

Aging-Based Weighting for Session Classification in User Behavior Analysis

Nail Taşgetiren[1(✉)], Ilgn Şafak[2,3], and Mehmet S. Aktaş[4]

[1] Hepsiburada Research and Development Center, Istanbul, Turkey
nail.tasgetiren@hepsiburada.com
[2] Fibabanka R&D Center, Istanbul, Turkey
ilgin.safak@fibabanka.com.tr
[3] University of Jyväskylä, Jyvskylä, Finland
ilgin.i.safak@jyu.fi
[4] Yildiz Technical University, Istanbul, Turkey
aktas@yildiz.edu.tr

Abstract. Comprehending user behavior on e-commerce platforms is essential for augmenting customer interaction and refining recommendation algorithms. Clickstream data provides a significant resource for examining user navigation patterns; nevertheless, accurately describing and categorizing user sessions poses a challenge. This paper shows how to improve the accuracy of session-based classification using an embedding-based method combined with an aging-based weighting mechanism. The considered embedding methods, Word2Vec, Node2Vec, and LSTM Autoencoder, can turn session-based clickstream data into numbers. Furthermore, a dynamic weighting technique is introduced to emphasize recent interactions to improve classification performance. Our empirical assessment on an authentic e-commerce dataset reveals that the LSTM Autoencoder surpasses conventional embedding methods in capturing sequential dependencies. In addition, the age-based weighting technique markedly improves the classification accuracy, especially when used with deep learning models. A comparison of different classification algorithms, such as Random Forest, Logistic Regression, Gaussian Naive Bayes, and LSTM, shows that LSTM models are the best at finding correlations between events over time. The results also show the importance of temporal weighting in session-based clickstream analysis and provide a solid foundation for further research in behavioral analytics and personalized recommendation systems. This paper introduces an efficient method for clickstream-based user modeling that facilitates better user engagement in e-commerce systems.

Keywords: Clickstream data · E-commerce · Word2Vec · Node2Vec · LSTM Autoencoder · Aging-based weighting · User behavior modeling · Classification algorithms

A patent application related to this work has been filed by the author's institutions.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
O. Gervasi et al. (Eds.): ICCSA 2025 Workshops, LNCS 15886, pp. 16–34, 2026.
https://doi.org/10.1007/978-3-031-97576-9_2

1 Introduction

The swift growth of digital commerce has led to an unparalleled collection of user interaction data. Comprehending and analyzing this data is essential for improving user experience, refining recommendation systems, and augmenting conversion rates. Clickstream data, which records users' consecutive interactions on e-commerce platforms, is one of the most effective sources for evaluating online user behavior [13,22]. However, traditional data embedding methods may not account for user interactions in time [18,23], even though they are good at turning raw clickstream data into structured formats. This constraint hinders their capacity to adapt to evolving user behavior over time.

To solve this problem, new time-sensitive embedding strategies use different weighting algorithms to better represent user interactions over time [10,35]. Aging-based weighting shows the user actions based on how recently they happened. This technique enhances predictive modeling by emphasizing new interactions and progressively diminishing the impact of previous activities, so it accurately reflects changing user preferences.

Standard embedding methods, such as Word2Vec [4] for sequence-based representations and Node2Vec [7] for graph-based representations, are useful for finding important patterns in user navigation data. Nevertheless, these may result in inadequate representations, as they do not consider changes in user behavior with time. On the other hand, sequence-to-sequence architectures like Long Short-Term Memory (LSTM) Autoencoders [19] have shown great success in modeling complex time-dependent interactions within sequential data. However, the incorporation of aging-based weighting into LSTM Autoencoders remains a mostly unexamined field, especially concerning the representation of e-commerce session data.

This paper presents an innovative aging (time)-sensitive weighting method in LSTM Autoencoder-based embedding representations for improving session-based user behavior modeling in e-commerce systems. This approach makes it easier to understand the temporal behavior of user activity more dynamically and flexibly. This method provides a more detailed and accurate picture of how sessions work, and improves the accuracy of classification tasks predicting user behavior. This paper is structured as follows: Sect. 2 provides an overview of the literature review. Section 3 presents the methodology. Section 4 describes the implementation and evaluation details of the proposed weighting methodology. The conclusions and future work are the focus of Sect. 5.

2 Literature Review

In data analytics, especially within e-commerce, the study of user interactions via clickstream data has become increasingly vital [2,34]. Traditional embedding methods, such as natural language processing (NLP) models like Word2Vec [4,26,29] and graph-based approaches like Node2Vec [7,17,38], map unstructured user activities into structured insights to the user behavior. However,

as these methods do not account for the relative importance of different data points, they may lead to generalized and not sufficiently accurate conclusions in behavioral modeling. Recent research focused on graph embedding methods that use deep learning, specifically LSTM autoencoders [9,19,28]. [28] proposed to embed weighted graphs using LSTM autoencoders to get node-weight sequences and turn them into fixed-length vectors. [31] employs LSTM autoencoders to combine graph sequences from random walks, breadth-first search, and shortest paths. Both approaches exhibited efficacy in graph classification and similarity tasks. [6] classified graph embedding techniques into three groups: random walks, factorization methods, and deep learning approaches. They also launched GEM, an open-source Python library for graph embedding. [36] reviewed diverse deep learning techniques for graphs, encompassing graph recurrent neural networks, convolutional networks, autoencoders, reinforcement learning, and adversarial approaches while emphasizing their applications and prospective future trajectories.

