



Article

# It's Not Always about Wide and Deep Models: Click-Through Rate Prediction with a Customer Behavior-Embedding Representation

Miguel Alves Gomes <sup>1,\*</sup> , Richard Meyes <sup>1</sup> , Philipp Meisen <sup>2</sup> and Tobias Meisen <sup>1</sup>

<sup>1</sup> Institute for Technologies and Management of Digital Transformation, University of Wuppertal, 42119 Wuppertal, Germany; meyes@uni-wuppertal.de (R.M.); meisen@uni-wuppertal.de (T.M.)

<sup>2</sup> Breinify Inc., San Francisco, CA 94105, USA; philipp.meisen@breinify.com

\* Correspondence: alvesgomes@uni-wuppertal.de; Tel.: +49-202-439-1733

**Abstract:** Alongside natural language processing and computer vision, large learning models have found their way into e-commerce. Especially, for recommender systems and click-through rate prediction, these models have shown great predictive power. In this work, we aim to predict the probability that a customer will click on a given recommendation, given only its current session. Therefore, we propose a two-stage approach consisting of a customer behavior-embedding representation and a recurrent neural network. In the first stage, we train a self-supervised skip-gram embedding on customer activity data. The resulting embedding representation is used in the second stage to encode the customer sequences which are then used as input to the learning model. Our proposed approach diverges from the prevailing trend of utilizing extensive end-to-end models for click-through rate prediction. The experiments, which incorporate a real-world industrial use case and a widely used as well as openly available benchmark dataset, demonstrate that our approach outperforms the current state-of-the-art models. Our approach predicts customers' click intention with an average F1 accuracy of 94% for the industrial use case which is one percentage point higher than the state-of-the-art baseline and an average F1 accuracy of 79% for the benchmark dataset, which outperforms the best tested state-of-the-art baseline by more than seven percentage points. The results show that, contrary to current trends in that field, large end-to-end models are not always needed. The analysis of our experiments suggests that the reason for the performance of our approach is the self-supervised pre-trained embedding of customer behavior that we use as the customer representation.

**Keywords:** click-through rate prediction; e-commerce; customer representation; embedding; LSTM

**JEL Classification:** C45; C53; C55; L81;



**Citation:** Alves Gomes, M.; Meyes, R.; Meisen, P.; Meisen, T. It's Not Always about Wide and Deep Models: Click-Through Rate Prediction with a Customer Behavior-Embedding Representation. *J. Theor. Appl. Electron. Commer. Res.* **2024**, *19*, 135–151. <https://doi.org/10.3390/jtaer19010008>

Academic Editors: Chao Wen, Jiaming Fang and Benjamin George

Received: 26 September 2023

Revised: 10 December 2023

Accepted: 28 December 2023

Published: 12 January 2024



**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Recently, large deep learning models have dominated various domains such as natural language processing (NLP) and computer vision (CV) in academia and industry. Since the introduction of the transformer model [1] in 2017, they have been repeatedly archiving state-of-the-art results. Recent examples like ChatGPT (<https://openai.com/blog/chatgpt/>) (accessed on 26 September 2023), GPT-3 [2], or Dall-E [3] show what such deep models are capable of. A similar trend can be observed in e-commerce, especially with recommender models like “Wide & Deep” [4] or Bert4Rec [5] and Click-Through Rate (CTR) prediction (CTR-P) models like “Deep & Cross” [6].

In the last years, CTR-P became a core task in online advertisement (also called ads) [7,8]. This is mainly because search engines, and especially recommender systems, are playing a significant role in e-commerce businesses [9–12]. Furthermore, predicting CTR accurately leads to a better user experience which has been shown to have a great impact on business effectiveness [8,13]. Additionally, CTR is a key performance indicator for online ads and

therefore, its prediction influences the ranking and price for online ads and revenue sponsored search [13–15]. Although there is a huge amount of data in the e-commerce sector, unlike natural language or images which have recurring patterns, customer behavior is subject to constant change as it is highly dependent on a variety of factors such as season, inflation, and local as well as global developments. In addition, the data are typically use case- and user-specific and are therefore limited in their ability to be shared across organizations. These two reasons raise the question of the extent to which deep and wide models are suitable in the context of e-commerce. Besides that, another aspect is that deep learning models require a considerable amount of computing resources, which is an ever-growing concern in light of the rising energy costs in our modern world. Furthermore, companies have limited resources and need to plan them accordingly [16]. Consequently, in the e-commerce sector, companies should ideally only use their resources on reactive customers, e.g., only display recommendations to those customers who are most likely to click on them. Lastly, advertising and recommendations can lead to negative experiences for certain customers, resulting in negative attitudes towards the operating company. This leads to shorter visit duration, fewer visits, fewer referral opportunities, and increased negative word-of-mouth. Therefore, it is crucial to only display advertising and recommendations when success is probable. Therefore, it is of great importance for a business to understand its customers' intentions and engage them with personalized targeting.

In this work, we approach CTR-P on recommendations for an online shop. Specifically, the goal is to determine whether the customer will click on the recommendation with the constraint that customers are not always logged in and, therefore, can be unknown to the online shop. Furthermore, we aim for our approach to be transferable to other use cases, which is why we additionally evaluate it on an open benchmark dataset commonly used in CTR research.

Recent state-of-the-art CTR-P approaches are based on end-to-end deep learning models. We break with this trend and propose an approach that decouples the customer representation from the CTR-P task. We train an embedding for this representation prior to the actual CTR-P task. We use the resulting pre-trained representation as an input to a subsequent Long Short-Term Memory (LSTM) classifier to predict the CTR and show in empirical experiments on two real-world use cases, that our approach outperforms state-of-the-art CTR-P models. Not only is our approach more accurate in terms of AUC and F1 score, but we also show that the pre-trained embeddings are better capable of capturing contextual and behavioral features from customer interaction data and help the LSTM to better generalize this customer behavior by not overfitting on the training data.

Our contribution to the research is as follows:

1. Our results show that baseline elements such as pre-trained embeddings and a recurrent neural network such as an LSTM can predict customer click behavior better than modern CTR prediction models, without the need for large end-to-end models.
2. In this respect, our approach results in reduced training time, making it more resource efficient.
3. Furthermore, task-independent pre-training of embeddings based on customer click-stream data is sufficient to model customers and prevent overfitting.

The remainder of this paper is structured as follows: in the next section, we present recent literature for CTR-P. In Section 3, we introduce the use case and describe the used datasets in detail. Section 4 describes all the necessary steps of our experiments to solve the stated problem and in addition provides theoretical details of our approach. In Section 5, we present our results and discuss the outcomes. Finally, we summarize our work and discuss its limitations as well as future research opportunities.

## 2. Related Work

### 2.1. Approaching Click-Through Rate Prediction

CTR-P received a lot of attention in industry and academia in the past years. It is approached as a binary classification problem, where the probability of an item click should

be predicted regardless of the use case, e.g., retrieved item in a search, clicked ad, or clicked product. In the literature, there is not one CTR-P use case, but multiple kinds of use cases. For example, Chen et al. [17], Ge et al. [18], and Fan et al. [9] propose a CTR-P model to optimize the retrieved items of a search engine. Others predict the CTR for shown ads [6,19] or products in general [10,20]. Table 1 presents a comprehensive overview of state-of-the-art CTR-P approaches including information of the authorship, publication year, proposed approach, used dataset, evaluation metric, and corresponding scores. All approaches are based on deep neural networks which means mixtures and ensembles of multi-layer perceptrons, recurrent layers, and attention layers that should capture customers' behavioral information. Furthermore, all models contain an embedding input layer to embed available information, which is usually given by the use case and/or selected by the data engineers. Typical input information is the user id, target item id, additional user information, and additional target item information. The DIN [21], DIEN [10], TIEN [22], and MARN [8] approaches use sequential activity information, which we also rely on in our approach. CTR approaches are evaluated on different datasets, some publications and approaches rely only on closed data [13,18,23–25] which are not included in Table 1. Others, as shown in Table 1, use openly available datasets to evaluate their approach. Of all the reviewed publications, the Amazon review dataset is the most used and, therefore, we also use this dataset to evaluate our approach. From previous works, we see that the standard evaluation metric for CTR-P is the Area Under the Receiver Operating Characteristic (ROC) Curve (AUC) score, sometimes also referred to as ROC-AUC score. Some authors also utilize the log loss and F1-score. The AUC score is a metric to evaluate the performance of a binary classification model which quantifies the ability to distinguish between the two classes. The log loss can be used for binary and multiclass problems and measures the uncertainty of the made prediction. The F1 score is a metric that is especially useful when dealing with imbalanced data and is computed as the harmonic mean between precision and recall of the positive predictions [26–28]. When possible, some authors also validated the approaches in live online AB-tests. Based on the evaluation metrics used in the related work, we will use the AUC and F1 metrics to evaluate our approach. Note that we do not use the log loss for evaluation because it is the loss we use to optimize the learning models in our experiments.

