Temporal Decay Functions for Clickstream Session Representation: A Comparative Analysis of Linear and Exponential Weighting Strategies

Deniz Guzelcay[1(✉)], Osman Hayek[1], Ali Haydar Sariaydın[1], Halil İbrahim Simsek[2], Onur Cilgin[2], and Mehmet S. Aktas[1]

[1] Yildiz Technical University, Istanbul, Turkey
{deniz.guzelcay,osman.hayek,haydar.sariaydin}@std.yildiz.edu.tr, aktas@yildiz.edu.tr
[2] Trendyol R and D Center, Istanbul, Turkey
{halilibrahim.simsek,onur.cilgin}@trendyol.com

Abstract. Clickstream data from e-commerce platforms offers valuable insights into how users browse. Applications such as user analytics, recommendation systems, and behavioral segmentation depend on modeling these sequential interactions. However, many existing session representation techniques use straightforward averaging to produce session embeddings. These tactics ignore the timing of interactions within a session. In fact, events that occur near the end of a session often provide stronger indications of the user's immediate intent. This implies that the creation of useful session representations may depend on time information. This work examines the impact of temporal decay functions on the development of session embeddings for clickstream analytics. The study examines the effects of various decay strategies on behavioral clustering performance. We propose a session modeling framework that combines explicit temporal weighting techniques with transformer-based event embeddings to address these issues. Linear and exponential temporal decay functions are studied to understand how different weighting strategies affect clustering quality. Experiments on real-world clickstream datasets show that session representations with temporal weighting generally outperform traditional averaging methods. In particular, exponential temporal decay offers the best clustering performance across different embedding models. These results highlight how important it is to consider time in session representation learning. They also show that concentrating on recent interactions greatly improves behavioral clustering in clickstream analytics.

Keywords: clickstream analytics · session representation · user behavior modeling · temporal decay functions · behavioral clustering

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2027
O. Gervasi et al. (Eds.): ICCSA 2026 Workshops, LNCS 16762, pp. 599–618, 2027.
https://doi.org/10.1007/978-3-032-30524-4_37

1 Introduction

The rapid expansion of e-commerce platforms has led to the generation of massive volumes of user interaction data, commonly referred to as clickstream data. These data consist of sequential records of user actions such as product views, searches, and cart operations performed during a browsing session. Understanding behavioral patterns embedded within these sequences is essential for applications including recommendation systems, user segmentation, purchase prediction, and personalization. Consequently, representing user sessions in a compact numerical form that captures meaningful behavioral signals has become a fundamental problem in clickstream analytics and web usage mining.

A wide range of approaches have been proposed in the literature to model sequential behavioral data obtained from user interactions. Early studies relied on frequency-based representations and term-weighting techniques originally developed in information retrieval to capture the relative importance of elements within a dataset [1]. With the advancement of representation learning, embedding techniques have become widely used to map discrete events into continuous vector spaces that preserve contextual relationships. Recent developments in transformer-based architectures have further improved the ability to model sequential dependencies and contextual information in event sequences. In parallel, several studies have explored embedding techniques as an effective tool for capturing semantic relationships between behavioral events in large-scale datasets [2].

Previous studies conducted by our research group have extensively investigated embedding-based approaches for modeling user navigational behavior in clickstream environments. In particular, graph-based embedding techniques and sequence-based representation learning methods have been applied to convert user browsing events into numerical vectors suitable for downstream analytics tasks such as clustering, prediction, and intelligent prefetching [3,4]. These studies demonstrated that embedding-based representations can effectively capture structural relationships between user interactions and provide meaningful representations of browsing behavior in e-commerce and digital service platforms. Furthermore, several works have explored predictive modeling and sequential learning approaches for understanding user actions and navigation patterns in digital platforms [5–7]. In addition, recent studies have investigated the integration of frequency-based and time-sensitive weighting mechanisms within embedding pipelines to enhance the modeling of user behavioral patterns and improve session-based behavioral analysis [8].

Despite these advances, important limitations remain in existing session representation approaches. In many embedding-based models, session representations are constructed using simple averaging or uniform aggregation of event embeddings. Such approaches assume that all events within a session contribute equally to the resulting representation. However, in real-world clickstream environments, user interactions often follow a dynamic pattern in which actions occurring later in the session may reflect stronger behavioral signals regarding the user's immediate intent. Ignoring such temporal dynamics may therefore

lead to diluted session representations and reduce the discriminative power of behavioral clustering models.

Motivated by this limitation, this study investigates how the temporal position of events within a session can be incorporated into the construction of session representations. In particular, the study examines whether assigning different importance levels to events according to their temporal positions can improve the quality of behavioral clustering. Two key research questions guide the investigation. First, we examine whether temporally weighted session representations provide more discriminative clustering structures compared to standard unweighted aggregation approaches. Second, we analyze how different temporal decay functions influence clustering performance in modeling users' navigation behavior within clickstream sessions.

This study creates a modular framework for session modeling that integrates temporal weighting and event embeddings in order to investigate these questions. This framework uses transformer-based models to represent each event in a user session as a numerical vector. Each event's contribution is then modified based on its position during the session. Plain aggregation, linear weighting, and exponential weighting, which give more weight to more recent interactions, are the three weighting approaches taken into consideration.

Event embedding generation, temporal weighting, session vector construction, and clustering comprise the four stages of the methodology's modular workflow. Four transformer-based models, BERT4Rec, SASRec, BERT-base-uncased, and ALBERT-base-v2, are used to create event embeddings in order to compare how various models depict user behavior. The aforementioned weighting techniques are then used to aggregate these event-level embeddings into session-level vectors. The clustering analysis uses the resulting fixed-length session representations.

