2025 IEEE Swiss Conference on Data Science (SDS) | 979-8-3315-9467-1/25/$31.00 ©2025 IEEE | DOI: 10.1109/SDS66131.2025.00018

The power of session-level data: Predicting return intent along the browsing journey

Yielva Kunz[1], Adriana Ricklin[1,2], Guang Lu[1]* & Thomas Wozniak[1]
yielva.kunz@stud.hslu.ch, adriana.ricklin@hslu.ch, guang.lu@hslu.ch, thomas.wozniak@hslu.ch
[1]*Institute of Communication and Marketing*, *Lucerne School of Business*, Lucerne, Switzerland
[2]*Department of Mathematics and Computer Science*, *Eindhoven University of Technology*, Eindhoven, The Netherlands

***Abstract*—Despite the growing challenge of product returns in e-commerce, limited research explores real-time prediction of return intent. Existing studies typically rely on customer relationship management (CRM) data and require historical context, such as a user's return behavior the last *n* orders. However, many online shoppers lack this context, because they either browse as guests and can thus not be identified in real-time or are first-time visitors. This reduces the accuracy and coverage of existing return intent assessment methods, which would otherwise lay the groundwork for timely interventions. This paper explores the potential for return prediction based only on browsing behavior using Long Short-Term Memory (LSTM) models. We simulate real-world settings by training and testing on a rolling-window approach on logged browsing data. We find for our model to attain F1 = 0.68 and ROC AUC = 0.70 on average. Furthermore, our model demonstrates superior accuracy compared to methods relying on CRM data without historical user context. A subsequent analysis of the important features, based on the trained LSTM model, draws insights into which browsing behaviors may be early warning signals indicating return. For instance, having a high number of items in the shopping cart from a previous browsing session is strongly associated with high-value returns. Meanwhile, for desktop users, the use of filters appears to be a driving indicator for returns. In a hypothetical calculation, we illustrate the positive economic impact our approach enables to e-commerce companies. Our findings highlight the potential of utilizing browsing behavior for prediction of return intent, providing a foundation for developing real-time return intent assessment strategies.**

***Keywords*—browsing behavior, clickstream, user intent, product returns, e-commerce**

I. Introduction

Worldwide e-commerce and online retail sales have significantly grown in the past decade. From 1.3 trillion USD in 2014, they are expected to increase more than sixfold to 8 trillion USD by 2027 [1]. This expansion comes with a significant challenge: the increasing volume of product returns. In Switzerland for example, clothing and accessories exhibit return rates of up to 60% [2]. The surge in returns not only generates substantial economic costs for online retailers due to the costs of inspection or processing. It also increases the environmental impact of e-commerce, through increased CO_2 emissions [3], [4].

Existing studies have attempted to predict which items are likely to be returned or the reasons for returns after purchases have been completed [5], [6], [7]. Their focus was on post-purchase returns management or relied on user history, making them unsuitable for real-time prediction, as they are inherently reactive and overlook the pivotal window of opportunity to prevent returns during the active online shopping session itself.

However, analyzing the probability of returns at session level in real-time offers a crucial first step towards prevention. By identifying browsing behavior that indicates a likely return before a purchase is made, retailers can proactively implement return-reducing strategies, like providing behavioral interventions or personalized recommendations to those exhibiting such behavior. While knowing the predicted return probability in real-time is only part of the equation for successful interventions (the other part being the reaction to an intervention itself), it is a key component in targeting strategies: Return probability serves as a critical baseline for holistic targeting strategies that combine both user inclination and the effect of an intervention [8]. Furthermore, a focus on live analysis not only enables timely interventions when issues or risks emerge, but also aligns more closely with user interests. Research has shown that, for example, recommendations based on the current browsing session are more relevant to the user than those relying on previous browsing sessions [9]. Hence, to enable real-time interventions, a modeling approach capable of operating online is required that leverages live data rather than historical records. Such a framework enables retailers to dynamically assess the probability of a return while a shopper is browsing, using live behavioral data to inform immediate interventions.

This paper proposes an approach that models return intent in real time, based solely on the browsing behavior of users during a session. We analyze return probabilities for individual users on the levels of order, items and value. By addressing these three levels, we provide a granular understanding of returns. Furthermore, by analyzing browsing patterns during the pre-purchase stage through Long Short-Term Memory (LSTM) models, a method that has proven effective in e-commerce applications, we contribute to both methodological and domain-specific advancements. Methodologically, we expand the field of LSTM applications by adapting them to real-time return prediction. From a domain perspective, we advance

* Corresponding author: guang.lu@hslu.ch.

2835-3420/25/$31.00 ©2025 IEEE
DOI 10.1109/SDS66131.2025.00018

return research by focusing exclusively on online browsing behavior and identifying the factors driving returns before a purchase has even occurred.

II. Related Work

A. Prediction of Return Behavior

While return prediction has been studied under different problem formulations, we focus here on research leveraging pre-purchase data to predict returns, meaning approaches that utilize information available before a purchase transaction is completed. This mainly involves leveraging product characteristics, customer-specific factors, or a combination of both to study whether any return has occured. As a result, academic contributions that examine returns on an item or value level perspective remain relatively underexplored.