Prior studies are mostly based on simplistic weighting strategies or using a particular data type. Some recent studies have investigated aging-based weighting for data embeddings in multiple areas. FedTime integrates data age into federated learning for smart home energy prediction, enhancing consumption projections [30]. [24] studies aging pathways to show how the hierarchical organization of brain networks changes with age by using hyperbolic embeddings of MEG data. The embedding of self-estimated residual age is used in a unified network architecture to demonstrate that face aging occurs continuously for each person [15]. However, the results of sentiment analysis algorithms and word embeddings often have age-biased problems [5]. These results show the importance of including age-related variables in data embeddings for different uses, such as energy forecasting, neuroscience, facial recognition, and language-based models.

Prior studies either solely employed temporal dynamics in language models or graph structures, or employed static weighting schemes. A method that has not received enough attention in the literature is the simultaneous addition of aging-based temporal weighting across several embedding paradigms. In this paper, three different embedding methods are considered to show how users navigate. These include a natural-language-based model (Word2Vec), a graph-based model (Node2Vec), and one that works with a sequence-to-sequence model (the LSTM Autoencoder). An aging-based weighting method with the LSTM Autoencoder enhances the accuracy of user session data modeling. To the best of our knowledge, this is the first study to combine Word2Vec, Node2Vec, and LSTM Autoencoder in a session-based classification approach that considers time when assigning weights. Embeddings and F1 scores are also employed in diverse data categorization tasks. Domain-specific word embeddings are better at recognizing named entities than general embeddings, even when the corpus size is small [37]. Furthermore, tabular embeddings have been created to improve the precision of column and tuple categorization in relational data, demonstrating significant enhancements in F-measure relative to models lacking embedding [12].

Recent studies have tackled various aspects of information systems, user behavior modeling, and data representation across diverse application domains. The authors of [1] proposed a scalable infrastructure with fault-tolerant, high-performance information services in distributed systems. The authors of [27] performed a systematic mapping of mobile application verification techniques, underscoring the importance of software quality. In the context of clickstream data, the authors of [20] investigated how user navigational behavior can be modeled using embeddings extracted from clickstream sequences, initially focusing on representation methods and later enhancing them with Pattern2Vec to capture more structural and sequential user behavior patterns [21]. Furthermore, the authors of [33] presented a provenance-aware data collection platform for scientific modeling, emphasizing traceability and transparency in simulation workflows. While these studies contribute to user modeling, provenance tracking, and system robustness, none directly combine aging-based temporal weighting with multiple embedding strategies (Word2Vec, Node2Vec, LSTM Autoencoder) for session-level classification, which is the novel focus of this study.

3 Methodology

This paper presents a novel approach to group user sessions with similar navigation patterns by using users clickstream data from a monitored e-commerce website. Figure 1 shows the proposed business process that consists of several modules, such as the embedding function module, the weighting function module, the weighted average-based vectorization module, and the classification module. Together, these tools properly analyze user session data, enabling the creation of user groups with similar behavior patterns.

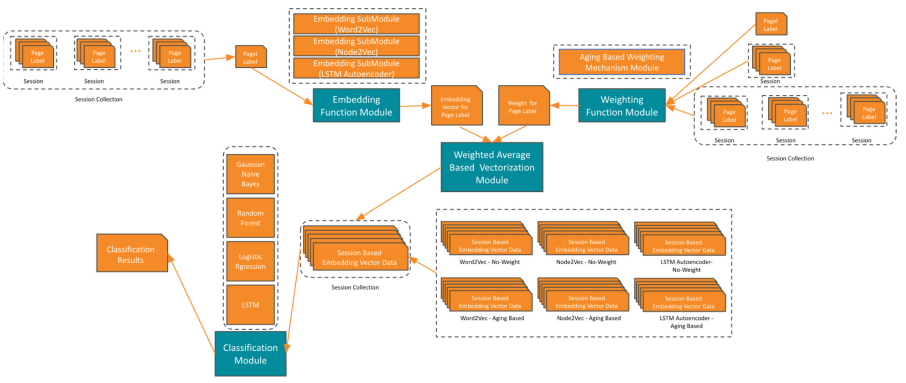

Fig. 1. Overview of the proposed business process

3.1 Embedding Function Module

The web log file provides extensive session information on user activities, with each click event monitored under a distinct sessionID. The embedding function module is an important part of this process since it uses different embedding techniques to encode session-based clickstream data into numerical vectors. This module contains a specific function for adaptively selecting the most appropriate embedding method to transform a page label into a numerical vector for precise and efficient data representation.

Word2Vec is an NLP-based embedding approach that generates vector representations for words. In this paper, page identifiers in the clickstream data are vectorized instead of words. The two fundamental architectures of Word2Vec are Continuous Bag of Words (CBOW) and Skip-Gram. The Skip-Gram technique was used in this paper to predict surrounding words (page IDs) that form the context of a specified target word (or page ID in this instance). Word2Vec lets you record the semantic links between web pages, bringing pages that are similar and close to each other in the resulting vector space. This approach makes it easier to model sequential interactions during user sessions and gives a good way to look at clickstreams.

