Prospective Modeling of Users for Online Display Advertising via Deep Time-Aware Model

Djordje Gligorijevic
djordje@verizonmedia.com
Yahoo Research
Sunnyvale, CA

Jelena Gligorijevic
jelenas@verizonmedia.com
Yahoo Research
Sunnyvale, CA

Aaron Flores
aaron.flores@verizonmedia.com
Yahoo Research
Sunnyvale, CA

ABSTRACT

Prospective display advertising poses a particular challenge for large advertising platforms. The existing machine learning algorithms are easily biased towards the highly predictable retargeting events that are often non-eligible for the prospective campaigns, thus exhibiting a decline in advertising performance. To that end, efforts are made to design powerful models that can learn from signals of various strength and temporal impact collected about each user from different data sources and provide a good quality and early estimation of users' conversion rates. In this study, we propose a novel deep time-aware approach designed to model sequences of users' activities and capture implicit temporal signals of users' conversion intents. On several real-world datasets, we show that the proposed approach consistently outperforms other, previously proposed approaches by a significant margin while providing interpretability of signal impact to conversion probability.

CCS CONCEPTS

• **Information systems → Display advertising**; **Web mining**;
• **Computing methodologies → Machine learning**.

KEYWORDS

prospective advertising, deep learning, time-aware prediction

ACM Reference Format:
Djordje Gligorijevic, Jelena Gligorijevic, and Aaron Flores. 2020. Prospective Modeling of Users for Online Display Advertising via Deep Time-Aware Model. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM '20), October 19–23, 2020, Virtual Event, Ireland.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3340531.3412689

1 INTRODUCTION

Online display advertising (DA) is a concept developed with the purpose of showing the most relevant ads to users anywhere online. It has been one of the fastest growing industries in the world, and in the U.S. alone, this industry amassed $100 billion dollars in 2018[1]. In order to have their ads shown to users, advertisers rely on Demand Side Platforms (DSPs) to reach relevant users through ad display opportunities, bid on the ad auctions and display advertisers' ads on their behalf. It is the job of the DSPs to learn which users could be interested in the advertisers' products and would become their business in the near future. In order to achieve that, DSPs try to learn as much as possible about users, by collecting their online footprints through the data collected from advertisers websites, won auctions, third-party data providers and from its properties.

Much of the DAs' business historically has been *retargeting*, a special case where ads are displayed to remind users who have already shown interest in advertisers business and hopefully generate conversions. As this particular form of DA by definition will not bring *new* customers to the advertisers, they have shown increased interest in *prospective* targeting of users. The goal of *prospective* targeting is the opposite of retargeting – users who have shown interest into advertisers business in the recent past should be excluded, and the goal becomes to generate new users as both visitors and converters for the advertiser. While the definition of retargeting users may significantly vary from one advertiser to another, in terms of the general advertising funnel (stages in which users are placed with respect to their probability of conversions [22]), prospective targeting should focus on users who are in the upper funnels (users further away from the conversion stage). Conversely, in terms of the advertising funnel, retargeting focuses on users in the lowest funnel stages (users very close to conversion).

Prospective modeling of users poses a particularly difficult task for DSPs, as the direct signals of users interests (such are visits to advertisers website or recent conversions with the same advertiser) are no longer viable. To maintain the high performance of user modeling, DSPs are given a challenging task to generate powerful models which are able to detect relevant, often weaker, signals users leave in their online trails and use them to the fullest extent. An example of such signals could be users' recent wedding related invoice signaling future interest in purchasing furniture or flight ticket to the honeymoon, whereas any signals related to furniture or flight browsing on advertiser's website could not be consumed.

Moreover, a very important aspect of prospecting user modeling is explainability. Advertisers often require DSPs to provide insights into how predictions were made, what individual signals and what signal combinations seemed important during the modeling process. For the case of prospective modeling, these signals when interpreted can bring exceptional value to the advertiser, as they would be able to fine tailor future campaigns for different user groups that resonate better and potentially reach more consumers.