Clickstream datasets gathered from e-commerce platforms are used for clustering experiments to assess the suggested representation framework. K-Means and Bisecting K-Means are used to analyze session vectors produced from various combinations of embedding models and weighting techniques. Standard internal validation metrics are used to evaluate cluster quality, with special focus on the Silhouette score. The right number of clusters is chosen using the Elbow method. With this configuration, the separability and consistency of behavioral clusters can be investigated in relation to temporal weighting.

The study's findings yield a useful method for generating session representations for clickstream analytics. Session representations that better represent user behavior are produced by combining transformer-based event embeddings with time-aware weighting. In large-scale e-commerce environments, these representations can facilitate behavioral segmentation and provide a practical means of modeling short-term user intent.

Although the experimental setup is built around clickstream data, the underlying problem is more general: a session can be viewed as a discrete realization of a time-dependent process in which the informational value of past states fades as new ones are observed. Similar formulations arise in sensor streams, event logs,

and short-horizon state descriptors used in data-driven modeling of dynamical systems, where the choice of decay function plays a role conceptually close to that of a memory kernel.

This is how the rest of the paper is structured. The literature on clickstream analytics, embedding-based user behavior modeling techniques, and session representation techniques is reviewed in Sect. 2. The suggested approach, which includes session vectorization, temporal weighting, event embedding creation, and the clustering framework, is presented in Sect. 3. The implementation and experimental setup, the evaluation procedure, and the clustering results obtained with various embedding models and weighting strategies are all described in Sect. 4. The paper is finally concluded in Sect. 5, which also explores potential future research directions.

2 Literature Review

In our earlier work, we explored several embedding-based approaches for modeling user navigational behavior in clickstream data. We first focused on graph-based methods such as Word2Vec, Node2Vec, and Struct2Vec to represent browsing patterns and group similar behavioral sequences [3]. These studies showed that embedding models can transform high-dimensional clickstream data into compact vector representations while preserving structural relationships among user interactions. Later, we extended this line of work to predictive modeling, where embedding-based representations were combined with deep learning methods to model customers' digital footprints and anticipate future actions on digital platforms [6]. More recent studies in our group incorporated temporal aspects into session representations. This included aging-based weighting methods that gave greater emphasis to recent interactions in session modeling [8]. We also examined other ways of representing user intent in clickstream sessions, including recency-weighted embeddings and sequential neural representations [7]. In addition, embedding-based navigation models were applied to predictive prefetching in data-intensive systems [4]. When combined, these studies imply that complex behavioral patterns in extensive web navigation data can be captured by embedding-based representations. Building on this work, the current study combines transformer-based embeddings with explicit temporal weighting strategies to investigate how various temporal decay functions affect the efficacy of session-based embedding representations in clickstream analytics.

Modeling user behavior from clickstream data has been a recurring topic in web usage mining and recommender systems research. Early work largely relied on statistical and frequency-based methods to represent user sessions. Approaches such as term weighting, association rule mining, and sequential pattern analysis were commonly applied to identify navigational patterns. The TF-IDF scheme proposed by Salton and Buckley [1] became one of the more widely adopted techniques for estimating the relative importance of elements in both document collections and behavior-based datasets. Similarly, early web usage mining studies explored Markov models and sequential pattern mining to characterize how users navigate through websites [9]. While these methods provided

useful descriptive insights, their reliance on sparse representations made it difficult to capture more nuanced semantic relationships between events.

Representation learning offered a different way to model user interaction data. Methods such as Word2Vec [10] introduced the idea of representing discrete entities as dense vectors, allowing semantic similarity to be preserved in a continuous space. This framework was subsequently extended to graph and network-structured data. Approaches such as DeepWalk [11] and Node2Vec [12] proposed scalable techniques for learning node representations directly from network topology. These methods were gradually adopted in recommender systems and behavioral analytics, where they proved useful for capturing relationships among users, items, and interaction events.

Embedding techniques have also been applied to session modeling in clickstream contexts. Actions such as product views, searches, and purchases can be mapped to continuous vectors, making it possible to surface latent behavioral patterns. Prior work suggests that such representations can support tasks including clustering, recommendation, and purchase prediction, largely by encoding similarity relationships across interaction sequences [13]. At the same time, many session representation methods still rely on simple aggregation schemes such as mean pooling or uniform averaging. In practice, this means that all events are treated as if they contribute equally to the final session representation.

At the same time, research on session-based recommendation systems introduced sequential neural architectures. These architectures model the dependencies among user interactions. One of the earliest neural approaches is GRU4Rec [14]. It uses recurrent neural networks to predict the next interaction in a session. Later studies suggested other architectures, including convolutional models [15] and hybrid neural methods, to better capture sequential dependencies. These models showed that taking the order of interactions into account greatly improves the way short-term user preferences are represented compared to static session representations.

More recently, transformer-based architectures have greatly improved sequential user behavior modeling. The Transformer model proposed by Vaswani et al. [16] allows for effective modeling of long-range dependencies using self-attention mechanisms. Several transformer-based models have been developed for sequential recommendation tasks, including SASRec [17], BERT4Rec [18], and related transformer-based frameworks for session-based recommendation [19]. These models create contextualized representations of interaction sequences by modeling dependencies across the entire session together. Although transformer architectures have performed well in recommendation tasks, their use in behavioral clustering and session segmentation is still quite limited.