In Word2Vec, each distinct token (such as a page ID in our dataset) is associated with a particular row in an embedding matrix. During the training phase, the model tries to find the best weights for this matrix so that tokens in similar situations are placed closer together in the vector space. This optimization aims to maximize the cosine similarity (or inner product) of adjacent embeddings while minimizing the similarity between randomly chosen embeddings. The CBOW model locates the target token by summing up the embedding vectors of adjacent tokens. Conversely, in the Skip-Gram paradigm, the target token infers the surrounding context tokens. Cosine similarity quantifies the alignment between vectors, resulting in related token vectors becoming more proximate. This method helps Word2Vec effectively capture the semantic and contextual links within sequential data, providing a strong representation to discover how users navigate.

Node2Vec elucidates the relationships among nodes by generating random walks within a graph, and provides a balanced exploration strategy by integrating Depth-First Search (DFS) and Breadth-First Search (BFS) algorithms. Random walks are stochastic systems characterized by probabilistic transitions among nodes. At each stage, a stochastic transition occurs from a specified node to the subsequent node, without following a fixed predetermined sequence. Two critical parameters govern the behavior of random walks: P (backward parameter) and Q (inside-outward exploration parameter). A high P value decreases the likelihood of retracing steps, so extending the walk and facilitating the exploration of a broader area inside the graph. A low P value increases the frequency of the random walk returning to the starting point, resulting in more localized movement. Similarly, a low Q value makes it more likely for the random walk to cover a larger area. A higher Q value makes it more likely for the walk to stay in a smaller area.

The LSTM Autoencoders follow sequence-to-sequence approach and excel in capturing complex temporal patterns and long-term dependencies. The LSTM autoencoder algorithm shows how nodes and their connections change over time in a network. Unlike Word2Vec or Node2Vec, the LSTM autoencoder doesn't use node identities (IDs) as input and doesn't have a separate layer that turns these IDs into embeddings. Each node is initially characterized by a distinct collection of attributes. These include fixed data like a node's degree and centrality, changing data about interactions like how often connections happen over time, or already calculated token embeddings if available. These attributes are represented in a time series format to capture temporal dependencies and trends.

For this method to work, the encoder packs sequential data into a small hidden state that holds both structural and temporal connections. The hidden state is then converted into an embedding vector containing the temporal dynamics of the node. The decoder subsequently reconstructs the original sequential data, maintaining both short-term and long-term dependencies. So, the LSTM autoencoder turns the nodes into numerical vectors that show their structural and temporal properties. The flexible design of the LSTM autoencoder allow to model sequential data effectively and to create meaningful representations that do not just rely on token embeddings.

NLP-based embedding methods are great at representing meaning linearly, while graph-based methods focus on learning about structural relationships. As for sequence-to-sequence approaches, they are successful in modeling sequence data changes over time.

3.2 Weighting Function Module

The weighting function module is a framework that calculates the suitable weight values for the data by implementing the specified weighting algorithm. This module uses aging-based weighting, which changes the importance of data items based on how they interact with each other over time. Unlike static weighting methods, aging-based weighting puts more weight on new interactions while decreasing the importance of older data over time. This method ensures that the weighting system adapts to changes in users behave over time. This improves the description of session-based interactions. These steps are activated according to the chosen weighting method, ensuring that the data is processed following the analytical framework.

Module for Weighted Session-Embedding Vector Calculation Aging-Based Approach. This module comprises code components that transform session-based clickstream data into numerical vectors via diverse embedding techniques. Traditional techniques assign equal significance to each page ID, but the aging-based weighting approach assigns weights to pages based on their visitation times during the session. This approach prioritizes more recently visited pages, hence effectively reflecting the session's dynamic character. This approach offers a robust mechanism for simulating the evolution of users' interests and browsing behaviors over time.

This paper uses an aging-based weighting approach to transform session-based clickstream data into numerical vectors. While each pageID has equal weight in traditional no-weight methods, the aging-based weighting method assigns more weight to pages visited more recently. Hence, the dynamic structure of the session over time is better represented.

We calculate the embedding vectors of sessions by utilizing various embedding vectors from each page. The aging-based weighting approach assigns a time-dependent weighting to each page. The mathematical representation of this method is given in (Eq. 1).

$$EV_{S_i} = \frac{\sum_{j=1}^{SessionLength_i} w_j EV_{P_j}}{\sum_{j=1}^{SessionLength_i} w_j}, \forall i \in 1, 2, \ldots, M \tag{1}$$

where;

(*) EV_{S_i}: Embedding Vector of Session i, and

(*) i is the index for the sessions in the log file (session-based clickstream data), and

(*) EV_{P_j}: embedding vector of the page ID of page j (p_j) , and

(*) j is the index for the page ID in the Session i (S_i) and ranges from 1 to $SessionLength$ of (S_i), and

(*) $SessionLength_i$: Total number of page IDs in the Session i (S_i), and

(*) M is the total number of the sessions, obtained from the session-based clickstream data, and

(*) w_j: Aging-based weight assigned to the page p_j in Session i, computed as:

$$w_j = \frac{1}{Recency_j + 1} \tag{2}$$

where $Recency_j$ represents the recency value of the page p_j based on its timestamp in the session.

The aging-based weighting algorithm provides time-dependent weights depending on page visit times during the session. The vector representation of the session is created by giving more importance to newer pages while decreasing the influence of older pages. The page weights are inversely proportional to time as defined in Eq. 2, where the $Recency_j$ is calculated based on the time of the page visit. The overall vector representation of the session is made by normalizing the weighted sum of all the pages' embedding vectors. Newer pages are given more weight.