In [10], returns are seen as a matter of basket compositions and return records. The authors use a graph representation, where nodes represent shopping baskets and edges connect baskets with shared products, enabling the identification of return-prone baskets through multi-product co-occurrences. [5] also focuses on products as the central element in return management by analyzing the product attributes as drivers for return probability. Their regression analysis finds attributes such as size, order quantity, or color to be associated with a higher propensity to return. [11] expands the traditional product-focused approach by incorporating customer-centric elements into various binary classification algorithms and finds that both product and customer characteristics - such as demographic attributes like age - play a crucial role in returns. Similarly to these findings, [6] emphasizes the importance of both product and customer information when modeling return intent. The authors note that leveraging product-related data alone does not explain the return propensity well enough; instead, customer-related information such as past return behavior is essential, as models without historical context struggle to achieve reliable accuracy. Although the inclusion of such data enhances the performance of several return prediction algorithms, the authors caution that the effectiveness of this approach is constrained by data availability, meaning it mainly benefits existing customers and cannot properly handle users that are not logged-in or new.

While these studies provide valuable insights into return prediction from a pre-purchase perspective, they hint at opportunities to further enrich predictive approaches. Their focus on traditional data sources like product attributes and customer demographics leaves room to explore the potential of user clickstream data. This dynamic form of data, which captures user behavior in real-time, represents a research gap to our knowledge, despite its potential to provide deeper insights into early warning signals for the return behavior of a new or unsubscribed user.

B. Modelling User Intent by LSTM within E-Commerce

User behavior on e-commerce platforms generates rich sequential data, reflecting actions like browsing, comparing, and decision-making. By leveraging LSTM networks, which are less resource intensive than more complex methods, these interaction sequences can be analyzed to identify patterns that indicate user intent, such as prolonged page visits, repeated product views, or frequent cart modifications. This flexibility makes LSTM suitable for inference in real-time scenarios.

The adaptability of LSTM has been demonstrated in various e-commerce applications. For example, [12] predicts the outcomes of the shopping session and demonstrated that LSTMs generalize better and with fewer data than high-order Markov chain models. [13] applies LSTM to capture changes in dwell time in user browsing behavior and [14] utilizes LSTM to predict user click behavior and dynamically model user interests. Similarly, [15] employs LSTM to perform real-time analysis of short-term browsing patterns for purchase intent prediction. [16] also models purchase intent by LSTM, successfully benchmarking it against other approaches. Furthermore, [17] develops LSTM-based frameworks to predict user abandonment within specific time frames.

Despite these advancements, the use of LSTM to predict product return intent remains underexplored. This presents a valuable opportunity to apply LSTM architectures to session-level clickstream data, potentially predicting the likelihood of product returns even before a purchase is completed.

III. Methodology

A. Problem Formulation

For any user at any point in their browsing journey, the objective is to estimate the probability of a return, assuming they make a purchase. Adapted from [18], return behavior r for a user u can be analyzed from three perspectives: Alpha, Beta, and Gamma. The Alpha return metric α_u, used to binarily predict whether a return will occur regardless of the number of items, is defined as:

$$\alpha_u = \begin{cases} 1, & \text{if user } u \text{ returns any item} \\ 0, & \text{if user } u \text{ does not return any item} \end{cases} \tag{1}$$

The Beta return metric β_u, a quantity-based metric, measures the proportion of returned items relative to the total number of items shipped:

$$\beta_u = \frac{\text{number of returned items by user } u}{\text{number of shipped items to user } u} \tag{2}$$

The Gamma return metric γ_u evaluates the monetary value of returned goods relative to the total value of shipped goods of a user. This metric provides a financial perspective on returns, offering insights into the revenue impact of returned orders:

$$\gamma_u = \frac{\text{value of returned items by user } u}{\text{value of shipped items to user } u} \tag{3}$$

We examine all three return definitions. The measure α_u is used to assess the overall probability associated with a return. Building upon the analysis of α_u, we extend our investigation to β_u and γ_u. This allows to address questions such as, ”Will the return involve all items?” (as characterized by β_u) or ”Will high-value items be returned?” (as characterized by γ_u), thus gaining a more comprehensive understanding of returns from multiple perspectives.

We formulate the return problem as follows. Let $X = \{x_1, x_2, \ldots, x_t\}$ represent the sequence of user interactions (clickstream data) up to any point of interest t, where each x_i is a feature vector corresponding to the i-th interaction (e.g., product views, time spent, cart modifications). The objective is to train a model that, given the observed sequence X and under consideration of a threshold θ, can reliably predict the outcome $\hat{r}_u$, where $\hat{r}_u \in [0, \theta)$ indicates no return and $\hat{r}_u \in [\theta, 1]$ indicates a return.

We choose to apply the problem only to α_u. For where $\hat{\alpha}_u$ indicates a return, we then follow up with an analysis on β_u, or γ_u, that are discretized into three bins to facilitate interpretation and analysis: (0.0, 0.6], (0.6, 0.9] and (0.9, 1.0].

B. Dataset

We leverage clickstream data together with corresponding returns data from a European fashion retailer as follows[1]:

1) Clickstream Data: The clickstream data spans a 26-week period from 1. February 2024 to 31. July 2024 and contains over two million unique browsing sessions. 30.8% of these were carried out on desktop, whereas 65.7% can be attributed to mobile devices. Only 3.5% of all unique browsing sessions were on tablet. The dataset further records detailed user interactions, such as page loads that identify which pages were accessed during a session, interactions on the pages - like clicks and hovers over specific elements or components within a page - as well as other actions that capture additional behavioral data, such as filter or search activities (see Table I). In addition to explicit actions, such as scrolling, adding a product to the cart, or changing pages, implicit signals are also recorded, like the time spent on a page, which results from the difference in `pageDate` between a previous and subsequent page.