[1] https://www.iab.com/wp-content/uploads/2019/05/Full-Year-2018-IAB-Internet-Advertising-Revenue-Report.pdf, accessed June 2020

CIKM '20, October 19–23, 2020, Virtual Event, Ireland

ACM ISBN 978-1-4503-6859-9/20/10…$15.00
https://doi.org/10.1145/3340531.3412689

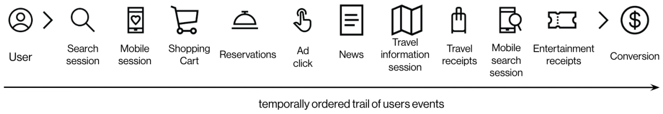


Figure 1: Visualization of user activity sequence with different groups of activities ordered by the time they occurred and ending with the action of advertiser's interest.

To create a generic view on signals users leave, the most natural choice is to create a time-ordered sequence of activities user performed collected by the DSP. An example of one such sequence or *trail* is provided in the Figure 1 where we observe multiple interactions of the user with different online properties such as mobile and desktop search, email receipts, reading news and interacting with ads. Modeling sequences of user events has been proposed in the past with great success [6, 21], however, to the best of our knowledge, it hasn't been used for strictly prospective modeling of users. Moreover, utilizing activities data to the full extent such as temporal aspect has been largely ignored when modeling conversions in DA. Motivated by the prior arts, and designing temporal activity transition impact to capture long- and short-term interests of users for prospecting conversion optimization, we developed a novel time-aware deep learning approach. We applied our approach on a dataset designed specifically for a prospective advertising problem and public purchase prediction dataset to show models applicability beyond prospective advertising use-case.

We summarize the contributions of this work below:

- We motivate and propose the problem of DAs prospective targeting. To the best of our knowledge we are the first to discuss challenges and opportunities of this task.
- We propose sequence learning approach to model time-ordered sequences of heterogeneous activities.
- We propose a novel time-aware mechanism to capture temporal aspect of events and thus better capturing their relevance to the conversion. The proposed approach accumulates up to 1.3% and 6% AUC lifts on public and proprietary datasets, demonstrating its superiority in multiple scenarios.
- Interpretability of novel time-aware mechanism is discussed in detail and it is used to contrast retargeting to prospective user modeling.

Prospecting audiences is a niche product requested by several major advertisers who typically spend more than $10 million per year on display advertising and is it on the path of being productized and sold. The offline results discussed here are the key milestone for this product, especially evaluating how little performance deteriorates compared to retargeting product which is shown in Section 5.2.5.

2 BACKGROUND AND RELATED WORK

A brief overview of online advertising ecosystem is given to stress the importance of predicting future conversions. Additionally, relevant prior works on conversion prediction and their contributions will be discussed with respect to this study.

2.1 Online Advertising

Major DSP platforms for display advertising (e.g., Google DoubleClick, Verizon Media DSP) allow advertisers to sign up and run campaigns and lines. The task for DSP platforms becomes to run advertiser's lines and serve users such that predefined key performance indicators (KPIs) goals are reached. This is achieved by participating in online auctions for different ad opportunities while optimizing DSP's goals.

The (simplified) optimization objective for each DSP line can often be formalized across all ad opportunities i in a window of time (i.e. a day) typically as:

$$\begin{aligned} &\arg\max_{bid_i} \sum_{i=1}^{N} \mathbb{I}(bid_i) * v_i \\ &\text{subject to: } \sum_{i=1}^{N} \mathbb{I}(bid_i) * c_i \le B, \end{aligned} \tag{1}$$

where number of won impressions is defined as

$$\mathbb{I}(bid_i) = \begin{Bmatrix} 1 & \text{if bid won} \\ 0 & \text{else} \end{Bmatrix}, \tag{2}$$

and impression value is defined as $v_i = pCVR_i * impression\_value_i$, with $pCVR_i$ being predicted conversion rate (in case the line is optimizing conversion, but it can be replaced with click-throught rate ($pCTR$) or any other activity estimate) and $impression\_value_i$ is the advertisers value of an impression. Optimization function is constrained such that total cost does not exceed the budget B.