Session-based representation improved classification performance through age-based weighting. The efficacy of this methodology is assessed by comparing it with the no-weighting strategy.

3.3 Weighted Average-Based Vectorization Module

This module creates a numerical vector for each pageID using a chosen embedding technique. The weighted average vector components consist of the page

weights within the session, and describe the session as a whole. This method makes it easier to see how pages are related in session-based clickstream data and to show sessions more meaningfully using numbers.

3.4 Classification Module

This module employs four distinct classification algorithms, which are essential in numerical data analysis and machine learning (ML) activities. They are designed to assign labels to data items based on predetermined classes. The considered classification algorithms are Gaussian Naive Bayes (GNB), Random Forest (RF), Logistic Regression (LR), and LSTM.

The GNB is a probabilistic classification model predicated on the assumption that the variables are independent of each other. Based on the assumption that the GNB classes have Gaussian (normal) distributions, Bayes' theorem is used to determine the likelihood of a data point belonging to a certain class. This rapid and computationally efficient model is particularly successful for short datasets and high-dimensional data.

RF is an ensemble learning technique based on decision trees. It aggregates the predictions by constructing numerous individual trees and identifying the most commonly observed class as the final prediction. RF can successfully predict complex relationships between data and uses sampling and random variable selection methods, like bagging, to keep overfitting to a minimum. This feature enables RF to deliver high accuracy and operate efficiently on extensive data sets.

LR is a statistical classification technique that effectively models linearly separable data. The model categorizes the data into two or more classes by computing a probability function based on the input variables. It standardizes the output values between 0 and 1 using the sigmoid activation function and generates a class prediction based on a specified threshold value. It is predominantly favored due to its simplicity, interpretability, and computing efficiency.

Finally, LSTM is a model for categorizing the output of Recurrent Neural Networks (RNN) and is very good at showing how things change over time. We engineered it to overcome the difficulties faced by conventional RNNs in acquiring long-term dependencies. LSTM can retain significant information over extended periods and discard extraneous information due to its gating processes. These properties render LSTM very proficient in time series analysis, sequential event forecasting, and text categorization jobs.

Each of the four methods possesses distinct advantages and disadvantages, necessitating selection based on the specific problem type. GNB is fast and needs little data. RF, on the other hand, is more accurate but may require high computing power for large datasets. Although LR is distinguished by its interpretability, it is constrained in its ability to describe nonlinear complicated interactions. LSTM, despite being one of the most effective techniques for sequential data, necessitates greater data and computational resources compared to alternative models.

3.5 Classification Results Evaluator Module

This module evaluates the efficacy of the employed categorization algorithms which determine the model's capability to accurately categorize data items. This paper employs four distinct evaluation metrics: Accuracy, Precision, Recall, and F1 Score.

Accuracy is a fundamental metric that assesses the precision of the model's predictions. It is defined as the ratio of accurately identified data points to the total number of data points. Nonetheless, it may not be adequate for skewed data sets.

Precision denotes the ratio of instances that the model identifies as positive. It is beneficial in situations where erroneous positive predictions are significant, particularly where the reduction of false positives is essential. Precision is a critical parameter for models aiming to achieve high-accuracy predictions.

The Recall measures the model's accuracy in identifying true positives. The ratio of genuine positives to the total number of true positives determines recall. This statistic is crucial, particularly in scenarios where incorrect negative predictions incur significant costs, as in essential domains like disease detection.

The F1 Score is a balanced metric combining Precision and Recall. It is defined as the harmonic mean of these two metrics, as shown in Eq. 3:

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3}$$

The F1 Score serves as a robust evaluation metric when it is essential to optimize both Precision and Recall.

The four evaluation metrics discussed above offer a thorough assessment of the efficacy of categorization algorithms. Accuracy assesses overall performance, whereas Precision and Recall elucidate the nature of errors. The F1 Score is the most effective evaluation tool for imbalanced datasets. The determination of the appropriate measure in model selection is contingent upon the study's objectives and the data architecture.

4 Implementation and Evaluation of the Proposed Business Process

This section presents the dataset utilized in the study, along with the execution and assessment of the business process depicted in Fig. 1. The given dataset, data pretreatment steps, embedding methods, and classification algorithms were carefully considered during the business process to observe how well the aging-based weighting mechanism worked at reflecting user behavior. The results of this procedure were examined, the advantages and drawbacks of the suggested method were deliberated, and insights into its applicability in various circumstances were offered.

4.1 Dataset

The dataset included in this study has been meticulously chosen to facilitate a thorough assessment of the proposed methodology. The dataset provides a complete testing ground to see how well the aging-based weighting method shows and groups user behavior. This subsection delineates the size, structure, and fundamental characteristics of the dataset. The dataset is obtained from publicly available sources that provide important information about how users interact and browse the web. Such information makes it easier to evaluate the proposed method.

This study utilized dataset from the mobile application of a coffee chain, available on the Kaggle website [32], This dataset encompasses multiple variables, including 14-day user interactions, and offers a thorough foundation for evaluating user behavior. The dataset comprises 96,200 distinct clients, and their interactions with the application were monitored. A total of 296,100 sessions were documented during the study, creating a robust analytical framework for investigating user interaction patterns.