Another important research direction concerns weighting mechanisms for modeling the importance of events within behavioral data. Traditional information retrieval methods like TF-IDF focus on informative elements based on frequency statistics [1]. In recommender systems, a range of importance-weighting strategies has been introduced to give greater emphasis to certain interactions within user sessions. Among these, attention-based mechanisms assign different

levels of contribution to individual events in the formation of user representations [20]. Most existing weighting strategies are mainly made for prediction or recommendation tasks. They are not designed for clustering-based behavioral analysis.

Temporal dynamics are also important in modeling user behavior in session-based environments. Earlier studies indicate that user preferences are not static over time and that recent interactions may carry more informative signals for short-term user intent. For example, Koren [21] incorporated temporal dynamics into collaborative filtering models to improve recommendation accuracy. Similarly, Campos et al. [22] provided a comprehensive overview of time-aware recommender systems, highlighting the importance of temporal information in modeling user interactions.

Recent research has also explored how contextual embeddings can be combined with sequential modeling frameworks. For example, Quadrana et al. [23] introduced hierarchical recurrent models that capture both session-level and user-level dynamics in recommendation systems. In a similar direction, industrial frameworks such as Transformers4Rec illustrate how transformer architectures can be used in real-world recommendation settings by bringing together contextual embeddings and sequential interaction modeling [24].

Despite these advances, several limitations remain in existing session representation approaches. Many embedding-based methods construct session representations using uniform aggregation strategies that ignore the temporal importance of interactions. Moreover, while time-aware recommender systems incorporate temporal signals for prediction tasks, relatively few studies investigate how temporal weighting influences behavioral clustering or session segmentation. To address these limitations, this study proposes a temporally sensitive session representation framework that integrates transformer-based event embeddings with explicit temporal weighting strategies. By combining contextual embedding models with analytically defined temporal decay functions, the proposed approach aims to construct more discriminative session representations that improve clustering-based behavioral segmentation in clickstream analytics.

3 Methodology

In this study, user sessions obtained from e-commerce platforms are analyzed using a clustering-based approach on clickstream data. Clickstream data refer to a time-ordered record of successive interactions (events) performed by a user during a single visit, and they contain important cues reflecting the user's intentions. The primary objective of this work is to generate vector representations (session embeddings) for user sessions using different methods and to systematically evaluate the impact of these representations on clustering performance. In particular, the effect of assigning different weights (linear, and exponential) to events within a session based on their temporal positions on the quality of cluster separation is investigated in detail.

The proposed method is built upon a modular structure consisting of event embedding generation, temporal weighting, session vectorization, and clustering

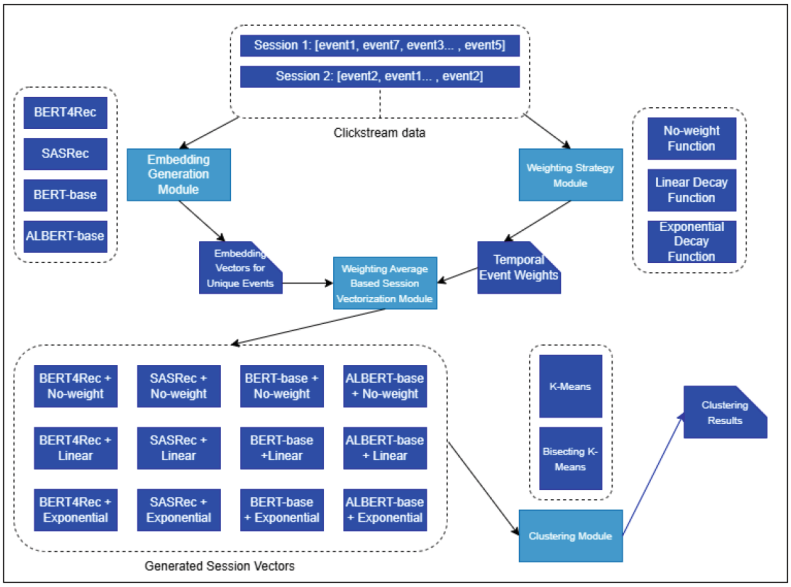

Fig. 1. Overview of the proposed methodology and its modular workflow

steps, starting from raw clickstream data. This modular structure and the data flow between components are illustrated in the flow diagram summarizing the general methodological framework of the study (Fig. 1). The flow diagram holistically presents the operations applied at each stage and how different method combinations are generated.

In the datasets employed, each observation represents a single session belonging to a user. Each session is defined as a sequence of consecutive events performed by the user during a visit, and these sequences are stored in chronological order. During the preprocessing stage, incomplete or erroneous records were removed from the dataset; subsequently, the event sequences for each session were parsed, and empty or invalid values were cleaned. As a result of these procedures, each user session was transformed into a cleaned list of events suitable for analysis. In this study, we use the following notation for a user session:

$$S = [e_1, e_2, \ldots, e_L] \tag{1}$$

where e_i denotes the i-th event within the session, and L denotes the length of the session (i.e., the total number of events). This notation serves as the fundamental representation for both event-level embedding generation and session-level vectorization processes. In the following subsections, each module in the flow diagram is discussed in detail. The components of the proposed method are explained.

Embedding Generation Module: The goal of this stage is to represent each user interaction (event) as a numerical vector, making it suitable for further analysis. Clickstream data has a categorical and sequential structure, so it is important to embed this information into a continuous vector space. In this study, four different embedding approaches are used: BERT4Rec, SASRec, BERT-base-uncased, and ALBERT-base-v2. These approaches are compared based on their ability to represent the data.

