.e., Word2Vec and DeepWalk embeddings. Hence, NONE stands for no weighting, FREQ shows frequency-based weighting and FREQ-1 denotes inverse frequency-based weighting. Table 4 shows that, for Word2Vec embedding, both the frequency based weighting and inverse frequency based weighting methods outperform the baseline performance where no weighting, NONE. Inverse event frequency-based weighting displays the highest performance. For DeepWalk embedding, the frequency based weighting method is seen to have a higher performance than the baseline model's performance, NONE. This experimental study answers research question 3 listed in the Introduction.

Table 4. Experimental study results for the optimized Isolation Forest ML Pipeline

	Word2Vec				DeepWalk			
	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1
NONE	74,01%	14,86%	15,16%	15,01%	74,09%	14,90%	15,68%	15,28%
FREQ	74,40%	15,12%	14,98%	15,05%	74,60%	15,66%	16,05%	15,85%
FREQ-1	74,20%	15,43%	15,71%	15,57%	74,17%	13,76%	13,91%	13,84%

As a result, Table 4 shows that the proposed framework improves the performance of base embedding models in predicting customers who execute FX transactions. The best performance is achieved with inverse frequency-based weighting. The proposed framework generally outperforms the base embedding models in terms of F1 score for predicting customers who execute FX transactions. As demonstrated by the experimental results, the proposed framework is more effective at predicting FX transactions than the baseline embedding models. Also note that the F1 scores of the proposed framework outperforms the base models in most cases. This indicates that the weighting techniques used in the framework effectively improve the base models' performance. It is observed that the performance of the models depends on the size of the vector, and larger vector sizes generally result in better results. However, it is worth considering the computational complexity of larger vector sizes and balancing this against potential performance improvements. Overall, these results suggest that the proposed framework can effectively improve the performance of embedding models in predicting customer behavior on online platforms. These results provide an answer to research questions listed in the Introduction section.

Overall, the study provides insights into the application of embedding models for predicting customer behavior in the FX domain. It proposes a novel framework to improve prediction accuracy. Further experimentation may be needed to identify the optimal approach for various types of data. Overall, the results demonstrate the potential of the proposed framework to improve customer behavior prediction tasks. Future research may investigate how the proposed framework performs in other financial domains and the further evaluation of its performance using larger datasets.

6 Conclusions and Future Work

This study introduces a novel customer embedding framework that leverages temporal and repetitive dynamics of navigational footprints generated in mobile banking applications. The framework utilizes cutting-edge embedding algorithms such as Word2Vec and DeepWalk, incorporating the overall customer tendency based on visit frequency to enhance the generated embedding representations. Using embedding vectors to predict customers' preferences for a financial product, the effectiveness of the proposed framework was evaluated. The results indicated that the proposed framework outperformed the baseline models in almost all test scenarios, and demonstrated its ability to capture customers' intent more accurately. It is important to note that the choice of embedding algorithm may vary depending on the dataset and the specific problem at hand. Further experimentation may be necessary to identify the optimal approach for different types of data. This study contributes to the development of a framework for capturing customer intent using digital financial footprints in mobile banking applications. The potential applications of this framework are not limited to the banking industry, but can also be applied in other fields where customer behavior analysis plays a key role, including e-commerce, healthcare, and social media. The framework proposed provides an avenue for organizations to better understand their

customers and tailor their services accordingly. Aside from the findings presented in this study, the authors intend to further improve their pro- posed framework by incorporating multi-modal data and improving its ability to predict customer intention towards a wide range of products. In addition, they intend to incorporate RNN-based algorithms into the framework to predict when customers will use specific products in the near future.

Acknowledgements. The authors of this study would like to express their gratitude to Fibabanka for their support in facilitating this research. In addition, one of the authors, Nail Tasgetiren, would like to thank Hepsiburada for providing the flexible working atmosphere to support this study.

References

1. Bordes, A., et al.: Translating Embeddings For Modeling Multi-relational Data. In: Advances in Neural Information Processing Systems. Burges, C.J. et al. (eds.), vol. 26. Curran Associates, Inc. (2013)
2. Chitsazan, N., et al.: Dynamic customer embeddings for financial service applications (2021). arXiv: 2106.11880
3. Erdem, I., et al.: Test script generation based on hidden markov models learning from user browsing behaviors. In: 2021 IEEE International Conference on Big Data (Big Data), pp. 2998–3005 (2021)
4. Ghannay, S., et al.: Word Embedding Evaluation and Combination. In: Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016). Calzolari, N.: (eds.) et al. Portorož, Slovenia: European Language Resources Association (ELRA), May 2016, pp. 300–305 (2016)
5. Grover, A., Leskovec, J.: Node2vec: scalable feature learning for networks. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD 2016. San Francisco, California, USA: Association for Computing Machinery, pp. 855–864 (2016). ISBN: 9781450342322
6. Hakvar, H., et al.: Time-sensitive embedding for understanding customer navigational behavior in mobile banking. In: Computational Intelligence, Data Analytics and Applications. García Márquez, F.P. et al. (eds.), pp. 257–270 Springer International Publishing, Cham (2023). ISBN: 978-3-031-27099-4, https://doi.org/10.1007/978-3-031-27099-4_20
7. Hamilton, W.L., Ying, R., Leskovec, J.: Inductive representation learning on large graphs (2018). arXiv: 1706.02216
8. Hochreiter, S., Schmidhuber, J.: Long Short-Term Memory. In: Neural Computation 9.8, pp. 1735–1780 (Nov. 1997). ISSN: 0899-7667
9. Hou, M., et al.: Network embedding: taxonomies, frameworks and applications. Comput. Sci. Rev. **38**, 100296 (2020). ISSN: 1574-0137
10. Jang, B., Kim, I., Kim, J.W.: Word2vec convolutional neural networks for classification of news articles and tweets. PLOS ONE **14**(8) 1–20 (2019)
11. Thomas N. Kipf and Max Welling. "Semi-Supervised Classification with Graph Convolutional Networks". In: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings. OpenReview.net, 2017