We established that each session of user exploratory activity may encompass between 1 and 11 page views. This data indicates that user interactions with the platform differ significantly. The users were found to spend a minimum of 1 s and a maximum of 1,349 s on a page. This data indicates that session durations can fluctuate significantly.

The dataset includes comprehensive information regarding the user activities. A total of 43,390,766 events was documented, yielding an average of 8.61 events per session. The maximum number of events documented in a session was established at 11. Furthermore, the dataset includes 46 extensive URLs, offering additional insights into users' interactions on the platform. The average session duration was determined to be 4,976.34 s; this sheds light on the evolution of user activity on the platform over time. Table 1 offers a comprehensive description of the dataset, facilitating a clearer observation of the evolution of user interactions over time.

To improve the reliability and accuracy of the classification results, we chose events that happened at least eight times for the embedding process. We established this criterion to mitigate the risk of brief sequences inadequately capturing significant patterns. By focusing on long interaction sequences, the model aimed to learn from a more stable dataset.

To get a better idea of how the session-based dataset is put together, the event statistics within the session was carefully studied. Consequently, sessions were classified based on the number of events they encompass. Five distinct groups were established: sessions with 1–6 events, those containing 7 events, those with 8 events, those encompassing 8–10 events, and those higher than 10 events. Figure 2 illustrates the distribution of the categories (number of sessions with n pages) within the dataset structure with the session length (number of pages n in a session). This study thus shows how the dataset is organized and what causes differences in session lengths and event frequencies.

Table 1. Dataset Features

	Dataset
Number of Distinct Customers	96.200
Number of Total Sessions	296.100
Number of Large-Scale URLs	46
Unique Event Count	46
Average Number of Events per Session	8,61
Maximum Number of Events per Session	11
Minimum Number of Events per Session	1
Total Number of Events	43.390.766
Time Period for the Dataset	14
Max. and Min. Visiting Page Count Per Session	Min = 1, Max = 11
Max. and Min. Time Spent on a Page	Min = 1 sec, Max = 1349 sec
Average Time Spent in Session	4.976,34 sec

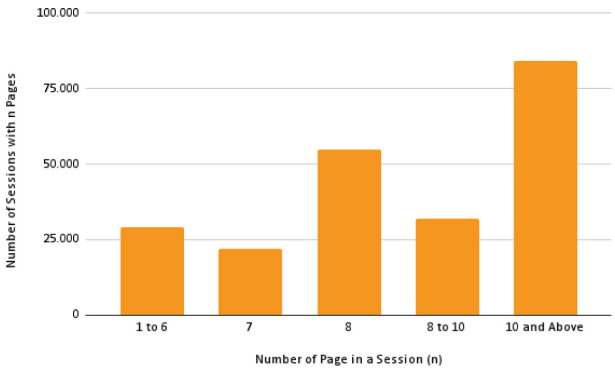

Fig. 2. Number of sessions with event counts for datasets

4.2 Implementation

The proposed method was executed with Python version 3.11.9. The Word2Vec, Node2Vec, and LSTM Autoencoder embedding techniques were implemented using the Gensim (4.2.0) [25], NetworkX [8], and Keras [11] packages, respectively. Apache Spark 3.2.3 [3,16] was employed in data processing and model training to handle large-scale datasets.

The implementation of the proposed approach involves multiple embedding and classification models, each requiring specific hyperparameter settings for optimal performance. Table 2 summarizes the key hyperparameters used in the Word2Vec, Node2Vec, and LSTM Autoencoder embedding techniques, as well as the settings for classification models. These parameters were determined based on prior research and fine-tuning experiments to achieve the best classification accuracy.

Table 2. Hyperparameter Settings for Embedding and Classification Models

Model	Parameter	Value	Description
Word2Vec	Vector Size	64	Dimensionality of word embeddings
	Window Size	5	Context window size
	Min Count	2	Minimum frequency threshold
Node2Vec	Walk Length	10	Number of nodes per walk
	Number of Walks	80	Walks per node
	P/Q Ratio	0.5	Return/In-out parameter
LSTM Autoencoder	Hidden Units	128	Number of LSTM neurons
	Dropout Rate	0.2	Regularization factor
	Batch Size	32	Number of samples per batch
Classification Models	Learning Rate	0.001	Step size for optimizer
	Epochs	20	Number of training iterations
	Optimizer	Adam	Optimization algorithm

We trained the Word2Vec model using the Gensim toolkit to create vector representations of session-based clickstream data within the framework of embedding approaches. The Node2Vec algorithm used graph-based random walks to figure out how the nodes were connected. We trained the node sequences and created vector representations using the Word2Vec model. The LSTM Autoencoder approach generated vector representations that incorporated temporal dependencies by acknowledging the sequential structure of the sessions.

Various ML and deep learning models were employed during the classification phase. We used the Scikit-Learn [14] module to run the GNB, RF, and LR models. Then, we used the results to group the session vectors into different categories. The LSTM-based classifier was developed and trained using the Keras package to analyze consecutive session data and more effectively capture user behavior over time.

We made a unique vector for each session by putting together embedding vectors that matched the PageIDs in that session. A method based on aging was used to give these embedding vectors more weight, which means that recent encounters were given more weight. Lastly, we got the total embedding vector for the session by taking the weighted average of these separate vectors.

This procedure was thoroughly tested to identify session-based user behaviors and model performance by using proposed methods with different classification algorithms.