**Table 1.** Overview of publications proposing CTR-P approaches with information on the datasets, evaluation metrics, and scores used.

Author	Year	Approach	Dataset	AUC	Score F1	Logloss
Fan et al. [9]	2022	RACP	Avito Taobao (closed)	0.794 0.7623		
C. Li et al. [20]	2021	Mul-AN	Criteo MovieLens-100k	0.8 0.847		0.483 0.395
X. Li et al. [8]	2020	MARN	Amazon Review Electro Amazon Review Clothing Taobao (closed)	0.803 0.791 0.749		
X. Lie et al. [22]	2020	TIEN	Amazon Review Beauty	0.8701	0.784	0.4479
			Amazon Review Clothing	0.7962	0.698	0.5476
			Amazon Review Grocery	0.8252	0.7524	0.5019
			Amazon Review Phones	0.839	0.7427	0.4949
			Amazon Review Sports	0.8266	0.7543	0.5101

Table 1. Cont.

Author	Year	Approach	Dataset	AUC	Score F1	Logloss
Zeng et al. [29]	2020	USRF	RetailRocket datasets	0.8888	0.8001	
			Amazon Review Digital Music	0.7086	0.6709	
			MovieLense-1M	0.9921	0.8445	
Zhou et al. [10]	2019	DIEN	Amazon Review Electro	0.7792		
			Amazon Review Books	0.8453		
			Taobao	0.6541		
Zhou et al. [21]	2018	DIN	Amazon Review Electro	0.8871		
			MovieLense-20M	0.7348		
			Alibaba (closed)			
Wang et al. [30]	2017	DCN	Criteo			0.4419

## 2.2. Customer Representation

Traditionally, customer behavior is modeled by domain experts to make predictions of their intentions and future behavior. Therefore, data like clickstream data or demographic information are incorporated into the data analysis and feature engineering process [31–35]. As shown by Alves Gomes et al. [36] most customer representations are modeled with manual features extracted by experts or with the RFM analysis [37]. For example, Perisic et al. [38] and Friedrich et al. [39] extracted RFM-based features by extending the RFM analysis from historical data for customer representation. Wu et al. [40] modeled and analyzed customer behavior with an extended RFM approach by adding customer contribution time and repeat purchase attributes and combining it with a k-means clustering. K-means clustering is also used by Hamed Fazlollahtabar [41]. The author chose different customer information gathered from their transactions and applied k-means clustering of different combinations of two features, e.g., gender and product or age and product. Wang et al. [42] analyzed influence factors of second-hand customer-to-customer e-commerce platforms by using questioner and demographic information of customers. Esmeli et al. [35] modeled customers based on twelve features solely based on session information. Berger et al. [43] used features that describe the change in customer behavior based on actual session information and the information retrieved from previous session history. This manual customer representation process is time-consuming and expensive, especially since it needs to be repeated for each new use case or marketing campaign.

Recent approaches that use embedding layers simplify customer modeling by only inserting information into the learning model without a proper feature engineering process. Most of the aforementioned CTR-P approaches utilize embedding layers to learn customer behavior. Sheil et al. [44] proposed an end-to-end three-layered LSTM to predict future customer behavior by learning patterns of the product the customer interacts with, the interaction time, and additional product-related information. Ni et al. [11] proposed a Deep User Perception Network (DUPN) an end-to-end Long-Short Term Memory (LSTM) with an embedding input that is trained on multiple tasks for a general customer representation. Yang et al. [45] and Wu et al. [46] represented customers based on textual features like product names, categories, and reviews written by the customers. However, in addition to using an embedding layer for input data, embeddings can also be used to represent features. Especially in the e-commerce context, embeddings were used in recommendation scenarios. For this purpose, product embeddings were created and trained [47–50]. A recent approach using pre-trained embedding features to represent customer behavior was proposed by Alves Gomes et al. [51,52]. The authors pre-trained an embedding to encode customers' behavior and used the representation to predict customers' purchase intention. In this regard, we adapt the pre-trained embedding representation and an LSTM learning model and show that it is sufficient to predict the CTR of products and recommendations without the need for deep and wide neural networks.

### 3. Use Case and Data Description

In this work, we approach CTR-P for recommendation in an online store scenario. Specifically, while browsing the online shop, customers will get recommendations displayed on the webpage from the online shop's recommendation engine. The question we want to answer is: **Is it possible to predict if the customer will click a recommendation given only their session?** When browsing an online shop, customers do not necessarily have to be logged in, so they are unknown to the operator. In our use case, for example, this concerns about 60% of the recorded events. On the session side, it looks even worse. Of the sessions, 91% start with unknown customers who, on average, only log in after eight interactions. For comparison, sessions are on average twelve interactions long. This showcases the necessity that utilized approaches work for known and unknown customers alike and do not rely on historical customer information.

The use case for predicting the CTR of recommendation is provided by a large US retailer group. The (closed) dataset contains customers' browsing information. Each entry is an event with information about the event time, event type, session id, user id (if known), additional meta information, and the URL of the event. The data were recorded over five months from January 2020 to May 2020. It consists of 53 million customer events in total, with 66,883 different URLs.

The benchmark use case is based on the Amazon review 2018 dataset, which was provided by Ni et al. [53] and is one of the most used datasets to benchmark CTR-P [8,10,22,29,54]. The Amazon review dataset consists of 233.1 million user reviews collected from May 1996 to October 2018 with additional product ratings, product meta information like categories, and images. Similar to [11,15,22], we use the rating information of the selected categories. The data contain tuples, each of which has an item id, a user id, a rating, and a timestamp. Using the item id, additional information can be joined with the help of the metadata.

### 4. Experiments

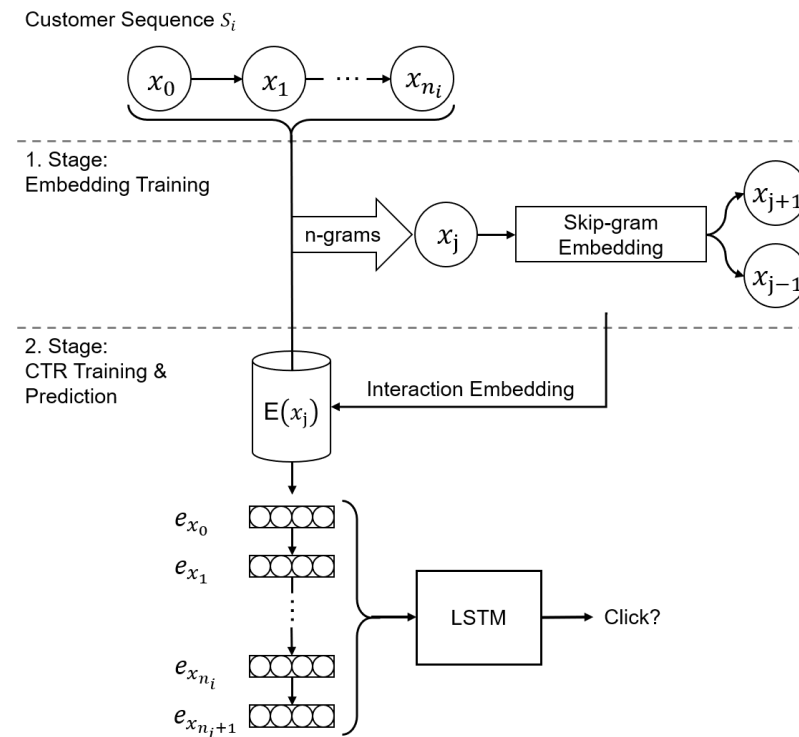
Typically, CTR-P models take a customer behavior sequence  $s_i$  of length  $n$  from customer  $c_i$  with the objective to predict  $P(x_{n+1}|s_i)$ , where  $x_{n+1}$  is a click interaction on the target, i.e., product, ad, or recommendation [8]. Over time, various learning models have been introduced that can be optimized for maximizing  $P(x_{n+1}|s_i)$ , including support vector machines, decision trees, and neural networks. Unfortunately, many of these models are limited to receiving input data of a fixed length. Nevertheless, since sequence  $s_i$  can be of any size, it is necessary to handle these. One option is to utilize padding for sequences that are shorter than the maximum length. Another option is utilizing recurrent neural networks (RNN) that were developed specifically for handling sequential data. The used notations and their descriptions are summarized in Table 2.