For conversion predictions the bid is often controlled by the probability of user converting after ad is displayed, more precisely, the maximum bid is defined as a factored conversion probability $bid_i = f(\alpha * v_i)$ [11]. Function f represents optimal bidding strategy (often a linear function[2]) and α is often called a control parameter which includes several signals such as pacing [9].

As it can be observed, deciding on the maximum bid has three main optimization aspects of participating in online auctions. And the main focus of this study, the estimate of conversion probability $pCVR$ is one of the key components in the DSP business that drives performance and directs the system towards displaying ads to relevant users.

2.2 Modeling users' conversion prediction

In large scale advertising setups, conversion probability estimation has been successfully tackled through logistic regression [14] or random forests models. However, such systems depend on time intensive efforts of manually designing and selecting features, while the utility of handcrafted features is largely dependent on the domain knowledge of experts curating the features. Moreover, since typical applications are nonlinear, considering feature interactions quickly becomes prohibitively expensive due to a combinatorial explosion [13].

Recently, models with representation learning capabilities have also been proposed for CTR and CVR prediction tasks, e.g., factorization machines [15] for CVR or deep residual networks [18] and Siamese networks [8] for CTR that tackle problems of learning nonlinear interactions of features. Also, more prominently, models that capture information from the sequence such are RNNs have been proposed recently [1, 5, 6, 21] and they reportedly perform significantly better than their non-sequential counterparts. Moreover [1]

[2] https://observablehq.com/d/9d739b2bc5b22bd8, accessed June 2020

and [22] have used sequences of events from diverse data sources, while [1] has additionally proposed adding temporal information of events as an additional source of information to better model representations for conversion attribution task.

3 METHODOLOGY

3.1 Proposed Approach

We propose a novel model - Deep Time Aware conversIoN (DTAIN) model (Fig. 2) for modeling both long and short-term impacts of user's events. The DTAIN model takes sequence of event id's $\{e_i | i = 1 \dots N, \forall_e \in V\}$ and time difference of events' timestamp and the time point of prediction (usually time of an ad opportunity or timestamp of the last event in a sequence) as inputs. It then forwards this information through 5 blocks specifically designed to learn conversion rate prediction.

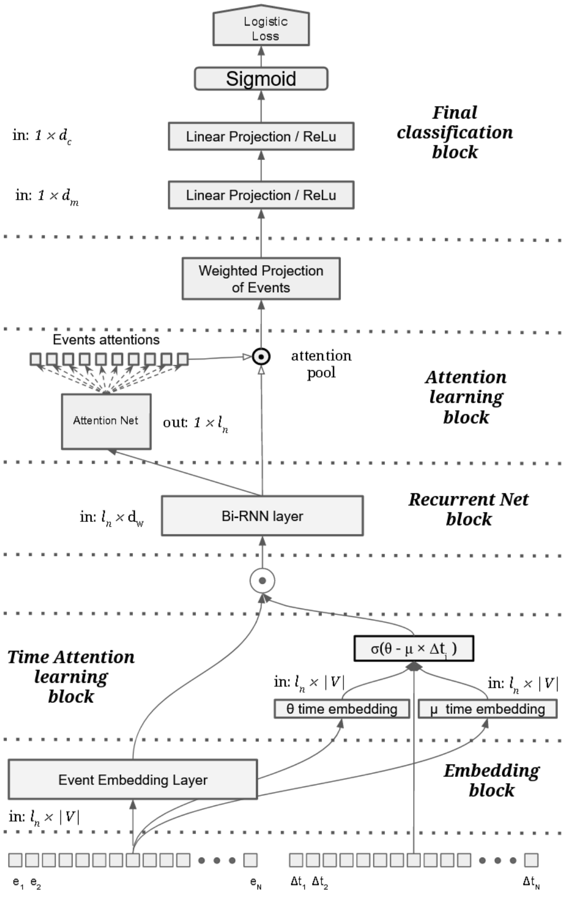


Figure 2: Graphical representation of the DTAIN model

3.1.1 Blocks of the DTAIN model.

Events and Temporal information embedding. Embeddings of events and temporal information are performed in two separate parts of the network. First, l_n events in the user's trail are embedded into vectors h_{e_i} of $h_{e_i} \in \mathbb{R}^{d_w=200}$ dimensional common space (*Embedding block*).