Our classification task's target variable is a binary label that denotes user engagement (for example, whether or not a session resulted in a particular kind of interaction, like a purchase). The dataset had these labels predefined. To preserve class distribution, the data was stratified into training (80%) and testing (20%) sets.

Regularization strategies like dropout (used in LSTM) and early stopping were used during model training to avoid overfitting. Additionally, a separate holdout test set and stratified 5-fold cross-validation were used to confirm that the model performance was generalizable. Because preprocessing and embedding were rigorously limited to the training set during validation, there was no test set leakage.

4.3 Evaluation

In this section, we test how well the suggested aging-based weighting method works for representing sessions in e-commerce platforms. We tested the performance gains by comparing the aging-based weighting method to the old no-weight strategy using four classification algorithms: GNB, RF, LR, and LSTM. The comparison was made across three embedding methods: Word2Vec, Node2Vec, and LSTM Autoencoder.

4.4 Comparison of Weighting Mechanisms

The no-weight approach assumes uniform importance across all page visits within a session, whereas the aging-based strategy gives higher importance to more recent user actions, capturing the temporal relevance of interactions more realistically.

To go into more depth about the test, the following tables show the classification performance metrics for each embedding method and classifier combination:

- The results for Word2Vec embeddings can be seen in Table 3. It has been shown that aging-based scaling makes a big difference in how well classifiers work, especially GNB (F1: 0.4892 0.8455) and LSTM (F1: 0.7205 0.8763).
- The classification results for Node2Vec are shown in Table 4. It does a great job even without weighing. With aging-based weighting, on the other hand, all models get better than an F1 score of 0.99, showing that adding temporal context is useful.
- The measurements for the LSTM Autoencoder are shown in Table 5. Its performance is already very good, but it gets even better when aging-based weighting is used (e.g., LSTM: 0.9959 0.9999 F1 score).

4.5 Impact of Weighting Strategies on Classification Performance

The observed results can be summarized with the following key findings:

- Aging-based weighting makes classification much better across all models and embeddings, especially in measures like F1 score and Recall that are sensitive to class imbalance.
- No matter how the weights are set, the LSTM classifier always does the best, especially when paired with the LSTM Autoencoder encoding.

Table 3. Classification results using Word2Vec embeddings

Weighting	Metric	GNB	RF	LR	LSTM
No-Weight	F1 Score	0.4892	0.5364	0.5312	0.7205
	Accuracy	0.7038	0.6849	0.6839	0.7908
	Precision	0.9814	0.7461	0.7494	0.9131
	Recall	0.3258	0.4188	0.4114	0.5950
Aging-Based	F1 Score	0.8455	0.8722	0.8215	0.8763
	Accuracy	0.8758	0.8809	0.8586	0.9008
	Precision	0.9223	0.8184	0.9119	0.9587
	Recall	0.7806	0.9335	0.7474	0.8069

Table 4. Classification results using Node2Vec embeddings

Weighting	Metric	GNB	RF	LR	LSTM
No-Weight	F1 Score	0.8869	0.9716	0.9180	0.9776
	Accuracy	0.9074	0.9753	0.9294	0.9804
	Precision	0.9469	0.9724	0.9293	0.9752
	Recall	0.8341	0.9708	0.9069	0.9799
Aging-Based	F1 Score	0.9908	0.9913	0.9913	0.9924
	Accuracy	0.9917	0.9924	0.9924	0.9934
	Precision	0.9972	0.9899	0.9899	0.9887
	Recall	0.9845	0.9927	0.9927	0.9962

- When aging-based weighting and algorithms like RF and LR are used with Word2Vec and Node2Vec embeddings, the results are much better.
- GNB still doesn't work as well as other models because it relies on features not being related to each other and is sensitive to skewed distributions.

4.6 Effect of Embedding Methods

Evaluating different embedding approaches yields further insight into the modeling of user behavior:

- It turns out that LSTM Autoencoder is the best embedding method, working almost perfectly, especially when used with aging-based weighting (Table 5).
- Because it can keep structure relationships when users navigate (Table 4), Node2Vec gives good results, especially when used with ensemble-based classifiers.
- Word2Vec is widely used, but it doesn't work as well as it could because it can't model time or structure (Table 3).

Table 5. Classification results using LSTM Autoencoder embeddings

Weighting	Metric	GNB	RF	LR	LSTM
No-Weight	F1 Score	0.9952	0.9930	0.9940	0.9959
	Accuracy	0.9958	0.9939	0.9948	0.9964
	Precision	0.9907	0.9876	0.9927	0.9929
	Recall	0.9999	0.9985	0.9954	0.9989
Aging-Based	F1 Score	0.9976	0.9984	0.9992	0.9999
	Accuracy	0.9979	0.9985	0.9993	0.9996
	Precision	0.9955	0.9972	0.9987	0.9998
	Recall	0.9998	0.9996	0.9997	0.9999

Given that LSTM architectures naturally capture temporal patterns, the slight improvements in the LSTM Autoencoder model that were seen when aging-based weighting was added are in line with expectations. Even so, modest gains imply that weighting to emphasize time still adds complementary value.

4.7 Final Remarks on Aging-Based Weighting

The study shows that weighting based on age gives a more accurate and time-aware picture of user sessions. This is especially important in dynamic areas like e-commerce. The aging-based method makes the model more sensitive and flexible by giving more weight to recent events in user sessions. This leads to better personalization, more accurate session classification, and a better understanding of how users behave.