**Table 2.** Notation and description.

Notation	Description
$C, X, S$	Set of all customers $C$ , customer interactions $X$ , and sequences $S$
$c_i, x_j$	A customer $c_i \in C$ and interaction $x_j \in X, i, j \in \mathbb{N}$
$s_i, n_i$	$s_i = \{x_0, x_1, \dots, x_n\}, s_i \in S$ is a ascended time-ordered customer behavior sequence with sequence length $n_i, n_i \in \mathbb{N}, n > 1, i \in \mathbb{N}$
$k_j, M$	A interaction $x_j$ of $s_i$ has a context $k_j = \{x_{j+m}, x_{j+m-1}, \dots, x_{j-m+1}, x_{j-m}\} \setminus x_j$ with context window size $M = 2 \times m, m \in \mathbb{N}, m \times 2 \leq n_i, i \in \mathbb{N}$
$e_{x_j}^D$	$D$ -dimensional embedding representation $e_{x_j}$ of interaction $x_j, e_{x_j} \in \mathbb{R}^D$
$E(x_j)$	Embedding function $E$ that uses the trained embedding and maps $x_j \rightarrow e_{x_j}, j \in \mathbb{N}$

#### 4.1. Approach Methodology

As mentioned above, nowadays, deep learning models for CTR-P mostly consist of an embedding layer to process categorical data and are trained end-to-end. We break with this paradigm and propose a two-step approach based on a pre-trained embedding customer interaction representation which encodes all  $x$  in  $s$  and a LSTM binary classifier for CTR-P. Our approach is illustrated in Figure 1. The first step of our approach is to train an embedding customer representation. Therefore, we adapted the “SkipGram” approach by Mikolov et al. [55] to train the interaction embedding representation. Figure 2 shows the embedding architecture when using the SkipGram approach, which consists of one input layer, one hidden layer, and one output layer. The input layer is a one-hot encoding of the customer interaction  $x_j$ . The hidden layer, which is the embedding representation  $e_{x_j}$ , consists of  $D$  neurons. The output layer size is based on the context size and the number of possible customer interactions and has the goal of predicting the context of the customer interaction  $e_{x_j}$ .



**Figure 1.** Our proposed two-stage CTR-P approach.

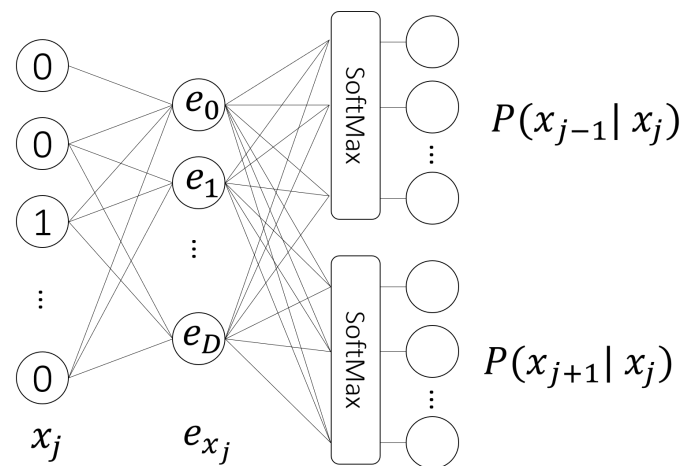
Specifically, a single hidden layer neural network with optimizable variables (weights)  $\theta$  is trained to maximize the likelihood,

$$L(\theta) = \prod_{j=0}^n \prod_{\alpha=-m; \alpha \neq 0}^m P(x_{j+\alpha} | x_j; \theta). \quad (1)$$

Based on the assumption that words with similar contexts are closer to each other in a  $D$  dimensional vector space, word vector representations have been proposed for natural language processing tasks. Our hypothesis is that this assumption is transferable to customer interaction on online platforms. Hence, the similarity between two interactions  $x_a$  and  $x_b$  is defined by the similarity of their context  $k_a$  and  $k_b$ .



After the training of embeddings, the second step of our approach is to train an RNN learning model to predict the probability that interaction  $x_{n+1}$  is a click interaction of customer  $c_i$  given  $s_i$ . One commonly used RNN architecture is the LSTM, which was proposed by Hochreiter et al. [56] in 1997. LSTMs have the ability to carry information from earlier inputs to later inputs by utilizing different kinds of gates and states. In addition to LSTMs, there are also Gated Recurrent Units (GRU) [57] with a similar mode of operation. Both RNN architectures are pre-implemented in various deep learning frameworks. In previously conducted experiments, LSTMs demonstrated superior performance compared to GRUs. Therefore, we opted for LSTMs. All interactions  $x_j \in s_i$  are embedded by  $E(\cdot)$  and are used as input for the LSTM binary classifier.



**Figure 2.** SkipGram embedding architecture as proposed by Mikolov et al. [55].

#### 4.2. Baseline Approaches

Based on our literature research presented in Section 2, we identified four state-of-the-art CTR-P approaches that are applicable to our use case, as they can handle customer behavior sequences. Our aim was to compare our approach with these four baseline models. Unfortunately, we were unable to replicate the outcomes of MARN and TIEN due to the unavailability of their source code and the inadequate amount of information provided in their original papers for recreating the approaches. We therefore decided to use only DIN and DIEN as baseline models. At this point, it is worth noting again that both MARN and TIEN are extensions of the DIN approach and do not follow a completely new concept. Therefore, DIN yields similar outcomes in terms of approach.

1. **LSTM baseline:** The LSTM baseline approach is similar to our approach, but is designed as an end-to-end model, lacking the embedding decoupling and thus the self-supervised pre-training of the customer behavior representation embedding.
2. **DIN:** Zhou et al. [21] proposed the Deep Interest Network (DIN) for CTR-P with the idea that the model captures user interests in past user interactions. This is realized with an attention mechanism that refers to the target item.
3. **DIEN:** Zhou et al. [10] proposed Deep Interest Evolution Network (DIEN) as the successor of DIN. It has a similar motivation to capture historical user interest but uses a different approach and model architecture. DIEN consists of three layers; (1) a Behavior Layer, (2) an Interest Extractor Layer, and (3) an Interest Evolving Layer. The Behavior Layer is the embedding layer that processes customers' historical sequence. The second layer consists of gated recurrent units (GRU) [57]. The third layer consists of an attention mechanism and AUGRU, a GRU with an attentional update gate.

### 4.3. Data Preprocessing

First, we prepared the datasets for further usage. As mentioned previously, the closed dataset consists of customers' browsing events. For each event, we defined an interaction  $x_i$  as a string of "event type:URL". For example, an "add to cart" event of the URL "www.123.abc" would result in "cart:www.123.abc". Within the dataset, six different event types exist, which are *add to cart*, *remove from cart*, *check out*, *page view*, *product view*, and *recommendation click*. In the next step, we removed the query strings from the URLs that are unique to each customer. Otherwise, they would over-specify the URL, resulting in many unique URLs and making it difficult to generalize the trained representation. Thereafter, we aggregated all events by their session id to customer behavior sessions and removed the session with less than three interactions as personalized prediction is not feasible in a real-world scenario with only one or two customer interactions. We labeled each session according to the information if at least one *recommendation click* event exists or not. Finally, we removed all click recommendation interactions (training objective) and customer interactions that happened thereafter. The resulting set of sessions is highly imbalanced with only 1.5% being positively labeled. Therefore, we applied a random undersampling (3:2) strategy to generate a balanced dataset for training and evaluation.

Regarding the baseline Amazon review dataset, we created two datasets. In the literature, due to the vast amount of data, the authors usually select one category at a time and predict the CTR only for that category, as shown in Table 1. Similarly, we decided to select the "Clothing" subset as a dataset, which was used to evaluate the not openly available approaches MARN and TIEN. Thus, we can get an indication of how our approach compares to these two approaches, even if we are not able to include them in our experiments.