Temporal attention learning. Each event e_i is also associated with two additional single-dimensional learnable parameters: $\mu_{e_i} \in \mathbb{R}^{d_t=1}$ and $\theta_{e_i} \in \mathbb{R}^{d_t=1}$. These parameters are designed to model the temporal increment Δ_t as time difference between current state i and the state of interest j (i.e. timestep when pCVR is served):

$$\Delta_t = \tau_{e_j} - \tau_{e_i} \tag{3}$$

$$\delta(e_i, \Delta_t) = \sigma(\theta_{e_i} - \mu_{e_i}\Delta_t) \tag{4}$$

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{5}$$

$\delta(e_i, \Delta_t)$ captures the influence of the current event to conversion with θ_{e_i} measuring initial influence and μ_{e_i} measuring the change of the influence of the event with the time difference. Smaller $|\mu_{e_i}|$ refers to events whose influence does not change as we observe the event through different points in the users trails, while larger $|\mu_{e_i}|$ means that position and time of the event is very important for measuring its effect on conversion probability. Given that the Δ_t is always positive and provided that θ_{e_i} doesn't change, larger positive values of μ_{e_i} would mean that the temporal score is closer to 0, and larger negative values that is closer to 1. Similar ways of modeling temporal increments can be seen in known results of Euler's forward method [4] for modeling change of state in dynamic linear systems. In our case, we opted for using time information as an event-level contribution to the final task, thus Sigmoid function was used to transform $\theta_{e_i} - \mu_{e_i}\Delta_t$ into (0, 1) range. This approach allows us to model same events that happened multiple times within the same user trail differently, i.e. giving more attention to events that happened more recently.

Other formulations of modeling time information given in the literature are mostly in the context of time decay which biases models to focus less on long term effect of the individual events. Several approaches [3, 16] propose a number of ways for generating time features such are linear, tanh, exponential or using temporal deltas, which are to be added to the existing feature set. However, all proposed strategies are cases of strict time decay effect where only events which happened close to prediction time may have higher values. Moreover, [23] proposed adding temporal gates to LSTM cells to model the time passed between subsequent events in a event-oblivious manner. Another direction involves embedding coarse grained categorical values of time, such as embedding event hour category [10, 12]. Attention regularization mechanism was an approach proposed to prevent embedding being similar for events that occur at a larger temporal distance [1]. Finally, [17] proposed using time features, similarly to [3, 16] for events attention generation.

The majority of existing approaches model time decay factors or high level temporal categorizations, with a few approaches modeling event-oblivious transition probabilities, not capturing the invariances of different events. Oppositely, our approach learns event-specific initial and time influence factors [2] which are then used as a gate to control how much information passes from each event embedding into the model's architecture through the Sigmoid activation, rather than creating a distribution of temporal signal across all events in the trail as Softmax activation would do [17]. Furthermore, impact of each event in the sequence is modeled regardless of how far away from prediction the event occurred which is achieved through the temporal event state change representation modeling. Such an approach resonates with prospective advertising use case the best, as preventing algorithm by design to look into early events may loose opportunities.

The learned embeddings and contributions of each event are then summarized to obtain new event representation v_{e_i}:

$$\forall_{h_{e_i} \in \{i=1\ldots l_n\}} \forall_{\delta(e_i,\Delta_t) \in \{i=1\ldots l_n\}} v_{e_i} = h_{e_i} * \delta(e_i, \Delta_t) \tag{6}$$

resulting again in $v_{e_i} \in \mathbb{R}^{d_w=100}$ dimensional space. This way of modeling preserves model interpretability, given that for each event we can measure its initial and time influence factors and interpret their values as described above.