BERT4Rec and SASRec are Transformer-based sequential recommender models specifically designed to model sequences of user interactions. Through the self-attention mechanism, both approaches are capable of learning long-range dependencies among events within a session and generating contextualized representations for each interaction. BERT4Rec employs a bidirectional Transformer architecture and a masked item prediction strategy, treating the session sequence holistically and representing each event by considering both preceding and succeeding interactions. In contrast, SASRec utilizes a unidirectional self-attention mechanism that explicitly accounts for temporal order, effectively modeling sequential dependencies based on the user's past behavior. In this study, using these transformer-model based embedding approaches, 64-dimensional vector representations were obtained for each event. Consequently, representations that reflect the sequential and contextual patterns in clickstream data and detailedly encode user behaviors were produced.

On the other hand, the BERT-base-uncased and ALBERT-base-v2 models are contextual language models built upon the Transformer architecture and pre-trained on large-scale text corpora. Since these models are designed for natural language inputs, each clickstream event was first converted into natural language sentences to reflect its semantic structure. For instance, an event such as "add_to_cart" was represented by the phrase "add product to shopping cart." When these sentences are provided as input to the model, hidden state vectors are generated for each token. In this study, the vector corresponding to the special [CLS] token, which is prepended to the sentence and trained to represent the entire sequence, was utilized as the embedding representation of the corresponding event. Since the [CLS] token aggregates the contextual information derived from the entire sentence into a single vector, it provides a summary that holistically represents the semantic content of the event. In this way, events defined through natural language are transformed into context-aware and semantically rich vector representations.

Weighting Strategy Module: The objective of this module is to assign varying degrees of importance to events within a user session by considering their temporal positions, thereby ensuring that session representations better reflect the user's current intent. In clickstream data, it is widely accepted that interactions occurring toward the end of a session more strongly represent the user's short-term interest. Therefore, weighting events based on their order within the session can produce more discriminative session vectors compared to plain averaging approaches. In this study, two distinct temporal weighting strategies, each

representing a different assumption, are evaluated: linear decay, and exponential decay. The weighting process depends only on the timing of the events, not on their content. A user session is labeled as S, with L indicating the total number of events. This sequence goes into the weighting module. The result is a set of normalized weights that matches the session length L. Each weight corresponds to a specific event in the sequence.

Plain (No-weight) Weighting: In the plain weighting approach, it is assumed that all events within a session possess equal importance. This approach serves as a baseline method where temporal information is completely disregarded. In this case, the weight value for each event is defined as:

$$w_i = \frac{1}{L}, \quad i = 1, 2, \ldots, L \tag{2}$$

In this approach, the sum of all weights equals one:

$$\sum_{i=1}^{L} w_i = 1 \tag{3}$$

In the implementation, this strategy is applied by generating unit weights for all events and dividing them by the session length.

Linear Weighting: In the linear weighting approach, weights are assigned to events in a linearly increasing manner according to their temporal order within the session. This approach relies on the assumption that events occurring toward the end of the session are more important, but this increase occurs in a controlled and gradual manner. First, the normalized position of each event within the session is defined as follows:

$$p_i = \frac{i-1}{L-1}, \quad i = 1, 2, \ldots, L \tag{4}$$

Here, p_i takes values in the range $[0, 1]$ and increases monotonically from the beginning to the end of the session. The linear weight value for each event is then calculated as:

$$\tilde{w}_i = a + b \cdot p_i \tag{5}$$

Here, a and b represent hyperparameters that control the initial level and the slope of the weight distribution, respectively. The resulting weights are then normalized to make sure their total adds up to one. This normalization step is important for keeping the scale the same across sessions of different lengths and for maintaining the consistency of the generated session vectors' magnitudes:

$$w_i = \frac{\tilde{w}_i}{\sum_{j=1}^{L} \tilde{w}_j} \tag{6}$$

Using this method, events that happen toward the end of the session get more weight. In contrast, the impact of early-stage events is lessened.

Exponential Weighting: The exponential weighting approach is a more aggressive strategy. In this method, the importance of time increases exponentially instead of linearly. In this method, the representational power of events near the end of the session is significantly amplified, making the user's most recent interactions dominant. First, normalized temporal positions are defined as in the linear method:

$$p_i = \frac{i-1}{L-1} \tag{7}$$

Subsequently, the exponential weight value for each event is calculated as follows:

$$\tilde{w}_i = \exp(\lambda \cdot p_i) \tag{8}$$

where λ is a positive hyperparameter that controls how rapidly the weights increase. Higher λ values make the influence of events near the end of the session even more dominant. The obtained exponential weights are normalized similarly to the linear method:

$$w_i = \frac{\tilde{w}_i}{\sum_{j=1}^{L} \tilde{w}_j} \tag{9}$$

Thanks to this normalization, the sum of the weights is equalized to one, ensuring comparability across sessions of different lengths.

The output of the weighting module is a weight vector for each session, having the same length as the number of events L:

$$\mathbf{w} = [w_1, w_2, \ldots, w_L] \tag{10}$$

This weight vector is combined with the event embedding vectors in the subsequent Session Vectorization Module to construct a single vector representation at the session level.

The two schemes differ in a way that matters in practice. Linear weighting is additive: every event keeps a non-negligible share of the final representation regardless of its position. Exponential weighting is multiplicative, with a fixed ratio between consecutive weights, so the tail of the session can dominate the aggregation once λ is large enough.