Instead of selecting other categories individually, we decided to combine the subsets of the categories used by X. Li et al. [22], "Clothing", "Beauty", "Grocery", "Phones", and "Sports" into one dataset. In the following, we refer to this dataset as "Amazon 5 categories". There were two major reasons to do so: (1) While investigating the dataset, we noticed that customers have interactions between different categories and therefore, we suspect that neglecting this might not reflect the reality of customer behavior; (2) we want to demonstrate that large amounts of data are not a limiting factor for the presented approach.

For the following dataset preprocessing steps, we adhered to previous works from [11,15,22]. The Amazon dataset consists of rating quadruples with item id, user id, item rating, and timestamp information. First, we removed all duplicate tuples and all products and users that occurred less than five times. Then we built all user behavior sequences by aggregating the user id and sorting them by the timestamp. Here, rated products are treated as clicked product interactions regardless of the user rating. Note that only positive examples exist. In order to create negative examples, we exchanged the last product in 50% of all user behavior sequences with a product that is not in the behavior sequence.

Following the works of [10,22], we split all datasets into 85% training data and 15% testing data. In order to avoid feature leakage, we ensured that all training sequences  $S_{train}$  take place before the testing sequences  $S_{test}$ .  $S_{test}$  is used for evaluation only. For our pre-trained embedding approach, we used a context window size  $M = 2$  and therefore, we create trigrams out of all sequences  $s \in S_{train}$ . Specifically, for each interaction  $x_i$ , a triple  $(x_i, x_{i-1}, x_{i+1})$  is formed. For  $i = 0 \vee i = n$ , with  $x_n$  the last interaction in a sequence, we introduced a "START" and a "END" token in order to not neglect the first and last interaction. New products are frequently introduced and the embedding needs to be able to deal with them. In our experiments, this is represented by interactions that are not included in the training set but in the test set. Therefore, we introduced the "unknown" token which is one way to deal with the out-of-vocabulary problem [51,58]. Note that the created negative examples of the Amazon training set are not included in the trigrams. The statistics of the three datasets after preprocessing as well as for the created training and test sets, are shown in Table 3.



**Table 3.** Dataset statistics with meta-information about the number of customers and interactions as well as information about training sessions, test sessions, and n-grams.

	Amazon Clothing	Amazon 5 Categories	Closed
#interactions	10,714,172	25,862,230	53,825,295
#users	1,164,752	2,547,663	574,890
#sequences	1,164,752	2,547,663	6,195,916
#unique interactions	372,593	801,890	66,891
#train samples	970,717	2,127,165	119,905
$\varnothing$ train sequence length	8.6689	9.5863	12.6557
#test samples	171,406	376,606	21,160
$\varnothing$ test sequence length	7.5048	7.8988	11.6173
#n-grams	7,452,387	18,742,161	3,849,627

#### 4.4. Reproduction of the Experiments

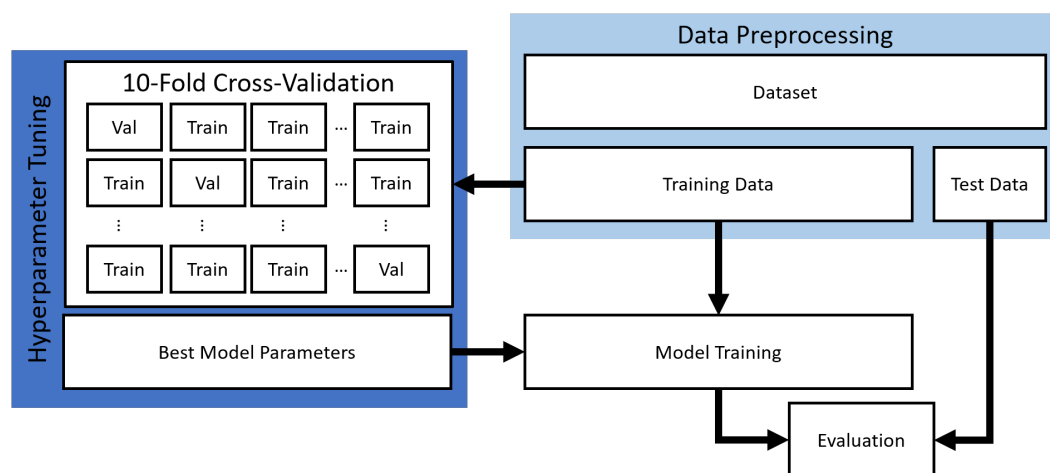
For the sake of the reproducibility of our experiments, we describe the necessary information in this section. The experiments were implemented in Python 3.8.10 [59] with NumPy 1.20 [60] and Pandas 1.3 [61] for the data preprocessing of the three datasets. We implemented our approach (embedding and LSTM) with PyTorch 1.11 [62] and for DIN and DIEN we used DeepCTR-Torch 0.2.9 [63]. For the evaluation metrics, we used implementations from scikit-learn 1.1.1 [64,65]. Optuna 2.10 [66] was used for hyperparameter tuning which uses different strategies like grid search, random, bayesian, and evolutionary algorithms. The data preprocessing and model validation were computed on a Windows 10 machine with an Intel i9-10885H and 64GB RAM. Model training and hyperparameter search was computed on an Ubuntu 18.04 machine with 96xIntel Xeon Platinum 8168 CPU @ 2.70GHz, 756GB RAM, and 8xNvidia Tesla V100 GPUs.

We trained each approach on each dataset and, additionally, conducted a hyperparameter search for all approaches. Figure 3 illustrates our conducted experiments. First, the data are preprocessed, as mentioned in Section 4.3, and are split into training and test data of which all the test data are in the future of the training data split. This is followed by a hyperparameter tuning process for each evaluated approach in our work. For the search, we employed 10-fold cross-validation, randomly selecting 10% of the training data as validation for each fold. Specifically, for 10-fold cross-validation, the training data were divided into ten equally sized pieces, also referred to as folds, whereby the subdivision is random. In each validation step, the model is trained with nine folds and the remaining fold is used for validation. This is repeated ten times with different folds each time and the validation results of each step are then averaged. The hyperparameters chosen by the hyperparameter search are described below. We always trained with a batch size of 128. For the embedding dimension, the hyperparameter search had the search space  $2^n, 2 \leq n \leq 10$ . The chosen dimension for our proposed embedding representation is 64 for both Amazon datasets and 32 for the closed dataset. For the embedding training, “AdamW” was selected as the optimizer and initialized with a learning rate of 0.001. As a loss function, the “CrossEntropyLoss” was selected, which combines the SoftMax activation and the negative log-likelihood. The embeddings are trained with early-stopping of the loss and it stops when the loss does not change more than  $10^{-4}$  after five epochs which happened around training 60 epochs. For the CTR-P, a one-layer LSTM was selected by the search algorithm which had the option to select between one to ten LSTM layers for the model. Another hyperparameter for the LSTM is the dimension for the hidden size. The search space for the hidden size is similar to the embedding dimension defined by  $2^n, 2 \leq n \leq 10$ . It turned out that the best hidden dimension always had the same shape as the embedding dimension. Also for the LSTM we utilized the “AdamW” optimizer with a 0.001 learning rate and “BCEWithLogitLoss” as a loss function. The best hyperparameters for the LSTM baseline approach are the same as our approach. As mentioned above, for our approach, only the sequence of customer interactions was used as features which were only the rated

products for both Amazon datasets and the combination “event type:URL” for the closed dataset. The same features were also used for the LSTM baseline model.

Regarding DIN and DIEN more features are used. For the two Amazon datasets, we utilized the history of rated products and their categories. Similarly, for the closed dataset, we used the sequence URLs and the sequence event type as input features. The features are each encoded with a 32-dimensional embedding layer for each dataset. Furthermore, for both models, a dnn-dropout of 50% leads to the best results. As the optimizer, “adagrad” was selected and “binary\_crossentropy” was selected as the loss function. Additionally for DIEN, we used “AUGRU” as the parameter for the gru-type. Note that, for unnamed hyperparameters, the default values from the libraries were the best option based on the conducted hyperparameter search.

After having the best hyperparameter for each approach, we trained the models on the whole training dataset. Thereafter, we ran ten training runs, each with random initialization, to eliminate any favorable starting conditions that could affect our results. Each tested approach was trained for 150 epochs and we recorded the performance and the loss on the training data. For evaluation, we utilized the test data and we always chose the checkpoint of the best-average-performance of the model.