Computational aspect. The proposed temporal attention block is also memory efficient. Even though it requires learning additional two parameters per unique event, with the current hyperparameter setup, temporal attention only adds additional 1% of parameters to the model that need to be learned.

Recurrent Net block. The resulting embeddings of events are then fed into bi-directional RNN model (with GRU cells used for both forward and backward pass networks):

$$g_{e_1}, g_{e_2}, \ldots, g_{e_N} = biRNN(v_{e_1}, v_{e_2}, \ldots, v_{e_N}, \theta_{GRU}) \tag{7}$$

Bi-directional RNN ensures that the model learns complex relations between events, which is particularly important for user trails where events may be grouped by sessions which carry higher order information than the events themselves [8]. The resulting embeddings g_{e_i} are in $\mathbb{R}^{d_m=100}$ dimensional space.

Attention learning block. In order to learn rich representations of user's trail, it is imperative to focus on events that carry the most information. To learn representations that focus on important parts of the user trail we employ a dedicated attention mechanism on top of sequence modeling features [8]. Employed attention block yields event scores, that highlight events of greater importance for the task at hand. In our particular case, attention model is implemented as a two-layered individual neural network $s_q(g_e; \theta_e)$ with hidden dimensions of $\mathbb{R}^{d_{a_1}=100}$ and $\mathbb{R}^{d_{a_2}=1}$, and Softmax at its final layer:

$$t_{e_i} = \frac{\exp(s_e(g_{e_i}; \theta_e))}{\sum_{i=1}^{l_n} \exp(s_e(g_{e_i}; \theta_e))}. \tag{8}$$

Neural network $s_e(g_{e_i}; \theta_e)$ learns real valued scores for each i^{th} event in a given user trail. Attention learning in the DTAIN model is coupled with the entire network (end-to-end).

Event attentions t_{e_i} are then used to re-weight their input representations g_{e_i} and to obtain compact representation of the entire sequence $s = \sum_i t_{e_i} * g_{e_i}$. There are other ways of obtaining compact representations s, such as sum, average or max of individual event vectors. However, our experiments, as well as available literature [7, 20], demonstrate that such strategies are inferior to using attention machanisms.

Learning to predict from the resulting representation. The summarized user trail representation from previous block is finally fed to a sequence of two fully connected layers with an inner dimension $\mathbb{R}^{d_c=100}$ and ReLU nonlinearities before finally passing through a sigmoid layer $\sigma(\cdot)$ to obtain the probability of conversion (pCVR).

Finally, to optimize the parameters of the DTAIN model, we have obtained logistic loss $\mathcal{L}$ for the CTR prediction based on logits from the topmost layer:

$$\mathcal{L} = -\frac{1}{N}\sum_{n=1}^{N}(y_n \log(\hat{y}_n) + (1 - y_n)\log(1 - \hat{y}_n)), \tag{9}$$

where $\hat{y}_n$ are obtained logits after final sigmoid layer and y_n is conversion label for the n^{th} user trail.

4 DATA DESCRIPTION

4.0.1 Public RecSys 2015 challenge dataset. We conducted purchase prediction experiments on publicly available dataset obtained from RecSys Challenge in 2015. This dataset contains a collection of sequences of click events with respective timesteps from Yoochoose website. Some of the click sessions ended with a purchase event (if so, label was set as positive, otherwise negative). There are $1,965,359$ sessions in the training and $279,999$ in test dataset, downsampled to obtain $\sim$ 11% of positives. These sessions are much shorter than the proprietary user trails, and they reflect reproduceability of retargeting results, as there is no publicly available prospecting dataset to the best of our knowledge.

4.0.2 User activity trails from Verizon Media (VM). We have also conducted experiments using user activity trails data from Verizon Media[3]. This includes activities chronologically collected from a user, derived from various sources, e.g., Yahoo Search, commercial email receipts, reading news and other content on publisher's webpages associated with Verizon Media such as Yahoo and AOL homepages, advertising data (e.g., ad impressions, clicks, conversions, and site visits). The representation of an activity comprises of activity ID, time stamp, its type (e.g., search, invoice, reservation, content view, order confirmation, parcel delivery), and a raw description of the activity (e.g., the exact search query for search activities) after stripping personally identifiable information. It is expected for all DSPs to have similar spectrum of data sources thanks to the existence of large number of third party data providers.