Session Vectorization Module: The objective of this module is to reduce multiple event embedding vectors contained within a user session into a single fixed-size session representation. Since clustering algorithms operate in fixed-dimensional vector spaces, it is necessary to represent variable-length event sequences with a singular vector. Accordingly, a weighted average-based vectorization process is applied for each session using the event embedding vectors obtained in the previous stages and the temporal weights generated from the weighting module. The final vector representation ($\mathbf{s}$) for a user session is computed using a weighted summation approach. In this process, the normalized coefficient (w_i) derived from the weighting module represents the temporal importance of the corresponding event, while the vector ($\mathbf{v}_i$) obtained from the

embedding module represents its semantic content. The session vector is obtained through the following equation:

$$\mathbf{s} = \sum_{i=1}^{L} w_i \cdot \mathbf{v}_i \tag{11}$$

This operation makes sure that event vectors are first scaled based on their time importance. Then, they are combined into a single vector. As a result, events happening towards the end of the session, which carry more weight, have a stronger influence on the final session representation.

To increase computational efficiency during the clustering stage and to enable a fair comparison between representations obtained from different models, the dimensions of all session vectors were standardized. In this study, the sequential recommendation models (BERT4Rec and SASRec) were set up to create 64-dimensional vectors during training with the RecBole library. As a result, no further dimensionality reduction was applied to the session vectors from these models. In contrast, a different approach was adopted for the pre-trained language models (BERT-base-uncased and ALBERT-base-v2), which naturally produce 768-dimensional vectors due to their architectures. To project these high dimensional vectors into the same dimensional plane (64 dimensions) as the recommendation models and to alleviate the computational burden, Principal Component Analysis (PCA) was applied. Since different weighting strategies (plain, linear, exponential) create distinct distributions in the vector space, the PCA process was performed separately for each strategy. Consequently, regardless of the source model, each user session is represented by a fixed 64-dimensional vector ($\mathbf{s}$), making it ready for the clustering module.

Clustering Module: The final stage of the proposed framework aims to identify groups of user sessions exhibiting similar behavioral patterns in clickstream data. The session vectors generated in the previous stages are used as inputs to clustering algorithms in order to identify groups of users with comparable navigation behaviors. These clustering methods help show how various temporal weighting strategies affect the separability of session representations.

In this study, the centroid-based K-Means algorithm is used as the main clustering method. K-Means divides session vectors into k clusters by repeatedly assigning each session to the closest centroid. It updates cluster centroids until it reaches a stable state. The aim of the algorithm is to form clusters whose members are similar to one another while remaining distinct from other clusters. In addition to K-Means, the study also considers Bisecting K-Means as an alternative clustering method. This approach follows a hierarchical divisive strategy. It begins by treating the entire dataset as a single cluster and then recursively splits it into smaller groups. At each stage, the selected cluster is divided using K-Means with $k = 2$. The process continues until the desired number of clusters is reached. Bisecting K-Means is included because it can produce more balanced cluster structures when behavioral patterns in the data are relatively diverse.

Selecting an appropriate number of clusters is an important part of clustering-based behavioral analysis. To guide this choice, the study uses both the Elbow method and Silhouette analysis. The Silhouette score evaluates clustering quality by considering both within-cluster cohesion and separation between clusters. The Elbow method, in contrast, examines how within-cluster variance changes as the number of clusters increases. In this study, clustering experiments were carried out with different values of k, and the final cluster configurations were determined by considering the results of these two methods together.

4 Implementation and Evaluation

4.1 Datasets

The proposed clustering approach was evaluated on two real-world e-commerce datasets. These datasets differ notably in terms of session volume, event diversity, and sequence characteristics, which makes it possible to examine the behavior of the method under different data conditions. The main statistical properties of both datasets are summarized in Table 1.

The first dataset contains roughly 197,000 user sessions. It includes a relatively small set of event types, with only nine distinct categories. Even so, it stands out for its high level of interaction within each session. As reported in Table 1, the average session length is 47.15 events, which is relatively high. The maximum sequence length is over 15,000, suggesting that some users remain active on the platform for long periods while repeatedly engaging in a limited set of core actions. The second dataset is larger, with over 547,000 sessions, but exhibits a different behavior pattern than Dataset 1. It offers a much richer diversity of interactions with 37 unique event types (e.g., detailed filtering, campaign interactions). In contrast, the average session length (16.56 events) is significantly shorter. This structure suggests that users exhibit more fragmented and goal-oriented browsing behavior characterized by shorter visits.

Table 1. Statistical Characteristics of the Experimental Datasets

Statistical Metric	Dataset 1	Dataset 2
Total Session Count	197,518	1,002,230
Total Event Volume	9,313,563	9,070,291
Number of Unique Event Types	9	37
Average Session Length	47.15	16.56
Maximum Sequence Length	15,221	2,500
Minimum Sequence Length	1	1

4.2 Implementation Details

The proposed framework was implemented in Python 3.10 within a cloud-based environment using Google Colab Pro. To support the computational demands of deep learning training and clustering on high-dimensional data, the experiments were run with a Tesla T4 GPU with 16 GB of VRAM and high-memory CPU resources. The implementation followed a modular design and made use of several open-source libraries suited to sequential data analysis and machine learning. Two separate approaches were used to generate event embeddings. For the sequential recommendation models, SASRec and BERT4Rec, the *RecBole* library (v2.0.1) was used. Both models were trained on the Next Item Prediction task with the same configuration: a hidden size of 64, an inner size of 256, 2 attention heads, and 2 transformer layers. Training was carried out for 5 epochs with a batch size of 2048, using the Adam optimizer and a learning rate of 0.001.