**Figure 3.** Flow chart of the conducted experiments including data preprocessing, hyperparameter tuning with cross-validation, model training, and evaluation.

## 5. Results and Discussion

Table 4 shows the resulting AUC and F1 scores of our experiments. The LSTM baseline has the lowest AUC and F1 score among all approaches for both Amazon datasets. Surprisingly, DIN has a slightly higher AUC score than DIEN for both experiments with the Amazon datasets, which contradicts the results of Zhou et al. [10]. For the Amazon Clothing dataset, DEIN outperforms DIN with an F1 score that is 0.059 points higher. Compared to the experiments conducted on the Amazon datasets, each approach’s experimental results on the closed dataset yielded higher scores. Also, with regard to our (closed) use case, DIEN is outperformed by DIN. For our use case, DIEN is the worst-performing approach tested. The LSTM baseline approach slightly outperforms DIN. Our approach outperforms all three baseline approaches on each of the three datasets. In the experiments for our use case, our approach has a 0.0079 higher AUC and a 0.0063 higher F1 score than the LSTM baseline and they are even higher when compared to the two state-of-the-art models. With regard to the two Amazon datasets, our approach outperforms the three baseline approaches by about 0.1 AUC and 0.07 F1. Shi et al. stated that, in CTR-P tasks, “a slightly higher AUC or Logloss value at 0.001-level is regarded as a huge improvement” [67].

**Table 4.** Experiment results for the three baselines and our approach on each dataset. Approaches were trained ten times with a random initialization each time. Scores are averages of the ten trained model scores. The standard deviation for all results is around  $10^{-3}$ . The best results for each dataset and score are shown in bold.

Approach	Amazon Clothing		Amazon 5 Categories		Closed	
	AUC	F1	AUC	F1	AUC	F1
LSTM baseline	0.7651	0.7071	0.7712	0.7106	0.9731	0.9387
DIN	0.7885	0.7221	0.7948	0.725	0.9626	0.9329
DIEN	0.7796	0.7280	0.7759	0.7246	0.9510	0.9251
ours	<b>0.8851</b>	<b>0.7962</b>	<b>0.8896</b>	<b>0.7996</b>	<b>0.9810</b>	<b>0.9450</b>

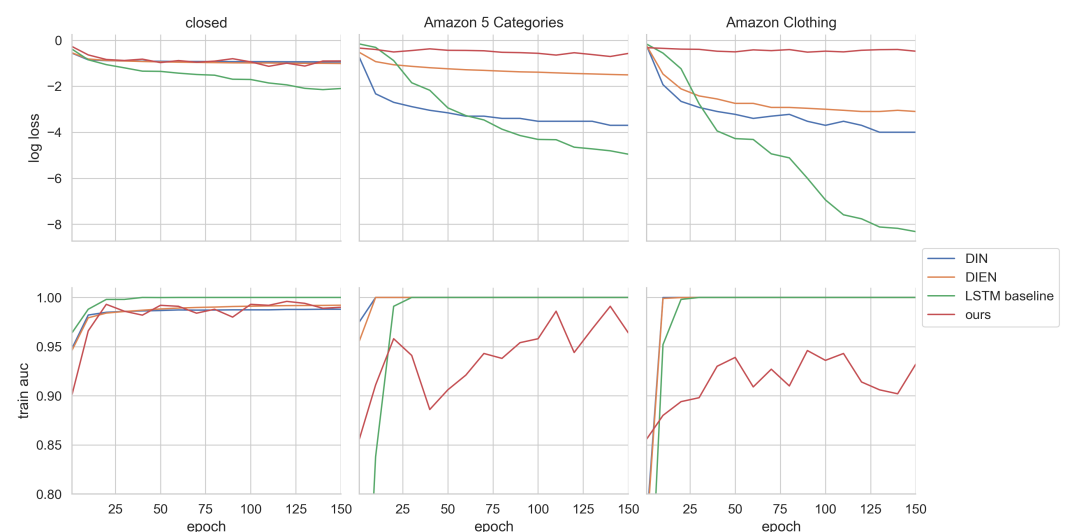
The results show that our method is suitable for predicting whether a customer will click on a recommendation during a session. Thus, we can confidently answer our initial research question, that **it is possible to predict a customer's click on a recommendation given the session**. Additionally, our experimental results on the two Amazon datasets demonstrate that our approach has the capability to predict CTR based on customer interaction history, surpassing previously proposed state-of-the-art approaches.

However, based on our literature review, TIEN [22] and MARN [8] are both more recent approaches for the CTR-P of customer behavior sequences that outperform DIEN. Unfortunately, in contrast to DIN and DIEN, the models were, as discussed previously, not accessible for use. Nevertheless, in both publications, the authors compare their approach against DIEN on the Amazon clothing subset, which helps us validate our results and compare them to their approaches. In the work of X. Li et al. [22], DIEN has an AUC score of 0.7564 and an F1 score of 0.6792, which are lower than our results for DIEN. TIEN has an AUC score of 0.7962, which is 0.0398 higher, and an F1 score of 0.698, which is 0.0188 higher than DIEN's scores. In the MARN publication by X. Li et al. [8], the DIEN AUC score on Amazon clothing is 0.7793, which is approximately the same AUC score that DIEN archives in our experiment. MARN archives an AUC score of 0.7909 on the Amazon clothing dataset, which is 0.0116 higher than DIEN's AUC score. In our experiments, the AUC score of our approach is 0.1055 higher and the F1 score is 0.0682 higher than DIEN's scores, which provides confidence that our approach would also outperform both TIEN and MARN.

Why does our approach outperform the two state-of-the-art models DIN and DIEN, as well as the LSTM baseline, even though it is structurally almost identical? The cause is likely the pre-trained embedding of customer interaction representation. We train the embeddings task independently in a self-supervised manner and, thus, the embedding was able to learn similarities between customer interactions from the data and the context. In contrast, an embedding input layer learns task-specific patterns and not necessarily the context of the interactions and their meaning. From this, we deduce that the quality of our pre-trained embedding features is significantly better for learning since only the LSTM cells need to be adjusted for CTR-P during training. With the other approaches, several layers have to be adjusted, which for example can lead to the vanishing gradient problem. Evidence is given by inspecting the training loss and score. Figure 4 shows the average log loss and AUC score of the approaches' CTR-P training. Loss-wise for the Amazon datasets, it can clearly be seen that our approach never reaches a log loss under  $-1$ . In fact, it stays almost constant. For the baseline approaches the loss is getting smaller, especially for the LSTM baseline, which can be interpreted as overfitting on the train data. This assumption is supported by the achieved AUC scores of the baseline models. All baseline models reach an AUC score of one after 25 epochs of training. On the other side, our approach never reaches such a high AUC score during training which speaks for a better generalization of the predictor. In the case of the closed dataset, no major difference between the approaches is seen between the loss and the AUC score. This is also evident in the results for the closed dataset.

Our results show that large, deep, and wide learning models are not always suitable for all use cases. Especially in e-commerce—which has a highly dynamic environment with ever-changing trends as well as customer preferences—smaller and easier-to-implement models pose to be superior at the moment. In fact, the LSTM baseline approach is only slightly worse on the Amazon dataset and slightly better on the closed dataset. This raises the question of whether it is suitable to construct novel architectures incorporating attention and other contemporary deep learning techniques that introduce supplementary trainable parameters to a model, thereby intensifying its training. Further research is necessary to shed more light on this matter and we encourage the community to take this alternative perspective of investigating alternative approaches to simply building larger and more complex models when trying to tackle e-commerce problems such as CTR-P.

In the last part, we want to discuss our approach's limitations and implications. One limitation is that, in contrast to DIN, DIEN, and other approaches, incorporating new information is not a straightforward process. Consequently, it becomes more difficult to take into account all available information, such as age, gender, and time. In this regard, further research is needed. A review of the literature reveals that there have been early attempts to propose product embeddings that encode additional data for embedding-based recommender systems, like Meta-prod2vec [47] or SkipGramEx [49]. However, it should be noted that personalized information is becoming increasingly difficult to obtain, for example, due to government restrictions and laws [68,69], which is why the pure use of interaction data represents a solution to this problem. Another challenge of our approach, but also of other approaches such as DIN and DIEN, is the interpretability. Traditional approaches that use handcrafted features can be understood and explained by domain experts. End-to-end deep learning approaches or our approach are difficult to explain due to the black-box characteristics of neural networks. With the emergence of the research field of explainable AI (XAI), scientists worldwide are making efforts to understand the decision-making of neural networks and to improve their interpretability and explainability.