To ensure legality of information used, datasets created for each advertiser strictly follow legal guidelines determined by the contract, i.e. data collected from advertiser A will never be used for any optimization task for advertiser B.

Datasets used in this study are collected from two major anonymized advertisers from the highly prevalent *retail* and *communications* categories that we will denote as Advertiser 1 and Advertiser 2, respectfully. Advertiser 1 has defined three different conversion rules for it's three retail portfolios, while Advertiser 2 defined a single conversion rule. Training sets for the two advertisers are comprised of $788,551$ users in train and $196,830$ for test set for Advertiser 1, and of $917,451$ users in train and $229,126$ for test set for Advertiser 2, collected over an undisclosed period longer than 100 days. As for unique activities dictionary, for Advertisers 1 and 2 we collected $243,190$ and $243,713$ most frequent events respectively. Filtering of events is done before downsampling negative users so as to select events that occurred in more than 1000 unique user trails. User activity trail consists of the last 500 events after deduplication as per dataset statistics, 80% of all users had sequence length <500.

[3] https://webscope.sandbox.yahoo.com/

Problem setup and dataset construction. Retargeting event is defined by the advertiser, i.e. user browsing furniture items on advertiser's website may be regarded as a retargeting user for furniture conversions for that advertiser in the next several months. As mentioned in the Introduction, advertisers who focus on prospective advertising are only interested in new converters from non-retargeting set of users, however, learning to target prospecting users by optimizing conversions is very difficult. Namely, a common theme for a majority of *retail and communication categories advertisers* is that most users will visit their webpage at least once before converting. Thus directly optimizing for conversion will likely result in a retargeting-biased conversion prediction as we demonstrate in Section 5.2.5. Conversion statistics with respect to advertiser site visits are provided in Table 1. 62.91% and 41.85% of the cases have users visiting Advertiser 1 and Advertiser 2 websites, respectively, a day or more before the conversion. As suspected the early visits happening a day or more before the conversion are more prominent for the retail advertiser. The objective of DSP prospective targeting is to show impressions to users before they become retargeting users, thus bringing new users to the advertiser and boosting their sales.

Conversion	Site visit within 24h	Site visit before 24h	Percentage
	Advertiser 1		
1	1	0	38.09%
1	1	1	62.91%
	Advertiser 2		
1	1	0	58.15%
1	1	1	41.85%

Table 1: Percentages of conversions with respect whether the user's first site visit (retargeting event) occurred within 24h of conversion or before.

Algorithms trained on the original data collected could be biased towards modeling retargeting signals only, i.e. rule-based system that observes data from a day before can easily achieve recall of 62.91% for Advertiser 1, while a near-real-time system would achieve recall of 100% (but a low precision as only 5% users who visited convert). To prevent this from happening, we propose a careful dataset construction that only keeps users' actions before they became retargeting users as highlighted by the advertiser. The process is shown in Fig. 3. In Fig. 3 a) and b), we can see two different strategies of constructing the dataset. First one includes all events up to the conversion including strong retargeting signals, while the second one stimulates prospecting signals by removing all events past the first retargeting event. One day trail cutting is used for building training dataset. During inference users are labeled as retargeting/prospecting and thus no events are removed.

Finally, for successful prospective advertising it is also important to predict the retargeting event occurrence. Figure 3 c) shows dataset for the new optimization task that allows DSPs to show the ad to the user before they become labeled as retargeting.