2) Return Data: The return data contains information for each item of any order within the 26-week period whether the item was returned. Moreover, the returns are linked to their corresponding session identifiers, enabling a direct connection between each return and the user session in which the purchase was made. In doing so, the data shows that 44.0% of all returns were linked to orders placed on mobile, while 53.4% originated from desktop orders. The remaining 2.6% of returns were associated with orders made on tablets.

C. Preprocessing and Feature Engineering

The dataset was first divided into mobile and desktop sessions, as device type is a critical factor influencing browsing behavior [19], [20]. Sessions with tablets were excluded due to low occurrence. The dataset was then filtered to include only `sessionId`'s associated with a purchase. Cleaning steps included excluding sessions with clickstream lengths (as measured by a session's last `socketId` value) that are fewer than 3 or more than 400 [21]. Sessions likely to be generated by bots based on characteristics such as fast, repetitive click sequences were also excluded. To handle imbalances in the variables of interest, balancing strategies were implemented. For α_u, this refers to weighted sampling as to balance orders with and without a return. For β_u and γ_u, oversampling techniques such as SMOTE, borderline SMOTE, or ADASYN [22], [23] were evaluated to augment minority classes, which refers to the bins (0.0, 0.6] and (0.6, 0.9].

In a feature engineering step, numerous variables were derived from Table I, Table II, and their combination. To ensure the robustness of the final feature set, feature permutation testing was employed. Features that exhibited high redundancy, low variance, or limited predictive value were systematically excluded (see Figure 1 for an example). This resulted in 29 features in four categories (see Table III). Transformation processing steps, such as one-hot-encoding for categorical variables or scaling for numerical values, were then tailored to the specific variable characteristics.

Lastly, we transformed the tabular data into sequences

TABLE I
VARIABLES OF THE CLICKSTREAM DATA

Data Type	Type	Description
sessionId	category	Session identifier for temporarily tracking within the same sessions (changes after 30 minutes of inactivity)
device	category	Device used to access the website (mobile, desktop, tablet)
socketId	integer	Unique identifier related to real-time data transmission sockets, which can also serve as a counter for the number of pages loaded during a session
pageDate	datetime	Date and time when a specific page (`pageUrl`) was accessed
pageType	category	Type of page the user visited (e.g., ”productList”, ”productPage”)
pageUrl	string	The full URL of the visited page
checkout	binary	Indicating whether a checkout occurred during the session
checkoutValue	float	Total revenue if a checkout occurred during the session
pageData	dict	Containing structured meta data related to the visited page, such as product details (e.g., color, size) on product detail pages or listed products on product list pages
eventType	category	The type of event triggered during the session, e.g. ”scroll” or ”click”
eventSource	category	The source of the event / certain submodules on a page that capture user interaction more granularly, e.g. button ”addToCart”

TABLE II
VARIABLES OF THE RETURN DATA

Data Type	Type	Description
sessionId	category	Session identifier of the order
itemNr	category	Article number a sessionId ordered
returnStatus	binary	Whether the item was returned (1) or not (0)

[1] The data sources used are confidential.

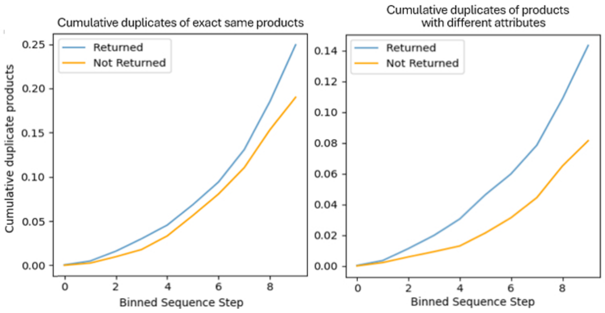

Fig. 1. Example of feature selection based on predictive value. The x-axis represents binned sequence steps, where user interactions within each order are normalized and divided into ten equal segments. By binning the sequences, patterns can be analyzed consistently, regardless of whether a user had a short or long shopping session. Cumulative duplicates of exact same products (cumulative_duplicate_products), which count identical products added multiple times to the cart, show little difference between returned and non-returned orders, indicating low predictive value. In contrast, product with different attributes (cumulative_diff_attr_duplicates), which track similar products with varying attributes (e.g., size or color), more clearly distinguish returned orders. This suggests that while exact duplicates do not strongly indicate return intent, attribute variations are more relevant for predicting returns.

TABLE III
SELECTED INPUT VARIABLES

Feature Group	Description
Cart features	Capturing cart-related actions related to the monetary value of the cart (e.g., cart_amount or relative_coupon_value), number of items in cart (items_in_cart), as well as cart modifications (e.g., cart_change). It also captures the total value of items added in the current session (cum_price_addToCart) as well as the cumulative sum of discounts added (cum_price_diff_addToCart)
Product attributes	Describing product diversity in cart, like number of duplicate articles in different sizes (e.g. cumulative_duplicateSizes) or different colors (cumulative_duplicateColors).
Interaction features	Representing user interactions along the browsing sessions, e.g., clicks (eventType_click), how many times the product image was changed (eventSource_imageChange), or whether filters or the search function was applied (filterSearchActivity).
Temporal features	Indicating time spent on the different page types (e.g. pageType_productList).

that are suitable for our LSTM approach. Using a sliding window technique, we leveraged user interactions (grouped by sessionId), to define sequence lengths of 50 and a stride of 30. This method generated overlapping sequences, where each window captured a segment of interactions and advanced by the stride length. By overlapping, temporal dependencies were maintained and data omissions were avoided at the same time (see Figure 2). To maintain consistent sequence dimensions, both sessions or single sequences with a length shorter than 50 are padded with a placeholder value of −999. This value is deliberately chosen to avoid conflicts with valid data, such as binary indicators where 0 holds meaningful information, ensuring a clear distinction between padding and actual data.