**Figure 4.** Averaged train history of all four approaches on the three datasets for 150 epochs of training. The first row displays the log loss to the tenth power and the second row the AUC score.

Our research indicates that the ongoing trend toward using large end-to-end models in various tasks should be re-evaluated. While these models may be revolutionary in many regards and change the way tasks are solved, they also require substantial amounts of resources which should always be considered. By examining the case of CTR-P in the e-commerce sector, we showed that such large models may not always be indispensable, and we look forward to promoting this awareness across other fields. For industrial applications, this shows that it is not always necessary to use the largest and most advanced models

from large technology companies, but that smaller and easier to implement approaches are sufficient and can even outperform these large models.

Moreover, it is worth highlighting the broad spectrum of data used by large models, including demographic customer data. Collecting such data is not always an effortless task, and regulations, as already discussed, need to be accounted for. Our approach refrains from using demographic and historical data, making it mostly conventional especially for regions with high data privacy requirements. In addition, by requiring less private customer information it makes the approach easier to transfer to other use cases.

Another aspect to consider in our study is how it all plays out for the customer. We are questioning whether ramping up personalization in e-commerce genuinely enhances the customer experience. Is it a positive experience because it makes shopping more enjoyable, or is there a flip side where customers might be nudged into actions that are not in their interest? Exploring these effects in more detail in future research will help us figure out if any regulations are needed down the line.

## 6. Summary and Outlook

CTR-P is a significant subject for both academia and industry. Our study predicted clicks on recommendations solely from sessions where customers are not necessarily logged in. Recent research demonstrates that end-to-end models can achieve satisfactory results in addressing this task. However, end-to-end models are becoming larger and utilizing more complex architectures, resulting in more challenging training due to the increase in trainable parameters. Additionally, the need for more computation resources is a growing concern not only in terms of financial costs but also in light of rising energy costs in our modern world. We proposed a decoupled CTR-P model and were able to show that wide and deep models are not necessarily needed to predict customer clicks. Our approach consists of a self-supervised pre-trained customer behavior embedding and a simple one-layer LSTM. We performed the experiments on our use case with our own closed dataset and furthermore, to demonstrate the transferability of our approach, we also conducted the experiments on an open-access benchmark dataset for CTR-P. For our use case, our approach has a 0.0079 higher AUC and 0.0063 F1 score than the second-best approach tested. For the benchmark dataset, our approach outperforms the three baseline approaches by about 0.1 AUC and 0.07 F1 score. Based on the experiment's results, we have reason to assume that the task-independent self-supervised pre-training embedded customer behavior is the key component of our approach's success. By investigating the training history, we showed that our approach provides evidence of better generalization compared to the baseline approaches and is presumably less susceptible to overfitting due to having less trainable parameters. Therefore, we can conclude that wide and deep models are not necessary to approach CTR-P.

In future research, we aim to investigate the effect of self-supervised pre-training on e-commerce task performance. Based on our findings it is possible that the self-supervised pre-training of the customer behavior context for our embeddings has also a positive impact on the end-to-end approaches. This way of approaching tasks by pre-trained models is already best practice for language and vision problems in which large language models are pre-trained on a huge amount of data and then fine-tuned for a specific use case. Furthermore, we aim to explore methods for integrating further interaction data, such as the time of each interaction, into the embedding. We anticipate that enhancing the customer representation quality through this approach will lead to increased predictive capabilities of the learning model.

From the literature, we know that similar embedding representation approaches were already used for product recommendation and purchase prediction. Therefore, we want to investigate how our approach is transferable to other e-commerce tasks. We assume that the approach if successful on other e-commerce tasks can have an impact on how to approach them. However, it is necessary to find ways to explain the embeddings so that

this transferability to different prediction tasks can also be justified without conducting experiments to show if it works.

In a real-world application, it is beneficial to comprehend the approach's decision-making. As mentioned above, neural networks-based approaches have the issue of being black boxes. Therefore, an understanding of its decisions a priori is not possible and requires extensive analysis of the data, training process, and prediction. Therefore, further research towards explainable AI is necessary, which needs to be undertaken for our approach and end-to-end approaches alike. We plan to approach explainability with visualization approaches and ablation studies in the future and hope this will promote the acceptance of new technologies among skeptics.

Currently, for our use case, we only consider information on the ongoing session. In the future, we want to investigate how the approach performs if using more information like historical sessions for known customers. Since this increases the information density of the input sequence, it would be necessary to investigate when the learning model needs more parameters. On the one hand, this can be tested with hyperparameter tuning, but it would be much more interesting if it could be measured to what extent more trainable parameters are necessary to achieve the optimal result. It is possible that, with longer sequences, the large end-to-end models perform better and we would have to extend the LSTM or the embeddings so that more patterns can be learned from the sequences.

**Author Contributions:** M.A.G.: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data Curation, Writing—Original Draft, Writing—Review & Editing, Visualization, Project administration. R.M.: Validation, Writing—Review & Editing, Supervision. P.M.: Validation, Data Curation, Writing—Review & Editing, Funding acquisition. T.M.: Validation, Writing—Review & Editing, Supervision. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The authors used the Amazon Review Dataset that can be found at <https://nijianmo.github.io/amazon/index.html> (accessed on 26 September 2023). The authors have no right to make the “closed” dataset publicly available.

**Conflicts of Interest:** The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## References

1. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is All you Need. In *Advances in Neural Information Processing Systems*; Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2017; Volume 30.
2. Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*; Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2020; Volume 33, pp. 1877–1901.
3. Ramesh, A.; Pavlov, M.; Goh, G.; Gray, S.; Voss, C.; Radford, A.; Chen, M.; Sutskever, I. Zero-Shot Text-to-Image Generation. In *Proceedings of the 38th International Conference on Machine Learning, PMLR, Virtual, 18–24 July 2021*; Volume 139, pp. 8821–8831.
4. Cheng, H.T.; Koc, L.; Harmsen, J.; Shaked, T.; Chandra, T.; Aradhye, H.; Anderson, G.; Corrado, G.; Chai, W.; Ispir, M.; et al. Wide & Deep Learning for Recommender Systems. In *Proceedings of the DLRS 2016 1st Workshop on Deep Learning for Recommender Systems, Boston, MA, USA, 15 September 2016*; pp. 7–10.
5. Sun, F.; Liu, J.; Wu, J.; Pei, C.; Lin, X.; Ou, W.; Jiang, P. BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer. In *Proceedings of the CIKM '19 28th ACM International Conference on Information and Knowledge Management, Beijing, China, 3–7 November 2019*; pp. 1441–1450. [CrossRef]
6. Huang, G.; Chen, Q.; Deng, C. A New Click-Through Rates Prediction Model Based on Deep&Cross Network. *Algorithms* **2020**, *13*, 342. [CrossRef]
7. Xia, Z.; Mao, S.; Bai, J.; Geng, X.; Yi, L. A Novel Integrated Network with LightGBM for Click-Through Rate Prediction. *Res. Sq.* 2021, preprint. [CrossRef]