12. Ajay Kulkarni, Deri Chong, and Feras A. Batarseh. "5 - Foundations of data imbalance and solutions for a data democracy". In: Data Democracy. Ed. by Feras A. Batarseh and Ruixin Yang. Academic Press, 2020, pp. 83- 106. ISBN: 978-0-12-818366-3
13. Juzheng Li, Jun Zhu, and Bo Zhang. "Discriminative Deep Random Walk for Network Classification". In: Annual Meeting of the Association for Computational Linguistics. 2016
14. Makarov, I., et al.: Survey on graph embeddings and their applications to machine learning problems on graphs. PeerJ Comput. Sci. **7**, e357 (2021). ISSN: 2376-5992
15. Oguz, R.F., Erdi Oz, M., Olmezogullari, E., Aktas, M.S.: Extracting information from large scale graph data: case study on automated UI testing. In: Chaves, R., et al. Par 2021: Parallel Processing Workshops, vol. 13098. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-06156-1_29
16. Olmezogullari, E., Aktas, M.S.: Pattern2Vec: representation of clickstream data sequences for learning user navigational behavior. Concurrency Comput. Pract. Experience **34**(9), e6546 (2022)
17. Olmezogullari, E., Aktas, M.S.: Representation of click-stream data sequences for learning user navigational behavior by using embeddings. In: 2020 IEEE International Conference on Big Data (Big Data), pp. 3173–3179 (2020)
18. Oz, M., et al.: On the use of generative deep learning approaches for generating hidden test scripts. Int. J. Software Eng. Knowl. Eng. **31**, 1447–1468 (2021)
19. Papineni, K., et al.: BLEU: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics. ACL 2002. Philadelphia, Pennsylvania: Association for Computational Linguistics, pp. 311–318 (2002)
20. Perozzi, B., Al-Rfou, R., Skiena, S.: DeepWalk: online learning of social representations. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD 2014. Association for Computing Machinery, pp. 701–710 (2014)
21. Python Gensim library offical website. http://pypi.org. Accessed 22 Nov 2023
22. Python Networkx library offical website. http://networkx.org. Accessed 22 Nov 2023
23. Python Numpy library offical website. http://numpy.org. Accessed 22 Nov 2023
24. Python offical website. http://www.python.org. Accessed 22 Nov 2023
25. Python Pandas library offical website. http://pandas.pydata.org. Accessed 22 Nov 2023
26. Ribeiro, L.F.R., Saverese, P.H.P., Figueiredo, D.R.: struc2vec: learning node representations from structural identity. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD 2017. Halifax, NS, Canada: Association for Computing Machinery, pp. 385–394 (2017). ISBN: 9781450348874
27. Rossi, R.A., Zhou, R., Ahmed, N.K.: Deep inductive graph representation learning. IEEE Trans. Knowl. Data Eng. **32**(3), 438–452 (2020)
28. Rizi, F.S., Granitzer, M.: Properties of vector embeddings in social networks. Algorithms **10**(4) (2017). ISSN: 1999-4893
29. Tang, J., et al.: LINE: large-scale information network embedding. In: Proceedings of the 24th International Conference on World Wide Web. WWW 2015. Florence, Italy: International World Wide Web Conferences Steering Committee, pp. 1067–1077 (2015). ISBN: 9781450334693

30. Tasgetiren, N., Şafak, I., Aktas.,M.S.: On the use of graph embedding techniques for clustering user browsing navigational behaviours. In: International Journal of Web and Grid Services, Special Issue for ICCSA 2023 (2024)
31. Taşgetiren, N., Aktas, M.S.: Mining web user behavior: a systematic mapping study. In: Computational Science and Its Applications - ICCSA 2022 Workshops. In: Gervasi, O., Murgante, B., Misra, S., Rocha, A.M.A.C., Garau, C. (eds), pp. 667–683. Springer International Publishing, Cham (2022). ISBN: 978-3-031-10536-4, https://doi.org/10.1007/978-3-031-10536-4_44
32. Uygun, Y., 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), pp. 2049–2056 (2020)
33. Wang, B., et al.: Evaluating word embedding models: methods and experimental results. APSIPA Trans. Signal Inform. Process. **8**, e19 (2019)
34. Wu, L., et al.: Word mover's embedding: from word2vec to document embedding. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. Riloff, E. (ed.) et al. Brussels, Belgium: Association for Computational Linguistics, Oct. 2018, pp. 4524–4534 (2018)
35. Zuckerman, M., Last, M.: Using graphs for word embedding with enhanced semantic relations. In: Proceedings of the Thirteenth Workshop on Graph-Based Methods for Natural Language Processing (Text- Graphs-13). Ustalov, D. (ed.) et al. Hong Kong: Association for Computational Linguistics, Nov. 2019, pp. 32-41 (2019)