Our main goal was to improve classification accuracy, but we know that explainability is important for real-world use. Attention mechanism values can be added to LSTM models to make the temporal effect easier to understand. Ensemble models, like Random Forest, offer feature importance rankings. In the future, work will be done to incorporate these kinds of techniques to make things easier to understand.

5 Conclusion and Future Work

This paper proposes a novel strategy that combines an aging-based weighting mechanism with data embedding techniques to analyze user browsing activity on e-commerce platforms. The proposed method using embedding techniques such as Word2Vec, Node2Vec, and LSTM Autoencoder was observed to enhance the performance of classification algorithms. Using ML algorithms like GNB, RF, LR, and LSTM in experiments shows that the aging-based weighting method accurately describes user sessions and greatly improves classification performance.

Experimental results show that the aging-based weighting outperforms the no-weighting approach in accuracy, precision, recall, and F1 score, among other things. The LSTM-based model demonstrated superior performance due to its capacity to assess sequential user sessions. The RF and LR models performed well on some datasets, however, LSTM was observed to show the highest performance in predicting complex temporal connections.

The results of this study indicate that the aging-based weighting method enhances comprehension of users' evolving interaction patterns over time. It also notably enhanced categorization performance in scenarios when page content fluctuates and users regularly revisit specific pages. We also observed that this approach generates more significant representation vectors and enhances model correctness in web sessions with infrequent occurrences.

In the future, we intend to extend our work to cover hybrid methodologies underpinned by supervised learning techniques. This approach enables the development of proactive recommendation systems through user segmentation and the prediction of future user behaviors. The integration of the aging-based weighting mechanism with some other weighting algorithms to enhance tailored user experiences is a significant field for future research.

The proposed aging-based weighting framework can be applied to other domains that involve sequential interaction data, such as social media content engagement patterns, student learning activities on educational platforms, or patient visit logs in healthcare, despite the fact that this study is based on e-commerce data. This study offers a fundamental approach for time-sensitive behavior modeling, which can be useful in each of these domains.

As a result, using an aging-based weighting mechanism along with data embedding methods and ML can accurately model user navigation in e-commerce platforms. This approach establishes a robust basis for subsequent study and applications in e-commerce analytics.

Acknowledgments. This work was supported by Business Finland within the EUREKA CELTIC-NEXT project CISSAN (www.celticnext.eu). The authors would like to thank Hepsiburada R&D Center, Fibabanka R&D Center, the University of Jyvskyl and Yildiz Technical University management for their support.

References

1. Aktas, M.S., Fox, G.C., Pierce, M.: Fault tolerant high performance information services for dynamic collections of grid and web services. Futur. Gener. Comput. Syst. **23**(3), 317–337 (2007)
2. Bharathi, A.V., Rao, J.M., Tripathy, A.K.: Click stream analysis in e-commerce websites-a framework. In: 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), pp. 1–5 (2018). https://api.semanticscholar.org/CorpusID:133605302
3. Caswell, T.A., et al.: matplotlib/matplotlib: Rel: v3. 5.0. Zenodo (2021). https://doi.org/10.5281/zenodo.5706396

4. Churc, K.W.: Word2Vec. Nat. Lang. Eng. **23**(1), 155–162 (2017). https://doi.org/10.1017/S1351324916000334
5. Diaz, M., Johnson, I., Lazar, A., Piper, A.M., Gergle, D.: Addressing age-related bias in sentiment analysis. In: Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, pp. 1–14. CHI 2018, Association for Computing Machinery, New York, NY, USA (2018). https://doi.org/10.1145/3173574.3173986
6. Goyal, P., Ferrara, E.: Graph embedding techniques, applications, and performance: a survey. Knowl. Based Syst. **151**, 78–94 (2018)
7. 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, pp. 855–864 (2016)
8. Hagberg, A., Swart, P., Chult, D.: Exploring network structure, dynamics, and function using networkX (2008). https://doi.org/10.25080/TCWV9851
9. Hamilton, W.L., Ying, R., Leskovec, J.: Representation learning on graphs: methods and applications. IEEE Data Eng. Bull. **40**, 52–74 (2017). https://api.semanticscholar.org/CorpusID:3215337
10. Kang, W.C., McAuley, J.: Self-attentive sequential recommendation. In: 2018 IEEE International Conference on Data Mining (ICDM), pp. 197–206 (2018). https://api.semanticscholar.org/CorpusID:52127932
11. Ketkar, N.: Introduction to Keras, pp. 97–111. Apress, Berkeley, CA (2017). https://doi.org/10.1007/978-1-4842-2766-4_7
12. Khan, R., Gubanov, M.N.: Towards tabular embeddings, training the relational models. In: 2020 IEEE International Conference on Big Data (Big Data), pp. 5724–5726 (2020). https://api.semanticscholar.org/CorpusID:232373210
13. Koehn, D., Lessmann, S., Schaal, M.: Predicting online shopping behaviour from clickstream data using deep learning. Expert Syst. Appl. **150**, 113342 (2020). https://api.semanticscholar.org/CorpusID:214300556
14. Kramer, O.: Scikit-Learn, pp. 45–53. Springer International Publishing, Cham (2016). https://doi.org/10.1007/978-3-319-33383-0_5
15. Li, Z., Jiang, R., Aarabi, P.: Continuous face aging via self-estimated residual age embedding. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 15003–15012 (2021). https://api.semanticscholar.org/CorpusID:233481240
16. Van der Maaten, L., Hinton, G.: Visualizing data using T-SNE. J. Mach. Learn. Res. **9**(11), 2579–2605 (2008)
17. Mahdavi, S., Khoshraftar, S., An, A.: dynnode2Vec: scalable dynamic network embedding. In: 2018 IEEE International Conference on Big Data (Big Data), pp. 3762–3765 (2018). https://api.semanticscholar.org/CorpusID:54448357
18. Matsui, A., Ferrara, E.: Extracting fast and slow: user-action embedding with inter-temporal information. ArXiv abs/2206.09535 (2022). https://api.semanticscholar.org/CorpusID:249888948
19. Nguyen, H.D., Tran, K.P., Thomassey, S., Hamad, M.: Forecasting and anomaly detection approaches using LSTM and LSTM autoencoder techniques with the applications in supply chain management. Int. J. Inf. Manage. **57**, 102282 (2021)
20. 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)
21. Olmezogullari, E., Aktas, M.S.: Pattern2Vec: representation of clickstream data sequences for learning user navigational behavior. Concurr. Comput. Pract. Exp. **34**(9), e6546 (2022)