8. Li, X.; Wang, C.; Tan, J.; Zeng, X.; Ou, D.; Zheng, B. Adversarial Multimodal Representation Learning for Click-Through Rate Prediction. In Proceedings of the Web Conference 2020, Taipei, Taiwan, 20–24 April 2020; pp. 827–836. [\[CrossRef\]](#)
9. Fan, Z.; Ou, D.; Gu, Y.; Fu, B.; Li, X.; Bao, W.; Dai, X.Y.; Zeng, X.; Zhuang, T.; Liu, Q. Modeling Users' Contextualized Page-wise Feedback for Click-Through Rate Prediction in E-commerce Search. In Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining, Virtual, 21–25 February 2022; pp. 262–270. [\[CrossRef\]](#)
10. Zhou, G.; Mou, N.; Fan, Y.; Pi, Q.; Bian, W.; Zhou, C.; Zhu, X.; Gai, K. Deep Interest Evolution Network for Click-Through Rate Prediction. *Proc. AAAI Conf. Artif. Intell.* **2019**, *33*, 5941–5948. [\[CrossRef\]](#)
11. Ni, Y.; Ou, D.; Liu, S.; Li, X.; Ou, W.; Zeng, A.; Si, L. Perceive Your Users in Depth: Learning Universal User Representations from Multiple E-Commerce Tasks. In Proceedings of the KDD '18 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, London, UK, 19–23 August 2018; pp. 596–605. [\[CrossRef\]](#)
12. Carmel, D.; Haramaty, E.; Lazerson, A.; Lewin-Eytan, L. Multi-Objective Ranking Optimization for Product Search Using Stochastic Label Aggregation. In Proceedings of the WWW '20 Web Conference 2020, Taipei, Taiwan, 20–24 April 2020; pp. 373–383. [\[CrossRef\]](#)
13. Li, F.; Chen, Z.; Wang, P.; Ren, Y.; Zhang, D.; Zhu, X. Graph Intention Network for Click-through Rate Prediction in Sponsored Search. In Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, Paris, France, 21–25 July 2019; pp. 961–964. [\[CrossRef\]](#)
14. Pan, Z.; Chen, E.; Liu, Q.; Xu, T.; Ma, H.; Lin, H. Sparse Factorization Machines for Click-Through Rate Prediction. In Proceedings of the 2016 IEEE 16th International Conference on Data Mining (ICDM), Barcelona, Spain, 12–15 December 2016; pp. 400–409. [\[CrossRef\]](#)
15. Ren, K.; Zhang, W.; Rong, Y.; Zhang, H.; Yu, Y.; Wang, J. User Response Learning for Directly Optimizing Campaign Performance in Display Advertising. In Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, Indianapolis, IN, USA, 24–28 October 2016; pp. 679–688. [\[CrossRef\]](#)
16. Kumar, V.; Venkatesan, R.; Reinartz, W. Performance Implications of Adopting a Customer-Focused Sales Campaign. *J. Mark.* **2008**, *72*, 50–68. [\[CrossRef\]](#)
17. Chen, C.; Chen, H.; Zhao, K.; Zhou, J.; He, L.; Deng, H.; Xu, J.; Zheng, B.; Zhang, Y.; Xing, C. EXTR: Click-Through Rate Prediction with Externalities in E-Commerce Sponsored Search. In Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, 14–18 August 2022; pp. 2732–2740. [\[CrossRef\]](#)
18. Ge, T.; Zhao, L.; Zhou, G.; Chen, K.; Liu, S.; Yi, H.; Hu, Z.; Liu, B.; Sun, P.; Liu, H.; et al. Image Matters: Visually Modeling User Behaviors Using Advanced Model Server. In Proceedings of the 27th ACM International Conference on Information and Knowledge Management, Torino, Italy, 22–26 October 2018; pp. 2087–2095. [\[CrossRef\]](#)
19. Gulhane, P.R.; Kumar, T.S.P. TensorFlow Based Website Click through Rate (CTR) Prediction Using Heat maps. In Proceedings of the 2018 International Conference on Recent Trends in Advance Computing (ICRTAC), Chennai, India, 10–11 September 2018; pp. 97–102. [\[CrossRef\]](#)
20. Li, C.; Yi, K.; Fei, M.; Zhou, W.; Wu, X.; Chen, Y. Multiple-structure Attentional Network for Click-through Prediction in Recommendation System. In Proceedings of the 2021 IEEE International Conference on Recent Advances in Systems Science and Engineering (RASSE), Shanghai, China, 12–14 December 2021; pp. 1–6. [\[CrossRef\]](#)
21. Zhou, G.; Zhu, X.; Song, C.; Fan, Y.; Zhu, H.; Ma, X.; Yan, Y.; Jin, J.; Li, H.; Gai, K. Deep Interest Network for Click-Through Rate Prediction. In Proceedings of the KDD '18 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, London, UK, 19–23 August 2018; pp. 1059–1068. [\[CrossRef\]](#)
22. Li, X.; Wang, C.; Tong, B.; Tan, J.; Zeng, X.; Zhuang, T. Deep Time-Aware Item Evolution Network for Click-Through Rate Prediction. In Proceedings of the 29th ACM International Conference on Information & Knowledge Management, Virtual, 19–23 October 2020; pp. 785–794. [\[CrossRef\]](#)
23. Wang, F.; Zhao, L. A Hybrid Model for Commercial Brand Marketing Prediction Based on Multiple Features with Image Processing. *Secur. Commun. Netw.* **2022**, *2022*, 5455745. [\[CrossRef\]](#)
24. Wong, C.M.; Feng, F.; Zhang, W.; Vong, C.M.; Chen, H.; Zhang, Y.; He, P.; Chen, H.; Zhao, K.; Chen, H. Improving Conversational Recommender System by Pretraining Billion-scale Knowledge Graph. In Proceedings of the 2021 IEEE 37th International Conference on Data Engineering (ICDE), Chania, Greece, 19–22 April 2021; pp. 2607–2612. [\[CrossRef\]](#)
25. Yao, S.; Tan, J.; Chen, X.; Yang, K.; Xiao, R.; Deng, H.; Wan, X. Learning a Product Relevance Model from Click-Through Data in E-Commerce. In Proceedings of the Web Conference 2021, Online, 19–23 April 2021; pp. 2890–2899. [\[CrossRef\]](#)
26. Rosasco, L.; De Vito, E.; Caponnetto, A.; Piana, M.; Verri, A. Are loss functions all the same? *Neural Comput.* **2004**, *16*, 1063–1076. [\[CrossRef\]](#) [\[PubMed\]](#)
27. Fawcett, T. An introduction to ROC analysis. *Pattern Recognit. Lett.* **2006**, *27*, 861–874. [\[CrossRef\]](#)
28. Sasaki, Y. The truth of the F-measure. *Teach Tutor Mater* **2007**, *1*, 1–5.
29. Zeng, J.; Chen, Y.; Zhu, H.; Tian, F.; Miao, K.; Liu, Y.; Zheng, Q. User Sequential Behavior Classification for Click-Through Rate Prediction. In Proceedings of the Database Systems for Advanced Applications, DASFAA 2020 International Workshops: BDMS, SeCoP, BDQM, GDMA, and AIDE, Jeju, Republic of Korea, 24–27 September 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 267–280. [\[CrossRef\]](#)
30. Wang, R.; Fu, B.; Fu, G.; Wang, M. Deep & Cross Network for Ad Click Predictions. In Proceedings of the ADKDD'17, Halifax, NS, Canada, 14 August 2017. [\[CrossRef\]](#)