For the pre-trained language models (BERT and ALBERT), *Hugging Face Transformers* library was utilized. The `bert-base-uncased` and `albert-base-v2` models were deployed to extract semantic features. Custom text templates (e.g., "User performed [event] action") were mapped to event tokens, and the hidden state of the [CLS] token was extracted as the event representation. Following the embedding generation, session representations were constructed using the proposed temporal weighting strategies (Plain, Linear, and Exponential), implemented using the *NumPy* and *Pandas* libraries. To standardize the dimensionality across different models—specifically to reduce the 768-dimensional vectors from language models to a comparable 64-dimensional space—Principal Component Analysis (PCA) was applied using the *Scikit-learn* library. The clustering phase was engineered to support both CPU and GPU execution to handle large-scale session data. The *RAPIDS cuML* library was utilized for GPU-accelerated K-Means and DBSCAN algorithms, significantly reducing the computation time for hyperparameter tuning. For the Bisecting K-Means algorithm, the implementation provided by *Scikit-learn* was used on the CPU. The optimal number of clusters (k) was determined dynamically within the range of $k \in [3, 15]$ by evaluating Silhouette, Davies-Bouldin, and Inertia (Elbow) metrics iteratively.

4.3 Evaluation

One of the most important parameters that directly affects the clustering performance of the K-Means and Bisecting K-Means algorithms is the number of clusters (k). In this study, Silhouette analysis, which is widely used in the literature, was employed to determine the optimal number of clusters for both clustering approaches. Within the scope of the experiments, the number of clusters was systematically evaluated in the range of $k = 3$ to $k = 15$, and the results obtained by applying both K-Means and Bisecting K-Means algorithms for each k value were analyzed.

Silhouette analysis evaluates clustering quality by comparing the similarity of each data point to its assigned cluster with its distance to other clusters. A higher

Silhouette score indicates that clusters are both internally coherent and well-separated from each other. In the experiments conducted, the Silhouette scores were analyzed to determine the point where the clustering structure achieves maximum separation and consistency.

The final number of clusters used in the experiments was chosen based on the level of separation indicated by these scores. This provided a more stable basis for selecting an appropriate cluster count for both datasets and helped ensure that the resulting partitions were aligned with the underlying structure of the data.

Table 2. Clustering Performance of SAS4Rec on Dataset 1

Weighting Strategy	K-Means (Sil.)	Bisecting K-Means (Sil.)
Plain	0.475	0.443
Linear	0.517	0.488
Exponential	**0.566**	0.539

Table 3. Clustering Performance of ALBERT on Dataset 1

Weighting Strategy	K-Means (Sil.)	Bisecting K-Means (Sil.)
Plain	0.466	0.430
Linear	0.507	0.469
Exponential	**0.532**	0.513

Table 4. Clustering Performance of BERT (bert-uncased) on Dataset 1

Weighting Strategy	K-Means (Sil.)	Bisecting K-Means (Sil.)
Plain	0.467	0.439
Linear	0.508	0.482
Exponential	**0.547**	0.511

The clustering results for Dataset 1 and Dataset 2 are reported in eight tables, with four tables for each dataset. For both datasets, the analysis includes four embedding models: SASRec, ALBERT, BERT, and BERT4Rec. Session representations were constructed using three temporal weighting strategies (Plain, Linear, and Exponential). Clustering was then carried out with both K-Means and Bisecting K-Means, and cluster quality was evaluated using the Silhouette

Table 5. Clustering Performance of BERT4Rec on Dataset 1

Weighting Strategy	K-Means (Sil.)	Bisecting K-Means (Sil.)
Plain	0.473	0.427
Linear	0.523	0.480
Exponential	**0.557**	0.523

Table 6. Clustering Performance of SAS4Rec on Dataset 2

Weighting Strategy	K-Means (Sil.)	Bisecting K-Means (Sil.)
Plain	0.548	0.543
Linear	0.603	0.566
Exponential	**0.639**	0.605

score. This metric was used as an internal validation measure because it reflects both the compactness of clusters and the degree of separation between them.

Tables 2, 3, 4 and 5 for Dataset 1 and Tables 6, 7, 8 and 9 for Dataset 2 show a consistent pattern across models and datasets. Among the three weighting strategies, Exponential weighting gives the highest Silhouette scores in all experimental settings under both K-Means and Bisecting K-Means. Linear weighting usually performs better than plain averaging, but it does not reach the same level as the exponential scheme. These findings suggest that placing more emphasis on recent events may improve session representations for clustering-based analysis.

Overall, the results suggest that temporal sensitivity plays a meaningful role in session-based clustering. Assigning greater weight to recent interactions through exponential weighting is associated with improved clustering performance, reflected in clusters that tend to be more compact and better separated. The consistency of this pattern across different embedding models, datasets, and clustering algorithms indicates that the proposed weighting scheme is not specific to a particular experimental configuration. These findings suggest that temporal weighting may help capture session-level behavioral patterns more effectively than uniform aggregation.