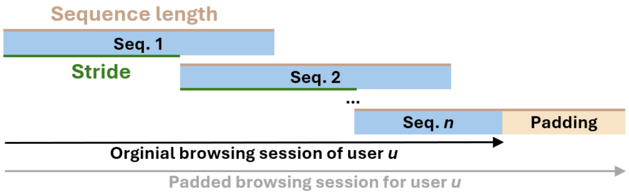
Sequence length
Seq. 1
Stride
Seq. 2
...
Seq. n
Padding
Orginial browsing session of user u
Padded browsing session for user u

Fig. 2. Transforming data into sequences and strides for LSTM.

D. Model Fitting

We tested three distinct LSTM architectures (see Table IV): a single-layer LSTM, a two-layer LSTM, and an architecture consisting of an LSTM layer followed by a Gated Recurrent Unit (GRU) layer. Since the binary classification task α_u served as the starting point, each architecture was hyperparameter-tuned for α_u. To provide a baseline, we also evaluated a logistic regression model on the α_u task.

The best performing set-up of the three LSTM models was then adapted for β_u and γ_u. We wanted to see whether a model set-up that captures meaningful patterns for α_u is suited to do the same for β_u and γ_u. In application, this meant that for where α_u indicated a return, the multi-class classification models for β_u and γ_u extended the analysis to encompass more nuanced return behaviors.

For α_u, we used binary cross-entropy as a loss function, whereas for β_u and γ_u, we applied categorical cross-entropy. We trained the networks using the Adam optimizer for its adaptive learning rate. Different configurations of variables, including batch_size and dropout_rate, were explored. Combined with advanced callbacks such as ReduceLROnPlateau and EarlyStopping, these adjustments facilitated stable convergence while minimizing overfitting. These settings were then applied to three different architectures to evaluate which performs best.

TABLE IV
SELECTED ARCHITECTURES AND PARAMETERS

Parameter	single-layer LSTM	two-layer LSTM	LSTM-GRU hybrid
Number of layers	1 (LSTM)	2 (LSTM)	2 (1 LSTM, 1 GRU)
Units	200	200	LSTM = 256, GRU = 128
Epochs	40	40	40
Batch size	32*	32*	32*
Dropout rate	0.3	0.3	0.4

*batch size = 64 for the multi-class classification setting (β_u and γ_u) .

E. Evaluation

1) Error Metrics: The evaluation of all three return metrics is conceptualized as a classification task, where α_u is treated as a binary classification problem, and β_u and γ_u are framed as multiclass classification problems. The performance of the models are assessed using evaluation metrics that leverage different ratios of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) predictions. These metrics include accuracy, recall, and F1-score, as well as specificity, which measures whether a negative prediction is likely to be correct.

$$\text{Specificity} = \frac{TN}{TN + FP} \tag{4}$$

In classification problems, the threshold determines how predicted probabilities are assigned to classes, influencing the balance between false positives and false negatives. Lower thresholds increase sensitivity (recall) but raise false positive rates. To take care of this, we also evaluate model performance by the Receiver Operating Characteristic Area Under the Curve score (ROC AUC), as it is a threshold-independent metric that captures the trade-off between sensitivity and specificity across thresholds, offering a comprehensive assessment of classifier performance. For the multi-class classification tasks, the classification metrics were extended by a weighted averaging of class-wise performances, whereas ROC AUC was evaluated using a One-vs-Rest approach.

2) Train-Test Setting: To account for potential drift commonly observed in clickstream data and assess the broader applicability of our approach in real-world settings, we implemented a rolling window retraining strategy to simulate a dynamic environment. This method reflects real-world conditions, where new data becomes periodically available, enabling the model to continuously adapt to evolving patterns and concept drift. The simulation is governed by four key elements: a train window, a test window, a gap between the train and the test set to better simulate the evolving conditions between the sets (see Figure 3), and an iterative approach.

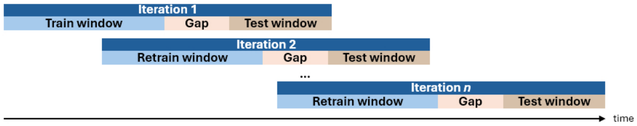

Fig. 3. Generalized framework of our train-test setting.

In each iteration, the model was trained on 12 weeks of data. To ensure a clear separation between training and testing, a 15-day gap was introduced. This gap was included to simulate potential data drift and thus helped to evaluate model performance based on a realistic scenario. Following the gap, the model was tested on the subsequent 10-day set of data. After each cycle, the data window was shifted by 10 days, and the model was incrementally re-trained using the updated 12-week window. This iterative process ensured that the model continuously adapts to the latest data, incorporating new information with each update. This training procedure led to six iterations.

3) Feature Importance: We analyzed feature importance by the application of feature permutation, accuracy dropout, and integrated gradients (IG) to understand drivers behind returns. IGs quantify the contribution of each input feature to the prediction by integrating gradients along a linear path from a baseline to an input [24]. Mathematically, it is expressed as:

$$\text{IG}_i(x) = (x_i - x_i') \cdot \int_{\alpha=0}^{1} \frac{\partial F\left(x' + \alpha \cdot (x - x')\right)}{\partial x_i} d\alpha, \tag{5}$$

where x_i is the feature value, F the output of the model, and x_i' the baseline, representing an input where there is no information. By integrating along the path from the baseline to the actual input, we capture how each feature contributes to the output prediction. IG integrates seamlessly with LSTM via backpropagation and is computationally efficient [24].