31. Sismeiro, C.; Bucklin, R.E. Modeling purchase behavior at an e-commerce web site: A task-completion approach. *J. Mark. Res.* **2004**, *41*, 306–323. [\[CrossRef\]](#)
32. Romov, P.; Sokolov, E. RecSys Challenge 2015: Ensemble Learning with Categorical Features. In Proceedings of the RecSys '15 Challenge: 2015 International ACM Recommender Systems Challenge, Vienna, Austria, 16–20 September 2015. [\[CrossRef\]](#)
33. Li, Q.; Gu, M.; Zhou, K.; Sun, X. Multi-Classes Feature Engineering with Sliding Window for Purchase Prediction in Mobile Commerce. In Proceedings of the 2015 IEEE International Conference on Data Mining Workshop (ICDMW), Atlantic City, NJ, USA, 14–17 November 2015; pp. 1048–1054. [\[CrossRef\]](#)
34. Martínez, A.; Schmuck, C.; Pereverzyev, S.; Pirker, C.; Haltmeier, M. A machine learning framework for customer purchase prediction in the non-contractual setting. *Eur. J. Oper. Res.* **2020**, *281*, 588–596. [\[CrossRef\]](#)
35. Esmeli, R.; Bader-El-Den, M.; Abdullahi, H. Towards early purchase intention prediction in online session based retailing systems. *Electron. Mark.* **2021**, *31*, 697–715. [\[CrossRef\]](#)
36. Alves Gomes, M.; Meisen, T. A review on customer segmentation methods for personalized customer targeting in e-commerce use cases. *Inf. Syst. e-Bus. Manag.* **2023**, *21*, 527–570. [\[CrossRef\]](#)
37. Hughes, A.M. *Strategic Database Marketing: The Masterplan for Starting and Managing a Profitable, Customer-Based Marketing Program*; Irwin Professional: Burr Ridge, IL, USA, 1994.
38. Perišić, A.; Pahor, M. RFM-LIR Feature Framework for Churn Prediction in the Mobile Games Market. *IEEE Trans. Games* **2022**, *14*, 126–137. [\[CrossRef\]](#)
39. Fridrich, M.; Dostál, P. User Churn Model in E-Commerce Retail. *Sci. Pap. Univ. Pardubic. Ser. D Fac. Econ. Adm.* **2022**, *30*. [\[CrossRef\]](#)
40. Wu, J.; Shi, L.; Yang, L.; Niu, X.; Li, Y.; Cui, X.; Tsai, S.B.; Zhang, Y. User value identification based on improved RFM model and k-means++ algorithm for complex data analysis. *Wirel. Commun. Mob. Comput.* **2021**, *2021*, 9982484. [\[CrossRef\]](#)
41. Fazlollahab, H. Intelligent marketing decision model based on customer behavior using integrated possibility theory and K-means algorithm. *J. Intell. Manag. Decis.* **2022**, *1*, 88–96. [\[CrossRef\]](#)
42. Wang, L.; Sun, H. Influencing Factors of Second-Hand Platform Trading in C2C E-commerce. *J. Intell. Manag. Decis.* **2023**, *2*, 21–29. [\[CrossRef\]](#)
43. Berger, P.; Kompan, M. User Modeling for Churn Prediction in E-Commerce. *IEEE Intell. Syst.* **2019**, *34*, 44–52. [\[CrossRef\]](#)
44. Sheil, H.; Rana, O.; Reilly, R. Predicting purchasing intent: Automatic feature learning using recurrent neural networks. *arXiv* **2018**, arXiv:1807.08207.
45. Yang, B.; Liu, K.; Xu, X.; Xu, R.; Liu, H.; Xu, H. Learning Universal User Representations via Self-Supervised Lifelong Behaviors Modeling. In Proceedings of the ICLR 2022 Conference, Virtual, 25–29 April 2022.
46. Wu, C.; Wu, F.; Qi, T.; Lian, J.; Huang, Y.; Xie, X. Ptum: Pre-training user model from unlabeled user behaviors via self-supervision. *arXiv* **2020**, arXiv:2010.01494.
47. Vasile, F.; Smirnova, E.; Conneau, A. Meta-Prod2Vec: Product Embeddings Using Side-Information for Recommendation. In Proceedings of the RecSys '16: 10th ACM Conference on Recommender Systems, Boston, MA, USA, 15–19 September 2016; pp. 225–232.
48. Tercan, H.; Bitter, C.; Bodnar, T.; Meisen, P.; Meisen, T. Evaluating a Session-based Recommender System using Prod2vec in a Commercial Application. In Proceedings of the 23rd International Conference on Enterprise Information Systems, Virtual, 26–28 April 2021; SciTePress: Setúbal, Portugal, 2021; Volume 1, pp. 610–617. [\[CrossRef\]](#)
49. Alves Gomes, M.; Tercan, H.; Bodnar, T.; Meisen, P.; Meisen, T. A Filter is Better Than None: Improving Deep Learning-Based Product Recommendation Models by Using a User Preference Filter. In Proceedings of the 2021 IEEE 23rd Int Conf on High Performance Computing & Communications; 7th Int Conf on Data Science & Systems; 19th Int Conf on Smart City; 7th Int Conf on Dependability in Sensor, Cloud & Big Data Systems & Application (HPCC/DSS/SmartCity/DependSys), Haikou, China, 20–22 December 2021; pp. 1278–1285. [\[CrossRef\]](#)
50. Srilakshmi, M.; Chowdhury, G.; Sarkar, S. Two-stage system using item features for next-item recommendation. *Intell. Syst. Appl.* **2022**, *14*, 200070. : 10.1016/j.iswa.2022.200070 [\[CrossRef\]](#)
51. Alves Gomes, M.; Meyes, R.; Meisen, P.; Meisen, T. Will This Online Shopping Session Succeed? Predicting Customer's Purchase Intention Using Embeddings. In Proceedings of the CIKM '22: 31st ACM International Conference on Information & Knowledge Management, Atlanta, GA, USA, 17–21 October 2022; pp. 2873–2882. [\[CrossRef\]](#)
52. Alves Gomes, M.; Wönkhaus, M.; Meisen, P.; Meisen, T. TEE: Real-Time Purchase Prediction Using Time Extended Embeddings for Representing Customer Behavior. *J. Theor. Appl. Electron. Commer. Res.* **2023**, *18*, 1404–1418. [\[CrossRef\]](#)
53. Ni, J.; Li, J.; McAuley, J. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China, 3–7 November 2019; pp. 188–197.
54. Liu, H.; Lu, J.; Yang, H.; Zhao, X.; Xu, S.; Peng, H.; Zhang, Z.; Niu, W.; Zhu, X.; Bao, Y.; et al. Category-Specific CNN for Visual-aware CTR Prediction at JD.com. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Virtual, 6–10 July 2020; pp. 2686–2696. [\[CrossRef\]](#)
55. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.; Dean, J. Distributed Representations of Words and Phrases and Their Compositionality. In Proceedings of the NIPS'13: 26th International Conference on Neural Information Processing Systems, Lake Tahoe, NV, USA, 5–10 December 2013; Volume 2, pp. 3111–3119.
56. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [\[CrossRef\]](#)

57. Cho, K.; van Merriënboer, B.; Bahdanau, D.; Bengio, Y. On the Properties of Neural Machine Translation: Encoder–Decoder Approaches. In Proceedings of the SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation, Doha, Qatar, 25 October 2014; pp. 103–111. [\[CrossRef\]](#)
58. Sutskever, I.; Vinyals, O.; Le, Q.V. Sequence to Sequence Learning with Neural Networks. In Proceedings of the NIPS'14: 27th International Conference on Neural Information Processing Systems, Montreal, QC, Canada, 8–13 December 2014; Volume 2, pp. 3104–3112.
59. Van Rossum, G.; Drake, F.L., Jr. *Python Reference Manual*; Centrum voor Wiskunde en Informatica: Amsterdam, The Netherlands, 1995.
60. Harris, C.R.; Millman, K.J.; van der Walt, S.J.; Gommers, R.; Virtanen, P.; Cournapeau, D.; Wieser, E.; Taylor, J.; Berg, S.; Smith, N.J.; et al. Array programming with NumPy. *Nature* **2020**, *585*, 357–362. [\[CrossRef\]](#)
61. McKinney, W. Data Structures for Statistical Computing in Python. In Proceedings of the Python in Science Conference, Austin, TX, USA, 10–16 July 2023; pp. 56–61. [\[CrossRef\]](#)
62. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*; Curran Associates, Inc.: Red Hook, NY, USA, 2019; pp. 8024–8035.
63. Shen, W. DeepCTR: Easy-to-Use, Modular and Extendible Package of Deep-Learning Based CTR Models. 2017. Available online: <https://github.com/shenweichen/deepctr> (accessed on 17 March 2023).
64. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
65. Buitinck, L.; Louppe, G.; Blondel, M.; Pedregosa, F.; Mueller, A.; Grisel, O.; Niculae, V.; Prettenhofer, P.; Gramfort, A.; Grobler, J.; et al. API design for machine learning software: Experiences from the scikit-learn project. In Proceedings of the ECML PKDD Workshop: Languages for Data Mining and Machine Learning, Prague, Czech Republic, 23–27 September 2013; pp. 108–122.
66. Akiba, T.; Sano, S.; Yanase, T.; Ohta, T.; Koyama, M. Optuna: A Next-generation Hyperparameter Optimization Framework. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Anchorage, AK, USA, 4–8 August 2019.
67. Shi, Y.; Yang, Y. HFF: Hybrid Feature Fusion Model for Click-Through Rate Prediction. In Proceedings of the Cognitive Computing—ICCC 2020: 4th International Conference, Held as Part of the Services Conference Federation, SCF 2020, Honolulu, HI, USA, 18–20 September 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 3–14. [\[CrossRef\]](#)
68. European-Parliament. *Regulation (EU) 2016/679 of the European Parliament and of the Council*; Official Journal of the European Union: Luxembourg, 2016.
69. Burri, M.; Schär, R. The reform of the EU data protection framework: Outlining key changes and assessing their fitness for a data-driven economy. *J. Inf. Policy* **2016**, *6*, 479–511.

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.