22. Padigela, P.K., Suguna, R.: A survey on analysis of user behavior on digital market by mining clickstream data (2020). https://api.semanticscholar.org/CorpusID:215831952
23. Rahmani, M., Caverlee, J., Wang, F.: Incorporating time in sequential recommendation models. In: Proceedings of the 17th ACM Conference on Recommender Systems, pp. 784–790. RecSys 2023, Association for Computing Machinery, New York, NY, USA (2023). https://doi.org/10.1145/3604915.3608830
24. Ramirez, H., et al.: Determining aging trajectories though hyperbolic embeddings of meg brain networks. Alzheimer's & Dementia **20** (2024). https://api.semanticscholar.org/CorpusID:275432018
25. Řehůřek, R., Sojka, P.: Software framework for topic modelling with large corpora. In: Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks, pp. 45–50. ELRA, Valletta, Malta (2010). http://is.muni.cz/publication/884893/en
26. Rong, X.: word2vec parameter learning explained. ArXiv abs/1411.2738 (2014). https://api.semanticscholar.org/CorpusID:2154019
27. Sahinoglu, M., Incki, K., Aktas, M.S.: Mobile application verification: a systematic mapping study. In: Gervasi, O., Murgante, B., Misra, S., Gavrilova, M.L., Rocha, A., Torre, C., Taniar, D., Apduhan, B.O. (eds.) ICCSA 2015. LNCS, vol. 9159, pp. 147–163. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-21413-9_11
28. Seo, M., Lee, K.Y.: A graph embedding technique for weighted graphs based on LSTM autoencoders. J. Inf. Process. Syst. **16**, 1407–1423 (2020). https://api.semanticscholar.org/CorpusID:234352267
29. Sivakumar, S., Sarvani, L., Videla, Itnal, S., Kumar, R., Haritha, D.: Review on word2vec word embedding neural net. In: 2020 International Conference on Smart Electronics and Communication (ICOSEC), pp. 282–290 (2020). https://api.semanticscholar.org/CorpusID:222222773
30. Skianis, K., Giannopoulos, A., Gkonis, P., Trakadas, P.: Data aging matters: federated learning-based consumption prediction in smart homes via age-based model weighting. Electronics **12**(14) (2023). https://doi.org/10.3390/electronics12143054, https://www.mdpi.com/2079-9292/12/14/3054
31. Taheri, A.: Learning graph representations with recurrent neural network autoencoders (2018). https://api.semanticscholar.org/CorpusID:52235675
32. Tasgetiren, N.: Clickstream of a e-commerce dataset. https://www.kaggle.com/code/nailtasgetiren/clickstream-of-a-e-commerce-dataset (2024). Accessed 7 Mar 2025
33. Tufek, A., Gurbuz, A., Aktas, M.S.: Provenance collection platform for the weather research and forecasting model. In: 2018 14th International Conference on Semantics, Knowledge and Grids (SKG), pp. 17–24 (2018)
34. Wei, J., Shen, Z., Sundaresan, N., Ma, K.L.: Visual cluster exploration of web clickstream data. In: 2012 IEEE Conference on Visual Analytics Science and Technology (VAST), pp. 3–12 (2012). https://api.semanticscholar.org/CorpusID:552440
35. Zhang, Y., Xiong, Y., Li, D., Shan, C., Ren, K., Zhu, Y.: Cope: Modeling continuous propagation and evolution on interaction graph. In: Proceedings of the 30th ACM International Conference on Information & Knowledge Management (2021). https://api.semanticscholar.org/CorpusID:240230819
36. Zhang, Z., Cui, P., Zhu, W.: Deep learning on graphs: a survey. IEEE Trans. Knowl. Data Eng. **34**(01), 249–270 (2022)

37. Zhao, M., Masino, A.J., Yang, C.C.: A framework for developing and evaluating word embeddings of drug-named entity. In: Workshop on Biomedical Natural Language Processing (2018). https://api.semanticscholar.org/CorpusID:51877450
38. Zhou, D., Niu, S., Chen, S.: Efficient graph computation for Node2Vec. ArXiv **abs/1805.00280** (2018). https://api.semanticscholar.org/CorpusID:25752861