IV. Results

A. Model Performance

Whereas all models surpass the baseline logistic regression that achieved on the test set an F1 score of 0.45, the performance evaluation of the single-layer, the two-layer as well as the LSTM-GRU hybrid for α_u reveals differences amongst each other (see Table V). On the validation set, the hybrid model achieves the highest accuracy (0.67), suggesting its robustness in recognizing patterns, although its slightly higher loss (0.62) compared to the single-layer LSTM (0.61) indicates a marginal trade-off in prediction confidence. Interestingly, all models show equal ROC AUC (0.73) and recall (0.67), highlighting comparable capabilities in distinguishing between positive and negative cases. Specificity values favor the single-layer LSTM and hybrid models, with both achieving 0.72, surpassing the two-layer model. On the test set, the single-layer LSTM demonstrates superior ability to generalize with the highest accuracy (0.65), specificity (0.68), and F1 score (0.68). While the two-layer LSTM marginally outperforms others in ROC AUC (0.71), the hybrid model achieves the lowest loss (0.70), indicating potential advantages in producing calibrated predictions. Although direct comparisons are limited by data set differences, it is also noteworthy that the methods reported in [6] struggled to surpass an accuracy of 0.61, whereas our approaches exceed this benchmark.

TABLE V
Average validation errors for α_u on desktop across 6 iterations

	Single-layer		**Two-layer**		**LSTM-GRU**	
	Val	**Test**	**Val**	**Test**	**Val**	**Test**
Accuracy	0.66	0.65	0.66	0.65	0.67	0.63
ROC AUC	0.73	0.70	0.72	0.71	0.73	0.69
Recall	0.63	0.62	0.76	0.62	0.63	0.58
F1	0.70	0.68	0.69	0.68	0.70	0.66
Specificity	0.72	0.69	0.70	0.68	0.72	0.71

Overall, the single-layer LSTM emerges as the most consistent model, balancing performance across metrics. Applying this setting to the mobile scenario shows the architecture sustains its performance on the mobile test set. However, when considering the more complex problems β_u and γ_u, the performance profile shifts (see Table VI). For both β_u and γ_u, there was a shift in performance resulting in reduced values for metrics such as accuracy, recall, and F1-score compared to α_u. Interestingly, specificity values were not similarly impacted and remained at comparable or slightly higher levels, possibly because specificity is less sensitive to the increase in classification complexity. ROC AUC also indicated some retention of ranking ability. Overall, distinguishing β_u and γ_u return patterns remains challenging despite the data augmentation, possibly due to the complexity of these return types, subtle differences in the predictive features or the sensitivity of the model to data characteristics. Nonetheless, the model provides valuable insights for analyzing the importance of features.

TABLE VI
AVERAGE TEST ERRORS ± STANDARD DEVIATION ACROSS 6 ITERATIONS

Return Metric	α_u		β_u		γ_u	
Device	**Desktop**	**Mobile**	**Desktop**	**Mobile**	**Desktop**	**Mobile**
Accuracy	0.65 ±0.01	0.64 ±0.01	0.51 ±0.02	0.51 ±0.01	0.50 ±0.03	0.51 ±0.02
ROC AUC	0.70 ±0.01	0.70 ±0.01	0.67 ±0.01	0.69 ±0.02	0.67 ±0.03	0.66 ±0.02
Recall	0.62 ±0.04	0.55 ±0.02	0.47 ±0.02	0.48 ±0.02	0.47 ±0.03	0.47 ±0.03
F1	0.68 ±0.03	0.64 ±0.02	0.46 ±0.02	0.46 ±0.02	0.47 ±0.03	0.47 ±0.03
Specificity	0.69 ±0.04	0.75 ±0.03	0.73 ±0.08	0.74 ±0.08	0.73 ±0.04	0.77 ±0.07

B. Drivers of Return in Browsing Behavior

1) Alpha Return α_u: We find that desktop users without returns often begin sessions with empty or minimally populated carts. Subsequent shopping behavior results in high `cum_price_addToCart` values, a variable that tracks the sum of the prices of all products added, similar to `cart_amount`, but without accounting for removals or items already present in the cart from a previous session. Conversly, users with a lower `cum_price_addToCart` but a high `cart_amount` are more likely to return items (see also Figure 4). This situation, where cart value is high but monetary value added during the current session is low, can likely be explained by customers who added items in a previous session, left, and then returned to finalize their purchases, hence performed cart abandonment in an earlier session. Also, high `relative_coupon_value` is likely to drive return items. This circumstance in combination with a high `cart_value` aligns with the tendency of desktop users to engage in bulk purchasing or "testing", where multiple items are bought with the intention of returning those that do not meet expectations. We also observe that the application of filters (`FilterSearchActivity`) increases the likelihood of a return, as does spending time on the cart page (`pageType_Cart`) or spending little time on product list pages (`pageType_ProductList`).

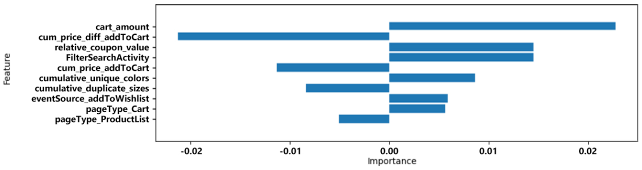

Fig. 4. Top 10 features for desktop users with $\hat{\alpha}_u = 1$.