Average Silhouette scores were calculated for $k = 3$ to $k = 15$ to provide a better assessment of clustering performance across different cluster setups. For each embedding model and dataset, we then calculated relative improvement

Table 7. Clustering Performance of ALBERT on Dataset 2

Weighting Strategy	K-Means (Sil.)	Bisecting K-Means (Sil.)
Plain	0.550	0.566
Linear	0.589	0.571
Exponential	**0.617**	0.582

Table 8. Clustering Performance of BERT (bert-uncased) on Dataset 2

Weighting Strategy	K-Means (Sil.)	Bisecting K-Means (Sil.)
Plain	0.557	0.541
Linear	0.597	0.562
Exponential	**0.630**	0.580

Table 9. Clustering Performance of BERT4Rec on Dataset 2

Weighting Strategy	K-Means (Sil.)	Bisecting K-Means (Sil.)
Plain	0.546	0.527
Linear	0.604	0.549
Exponential	**0.631**	0.607

ratios compared to the Plain (no-weight) baseline. By combining performance over various values of k, this approach lessens the impact of changes linked to individual cluster counts. It also allows us to examine the overall behavior of temporal weighting strategies in a more steady and accurate way. Therefore, the reported improvement ratios show general performance trends rather than results linked to a particular clustering setup.

The average percentage improvements for the sequential embedding models, SAS4Rec and BERT4Rec, are shown in Table 10. The results for the language-model-based embeddings, BERT and ALBERT, are in Table 11. For all the models tested and both datasets, temporally weighted session representations show higher average Silhouette scores than the unweighted baseline. In particular, Exponential weighting achieves the largest improvements in every case. Linear weighting offers more modest but steady gains.

These results show that including temporal position information in session representations improves clustering quality in a consistent way. The continued improvements across various cluster counts suggest that the benefits of temporal weighting are not limited to specific values of k, but instead represent a general increase in cluster separation and compactness. Overall, the results confirm that focusing on temporal dynamics, especially with exponential weighting, leads to stronger and more stable session-based clustering results.

Table 10. Avg. Silhouette Improvement Compared to No-Weight using K-Means (SAS4Rec vs BERT4Rec)

Weighting	SAS4Rec		BERT4Rec	
	Dataset 1 (%)	Dataset 2 (%)	Dataset 1 (%)	Dataset 2 (%)
Linear	9.87%	5.33%	9.59%	5.77%
Exponential	18.88%	15.27%	16.04%	13.90%

Table 11. Avg. Silhouette Improvement - Compared to No-Weight using K-Means (Bert vs Albert)

Weighting	BERT		ALBERT	
	Dataset 1 (%)	Dataset 2 (%)	Dataset 1 (%)	Dataset 2 (%)
Linear	8.51%	8.50%	9.61%	7.58%
Exponential	15.53%	17.26%	15.67%	13.72%

The experimental results offer a basis for addressing the research questions outlined in this study. The first question concerns the effect of temporal weighting on the segmentation performance of sequential embedding models. To examine this, session representations generated using Plain, Linear, and Exponential weighting schemes were compared across all evaluated models. The results indicate that incorporating temporal information is generally associated with improved clustering quality. Temporally weighted representations tended to produce higher Silhouette scores than the unweighted Plain baseline, suggesting that accounting for the positional order of events within a session may contribute to more coherent and better-separated clusters.

The second research question concerns the effect of different temporal decay functions on model performance. Linear and Exponential weighting were evaluated on both datasets under the same experimental conditions. While Linear weighting shows a modest improvement over Plain averaging, Exponential weighting tends to produce the most favorable results across the evaluated models and datasets. The corresponding increase in Silhouette scores suggests that stronger emphasis on recent interactions may improve the representation of short-term user intent. In this respect, the results point to the temporal decay function as an important design choice in session segmentation.

4.4 Discussion and Limitations

The consistent ordering of the three schemes admits a simple reading: linear weighting preserves the influence of early exploratory actions whose link to the user's final intent is often weak, whereas the exponential scheme acts as a soft selection mechanism in which the tail of the session dominates the aggregation. The same ordering across four embedding models, two datasets, and two clustering algorithms suggests that the effect is driven by the weighting mechanism itself rather than by a particular configuration. Three limitations should be noted: clustering quality is judged through internal validation only; the decay rate λ is fixed across sessions rather than tuned per session or learned jointly with the embedding model; and both datasets, although structurally different, come from e-commerce platforms.

5 Conclusion and Future Work

This study explores how temporal decay functions affect the construction of session representations in clickstream analytics. It focuses on whether the position of events within a session can make session embeddings more useful for behavioral clustering. It also examines how different temporal decay strategies influence this process. To address these questions, a modular session modeling framework was proposed that integrates transformer-based event embeddings with explicit temporal weighting mechanisms. The framework has four main components: embedding generation, temporal weighting, session vectorization, and clustering. Event embeddings were created using four transformer-based models, BERT4Rec, SASRec, BERT-base-uncased, and ALBERT-base-v2. Three temporal weighting strategies, plain, linear, and exponential, were tested to examine their effect on session representation quality and clustering performance.

Experiments took place on two real-world clickstream datasets, which included hundreds of thousands of user sessions and varied interaction patterns. We examined the session representations using K-Means and Bisecting K-Means clustering algorithms. We assessed clustering quality with the Silhouette score. The results show that temporally weighted session representations consistently outperform the baseline unweighted aggregation method across all tested embedding models and datasets. Specifically, exponential temporal weighting achieves the best clustering performance in almost all experimental configurations. This demonstrates that placing more importance on recent user interactions significantly improves the separation and compactness of behavioral clusters. These findings confirm that adding temporal decay methods to session-based embeddings improves behavioral clustering performance. More aggressive decay methods, like exponential weighting, capture users' short-term intent more effectively than slower approaches, such as linear decay.