In contrast, the dynamics for mobile users are reversed (see Figure 5). The high negative attribution of `cart_amount` indicates that smaller final cart values are associated with a higher likelihood of returns. In combination with high monetary savings in the cart, this suggests that mobile shoppers often make quick, impulsive purchases of a few, discounted items, which results in returns. The contrasting attributions of `saving_cart` and `relative_coupon_value` highlight two distinct ways mobile users evaluate discounts: Absolute savings encourage riskier, less deliberate purchases at the cart level, while relative discounts at the item level reduce returns by reinforcing the perceived value of specific items.

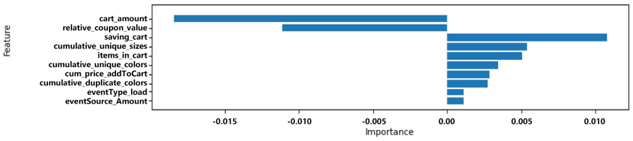

Fig. 5. Top 10 features for mobile users with $\hat{\alpha}_u = 1$.

2) Beta Return β_u: We find that high value features denoting the diversity in the cart (such as number of items expressed by `items_in_cart`), monetary aspects, or savings are strongly associated with an increased likelihood of full returns. A positive attribution of `saving_cart` highlights the tendency of discounted items or purchases by price-sensitive customers to exhibit higher return rates. Furthermore, we see the pattern strengthened that low `cum_price_addToCart` and high `cart_amount` values are crucial factors in the return question, as they enhance the probability for a full return. This holds true for desktop as well as for mobile (see Figure 6 and Figure 7).

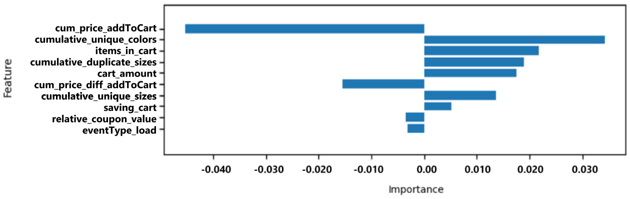

Fig. 6. Top 10 features for desktop users with $\hat{\beta}_u = 1$.

3) Gamma Return γ_u: We observe the same feature importance pattern in the case of full returns as for β_u: namely, a high `cart_amount` combined with a low `cum_price_addToCart`. In cases with lower returns $\hat{\gamma}_u$, the situation is more nuanced: Sessions where $\hat{\gamma}_u \in (0.6, 0.9]$ started with empty baskets and accumulated high cart values

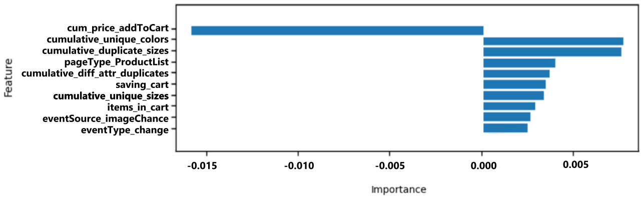

Fig. 7. Top 10 features for mobile users with $\hat{\beta}_u = 1$.

progressively (see Figure 8). Browsing instances with a lower $\hat{\gamma}_u \in (0.0, 0.6]$, are explained by smaller final cart values and minimal initial basket content, reflecting focused and intentional shopping (see Figure 9).

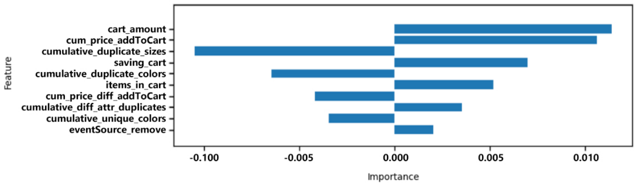

Fig. 8. Top 10 features for desktop users with $\hat{\gamma}_u \in (0.6, 0.9]$.

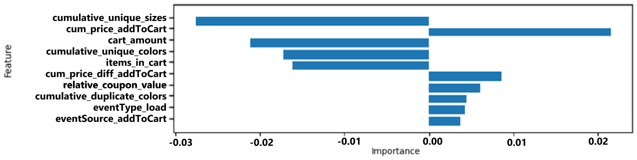

Fig. 9. Top 10 features for desktop users with $\hat{\gamma}_u \in (0.0, 0.6]$.

V. Business Implications

Followingly, we show how real-time return probability modeling could benefit e-commerce platforms by focusing on the financial impact our approach enables in a hypothetical scenario. We do this by calculating savings s from preventing returns through the successful identification of at-risk users, while accounting for misclassification costs c. Assume a fashion retailer with 1'800 monthly online orders, operating under conditions as follows: The average order value (AOV) is \$196 [25], with each order containing approximately 2.46 items [26], resulting in an average item price of \$80. Minimizing returns is a priority due to their high costs: 60% of orders return at least one item [18], incurring processing costs (PC) of \$20 and value depreciation (VC) of 18% per item [18]. Additionally, assume returns always only comprise one item and that 50% of returns are unrelated to product quality, so that they could be prevented with the right intervention. Applying the LSTM model for α_u, which correctly identifies returns in 62% of cases (see Table V), would lead to the following savings:

$$\begin{aligned} \mathrm{s} &= \mathrm{returns}_{\mathrm{avoided}} \times (\mathrm{PC}_{\mathrm{saved}} + \mathrm{VC}_{\mathrm{saved}}) \\ &= 0.62 \times (1'800 \times 0.60) \times \big[(20 \times 1) + (80 \times 0.18)\big] \\ &= 37'015 \times 50\%\ \text{success rate} \approx \$18'500 \end{aligned} \tag{6}$$

Costs of missclassification occur when customers who do not need interventions are targeted regardless, leading to disengagement, and revenue loss. The model specificity of 0.69 implies a FP rate that misclassifies non-returns as returns of 31%. We also assume a disengagement rate (DR) of 20%, meaning 20% of those misclassified decide to not follow-up on their purchase intentions. This creates lost revenue of:

$$\begin{aligned} \mathrm{c} &= (\text{non-returns} \times \mathrm{FP} \times \mathrm{DR}) \times \mathrm{AOV} \\ &= \big[(1'800 \times 0.4) \times 0.31 \times 0.2\big] \times 196 = \$8'750 \end{aligned} \tag{7}$$

Calculated on the basis of a year, this results in a yearly net impact (NI) of \$117'000 to be saved in this scenario by implementing targeting solely based on return probability:

$$\mathrm{NI} = 12 \times (18'500 - 8'750) = \$117'000 \tag{8}$$

VI. Conclusion

This paper introduces a real-time return prediction model that relies solely on browsing behavior. Validated through real-life simulation, it demonstrates strong practical effectiveness. Furthermore, our approach shows big potential, as our model for α_u achieves higher accuracy than other return intent models that do not leverage historical data.

We provide behavioral insights into return drivers: High cart values and increased cart diversity, represented by features such as `cumulative_unique_colors`, are strongly associated with higher return probabilities, reflecting exploratory shopping patterns and customer uncertainty. Similarly, pricing incentives, including `saving_cart`, influence return behavior by encouraging riskier purchase decisions, which often lead to returns. Our analysis further suggests that preexisting cart content heightens the probability of return.

We see distinct differences when it comes to feature importance across the different return metrics. For α_u, behavioral features are important: `FilterSearchActivity` or time spent on pages can be indicative. In contrast, for β_u and γ_u, cart-related features such as cart diversity are more crucial.

We show that return signals across devices are not necessarily the same. Desktop users tend to have a stronger inclination towards returns if they have a high `cart_amount` or applied `FilterSearchActivity`, whereas risk factors for mobile users are better expressed by metrics relating to potential monetary savings, such as `relative_coupon_value` or `saving_cart`. This indicates further that analyzing return probability with a contextual understanding can be beneficial.

Hence, we move beyond conventional approaches based on historical purchase data by demonstrating that real-time behavioral cues from clickstream data — without relying on personal information — can effectively predict return intent, offering a privacy-conscious alternative to prevailing methods in the literature.

A. Limitations

The results indicate that while the model optimized for α_u captures meaningful patterns, its suitability for β_u and γ_u is limited. The relatively stable ROC AUC and specificity suggest that some underlying predictive capacity transfers across tasks, but further adaption is needed to fully address the complexities of β_u and γ_u. Whereas this is partly due to the increased complexity of the multi-class tasks, it also highlights that the factors driving returns in a binary classification setting may differ from those influencing returns in a more granular, multi-class context. Additional to that, it is of influence that model tuning was not done for β_u and γ_u. Instead, the models that analyze the returns from these two perspectives run under an architecture fine-tuned for α_u. Furthermore, our work is based on a single data set spanning 26 weeks of a fashion retailer. Hence, the generalization of our findings need to be further validated. We also acknowledge that our findings might be of main interest to fashion retailers only, as some of the derived features, like `cumulative_unique_sizes`, might not be as predictive in industries like electronics or home goods, where product attributes are less varied.

B. Outlook

Future research modeling return intent should focus on evaluating model architectures, including on how to combine predictions or returns at different levels (α_u, β_u, γ_u). Furthermore, we advise to explore device-specific return indicators to refine and expand the applicability of the presented predictive frameworks. Finally, future research should investigate integrated targeting strategies that combine our approach with tailored actions to fully realize the potential of return prevention.

References

[1] Statista, "Retail e-commerce sales worldwide from 2014 to 2027," 2023. [Online]. Available: https://www.statista.com/statistics/379046/worldwide-retail-e-commerce-sales/

[2] R. Raemy and T. Wozniak, "Returns in Swiss online retail – Online Retailer Survey 2022," 2022. [Online]. Available: https://digital-commerce.post.ch/en/pages/blog/2022/returns-in-swiss-online-retail-online-retailer-survey-2022

[3] R. Frei, L. Jack, and S. Brown, "Product returns: a growing problem for business, society and environment," *International Journal of Operations & Production Management*, vol. 40, no. 10, pp. 1613–1621, Jun. 2020, doi: 10.1108/IJOPM-02-2020-0083.

[4] R. Mangiaracina, G. Marchet, S. Perotti, and A. Tumino, "A review of the environmental implications of B2C e-commerce: a logistics perspective," *International Journal of Physical Distribution & Logistics Management*, vol. 45, no. 6, pp. 565–591, Jul. 2015, doi: 10.1108/IJPDLM-06-2014-0133.

[5] B. Balasubramanian and K. T. Perannagari, "Investigating Return Behavior in e-commerce: A Data-Driven Study," in *2024 IEEE International Conference on Interdisciplinary Approaches in Technology and Management for Social Innovation (IATMSI)*. Gwalior, India: IEEE, Mar. 2024, pp. 1–6, doi: 10.1109/IATMSI60426.2024.10502719.

[6] M. Niederlaender, A. Lodi, S. Gry, R. Biswas, and D. Werth, "Garment Returns Prediction for AI-Based Processing and Waste Reduction in E-Commerce:," in *Proceedings of the 16th International Conference on Agents and Artificial Intelligence*. Rome, Italy: SCITEPRESS - Science and Technology Publications, 2024, pp. 156–164, doi: 10.5220/0012321300003636.