Beyond the specific research questions, this study offers a practical approach to constructing time-sensitive session embeddings in clickstream analytics. The combination of contextual transformer-based embeddings with explicit temporal weighting provides a structured means of representing user behavior in large-scale e-commerce settings. The results suggest that incorporating temporal importance during session aggregation can improve clustering quality across different embedding models and datasets. Together, these findings point to the relevance of temporal dynamics in session-based behavioral modeling.

Several directions remain open. Temporal importance is currently captured through ordinal position alone; richer signals such as elapsed time between events and overall session duration, together with an adaptive or learned decay rate, are natural next steps. The decay-weighted aggregation is also not tied to clickstream data in any essential way and can be applied to event logs in process mining, sensor traces, or short-window state descriptors in data-driven modeling of dynamical systems, where the weighting function plays a role analogous to a memory kernel. The same representations can also support recommendation, anomaly detection, and purchase prediction.

Acknowledgements. This study was supported by the Trendyol Research and Development Center. Generative AI tools, specifically Google Gemini and ChatGPT, were used only for language editing and improving readability. The authors remain fully responsible for the scientific content of the study.

References

1. Salton, G., Buckley, C.: Term-weighting approaches in automatic text retrieval. Inf. Process. Manag. **24**(5), 513–523 (1988)
2. Xiao, S., Liu, Z., Zhang, P., Muennighoff, N., Lian, D., Nie, J.Y.: C-pack: packed resources for general Chinese embeddings, pp. 641–649 (2024)
3. Taşgetiren, N., Kılıç, T., Aktas, M.S.: On the use of graph embedding techniques for clustering user browsing navigational behaviours. Int. J. Web Grid Serv. **20**(4), 482–504 (2024)
4. Buyuktanir, T., Aktas, M.S.: On the use of embedding techniques for modeling user navigational behavior in intelligent prefetching strategies. Concurr. Comput. Pract. Exp. **37**(3), e8356 (2025)
5. Subaşı, Y., Karadayı, Y., Şafak, I., Aktaş, M.S.: Predicting consumer actions in digital banking with time-sensitive user behavior analysis. In: 2024 the 14th International Workshop on Computer Science and Engineering (WCSE 2024), Phuket Island, Thailand, pp. 86–93 (2024)
6. Hakvar, H., et al.: On the use of predictive deep learning approaches in the frequency and uniqueness-based representation of sequential browsing events. In: Computational Science and Its Applications – ICCSA 2024. Lecture Notes in Computer Science, vol. 14814, pp. 83–100. Springer, Cham (2024)
7. Aslan, M., Can, Y.S., Giritli, E.B., Aktas, M.S.: Balancing recency and order in session representations: a temporal–sequential embedding approach for clickstream analytics. In: 2025 IEEE International Conference on Big Data (BigData), pp. 5362–5372. IEEE (2025)
8. Taşgetiren, N., Şafak, I., Aktaş, M.S.: Aging-based weighting for session classification in user behavior analysis. In: Computational Science and Its Applications – ICCSA 2025 Workshops. Lecture Notes in Computer Science, vol. 15886, pp. 16–34. Springer, Cham (2025)
9. Mobasher, B., Cooley, R., Srivastava, J.: Automatic personalization based on web usage mining. Commun. ACM **43**(8), 142–151 (2000)
10. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
11. 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, New York, USA, pp. 701–710 (2014)
12. 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)
13. Barkan, O., Koenigstein, N.: Item2vec: neural item embedding for collaborative filtering. In: 2016 IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP), pp. 1–6. IEEE (2016)
14. Hidasi, B., Karatzoglou, A., Baltrunas, L., Tikk, D.: Session-based recommendations with recurrent neural networks (2015)

15. Tang, J., Wang, K.: Personalized top-n sequential recommendation via convolutional sequence embedding. In: Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, pp. 565–573 (2018)
16. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, vol. 30 (2017)
17. Kang, W.-C., McAuley, J.: Self-attentive sequential recommendation. In: 2018 IEEE International Conference on Data Mining (ICDM), pp. 197–206. IEEE (2018)
18. Sun, F., et al.: Bert4rec: sequential recommendation with bidirectional encoder representations from transformer. In: Proceedings of the 28th ACM International Conference on Information and Knowledge Management, pp. 1441–1450 (2019)
19. Wu, S.: Session-based recommendation with graph neural networks. In: AAAI (2020)
20. Zhou, G., et al.: Deep interest network for click-through rate prediction. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 1059–1068 (2018)
21. Koren, Y.: Collaborative filtering with temporal dynamics. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 447–456 (2009)
22. Campos, P.G., Díez, F., Cantador, I.: Time-aware recommender systems: a comprehensive survey and analysis of existing evaluation protocols. User Model. User-Adap. Inter. **24**(1–2), 67–119 (2014)
23. Quadrana, M., Karatzoglou, A., Hidasi, B., Cremonesi, P.: Personalizing session-based recommendations with hierarchical recurrent neural networks. In: Proceedings of the Eleventh ACM Conference on Recommender Systems, pp. 130–137 (2017)
24. de Souza Pereira Moreira, G., Rabhi, S., Lee, J.M., Ak, R., Oldridge, E.: Transformers4Rec: bridging the gap between NLP and sequential/session-based recommendation. In: Proceedings of the 15th ACM Conference on Recommender Systems, pp. 143–153 (2021)