[7] M. Farber, S. Novgorodov, and I. Guy, "Learning Reasons for Product Returns on E-Commerce," in *Proceedings of the Seventh Workshop on e-Commerce and NLP @ LREC-COLING 2024*, May 2024. [Online]. Available: https://aclanthology.org/2024.ecnlp-1.1.pdf

[8] S. Athey, N. Keleher, and J. Spiess, "Machine Learning Who to Nudge: Causal vs Predictive Targeting in a Field Experiment on Student Financial Aid Renewal," May 2024, doi: 10.48550/arXiv.2310.08672.

[9] E. Pöyry, N. Hietaniemi, P. Parvinen, J. Hamari, and M. Kaptein, "Personalized Product Recommendations: Evidence from the Field," 2017, doi: 10.24251/HICSS.2017.467.

[10] J. Li, J. He, and Y. Zhu, "E-tail Product Return Prediction via Hypergraph-based Local Graph Cut," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. London United Kingdom: ACM, Jul. 2018, pp. 519–527, doi: 10.1145/3219819.3219829.

[11] E. Safari, S. Pourhashemi, and M. Gharakhani, "A New Model for Predicting the Probability of Product Return in Online Shopping," May 2020. [Online]. Available: https://www.jscdss.com/index.php/files/article/view/232

[12] A. Toth, L. Tan, G. D. Fabbrizio, and A. Datta, "Predicting Shopping Behavior with Mixture of RNNs," 2017. [Online]. Available: https://www.difabbrizio.com/papers/sigir-ecom-2017-cs.pdf

[13] Y. Zhu, H. Li, Y. Liao, B. Wang, Z. Guan, H. Liu, and D. Cai, "What to Do Next: Modeling User Behaviors by Time-LSTM," in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*. Melbourne, Australia: International Joint Conferences on Artificial Intelligence Organization, Aug. 2017, pp. 3602–3608, doi: 10.24963/ijcai.2017/504.

[14] Z. Gharibshah, X. Zhu, A. Hainline, and M. Conway, "Deep Learning for User Interest and Response Prediction in Online Display Advertising," *Data Science and Engineering*, vol. 5, no. 1, pp. 12–26, Mar. 2020, doi: 10.1007/s41019-019-00115-y.

[15] K. Diamantaras, M. Salampasis, A. Katsalis, and K. Christantonis, "Predicting Shopping Intent of e-Commerce Users using LSTM Recurrent Neural Networks:," in *Proceedings of the 10th International Conference on Data Science, Technology and Applications*. SCITEPRESS - Science and Technology Publications, 2021, pp. 252–259, doi: 10.5220/0010554102520259.

[16] B. Requena, G. Cassani, J. Tagliabue, C. Greco, and L. Lacasa, "Shopper intent prediction from clickstream e-commerce data with minimal browsing information," *Scientific Reports*, vol. 10, no. 1, p. 16983, Oct. 2020, doi: 10.1038/s41598-020-73622-y.

[17] C. O. Sakar, S. O. Polat, M. Katircioglu, and Y. Kastro, "Real-time prediction of online shoppers' purchasing intention using multilayer perceptron and LSTM recurrent neural networks," *Neural Computing and Applications*, vol. 31, no. 10, pp. 6893–6908, Oct. 2019, doi: 10.1007/s00521-018-3523-0.

[18] Die Post, "Onlinehändlerstudie 2022 - Retourenmanagement im Onlinehandel," 2022. [Online]. Available: https://digital-commerce.post.ch/-/media/post-maxisites/e-commerce/dokumente/Whitepaper-DE.pdf?

[19] S. Park and K. Cho, "Mobile vs desktop user search behaviours of the 1300K site, a Korean shopping search engine," *The Electronic Library*, vol. 39, no. 2, pp. 239–257, Jun. 2021, doi: 10.1108/EL-09-2020-0261.

[20] T. Jiang, J. Yang, C. Yu, and Y. Sang, "A Clickstream Data Analysis of the Differences between Visiting Behaviors of Desktop and Mobile Users," *Data and Information Management*, vol. 2, no. 3, pp. 130–140, Dec. 2018, doi: 10.2478/dim-2018-0012.

[21] M. Von Zahn, K. Bauer, C. Mihale-Wilson, J. Jagow, M. Speicher, and O. Hinz, "The Smart Green Nudge: Reducing Product Returns through Enriched Digital Footprints & Causal Machine Learning," *SSRN Electronic Journal*, 2022, doi: 10.2139/ssrn.4262656.

[22] A. Fernandez, S. Garcia, F. Herrera, and N. V. Chawla, "SMOTE for Learning from Imbalanced Data: Progress and Challenges, Marking the 15-year Anniversary," *Journal of Artificial Intelligence Research*, vol. 61, pp. 863–905, Apr. 2018, doi: 10.1613/jair.1.11192.

[23] H. Majzoub and I. Elgedawy, "(PDF) AB-SMOTE: An Affinitive Borderline SMOTE Approach for Imbalanced Data Binary Classification," *ResearchGate*, Nov. 2024, doi: 10.18178/ijmlc.2020.10.1.894.

[24] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic Attribution for Deep Networks," Jun. 2017, doi: 10.48550/arXiv.1703.01365.

[25] Oberlo, "Average Order Value in Ecommerce (2016–2024)." [Online]. Available: https://www.oberlo.com/statistics/average-order-value

[26] vibetrace, "Items per Order." [Online]. Available: https://vibetrace.com/items-per-order/items-per-order-by-industry