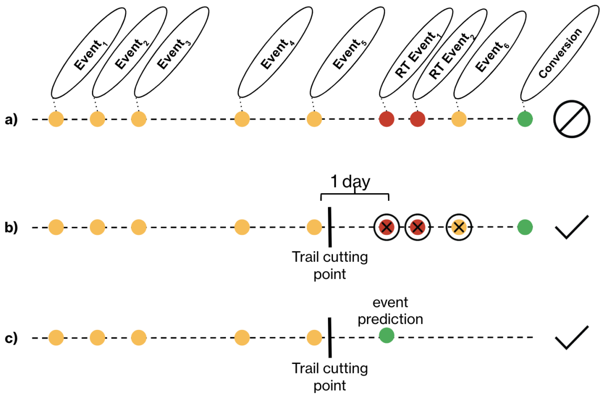


Figure 3: Visualisation of trail cutting process event before retargeting event happens.

5 EXPERIMENTS

Baseline algorithms, evaluation metrics and training and serving infrastructure are first described, followed by discussion of the results on both public and proprietary datasets.

5.1 Baselines

The following models are selected to either represent previously published studies or as models that are expected to fit well with the given setup.

1. Random Forests (RF): Random Forests algorithm ran on top of chi squared feature selection using features extracted from user sequential trails data that mimics the exact setup used in the current production.
2. Recurrent Neural Network (RNN): An RNN with embedding layer and GRU cells to ensure fast convergence with two fully connected layers for classification.
3. 1-dimensional Convolutional Neural Networks (CNN): A 1-dimensional CNN on top of learned event embeddings with two fully connected layers for classification.
4. RNN with attention layer (RNN+Attn): An extension of the RNN model with additional attention layer used to summarize the sequence [7].
5. RNN with self attention layer (RNN+SelfAttn): Alternative extension of the RNN model with self-attention layer used learn higher order interactions between events before the RNN block [19].

Models configuration and training. DTAIN model is based on the RNN+Attn baseline, thus hyperparameters used in DTAIN (given in Section 3.1.1) are kept the same for all RNN-based approaches. CNN architecture uses four 1-D convolutional blocks with 64 filters of width 3 and batch normalization between layers, resulting output is flattened and forwarded to the same classification block as used by the other algorithms. For all algorithms weights are initialized by a truncated normal initializer. To optimize $\mathcal{L}$, we use stochastic gradient descent with Adam optimizer, and the best learning rate found through grid search was 0.001.

The proposed prospecting solution is designed to run on collected user activities in an ad hock or near-real-time offline manner, as user fires an event. The training is conducted on offline data and inference is run on same data with new events appended, using distributed infrastructure. Obtained scores are both tiered for targeting and passed to the bidder for optimization purposes. All deep learning models were, thus, trained on distributed TensorflowOnSpark[4] infrastructure with 20 GPU (Nvidia K80) machines. This system complies with production guidelines where users need to be scored in near-real-time fashion.

5.1.1 Evaluation metrics. For assessing the quality of estimated CVR probabilities, we use the area under the ROC curve (AUC) classification performance measure, in addition to Accuracy, Precision and Recall obtained after choosing the classification threshold.

5.2 Experimental results

The DTAIN and baseline models are evaluated on two described datasets and the results are given below.

5.2.1 Results on public dataset. Results of the experiments on public data source purchase prediction task are given in Table 2. The ROC AUC and other metrics results show that the proposed DTAIN model overall outperforms all of the baselines, whereas all of the sequence learning baselines outperformed random forests by a significant margin. Competitive results of DTAIN models show that use of temporal information can truly help the predictive task even in short-sequence datasets such as this one, as all examples in the public dataset occur within one hour time window. It may be surprising that modeling temporal dynamics as proposed helps, however, as discussed in the Section 3.1, the temporal information has two aspects to it and thus can model initial impact of the events (in addition to temporally changing impact) to the purchase thus providing additional information to the classifier.

	ROC AUC	Accuracy	Precision	Recall
RF	0.6168	0.7608	0.2025	0.4093
CNN	0.7534	0.6779	0.2087	0.7041
GRU	0.7504	0.6958	0.2142	0.6746
GRU+SelfAttn	0.7029	0.6734	0.1907	0.6184
GRU+Attn	0.7639	**0.6997**	**0.2195**	0.6904
DTAIN	**0.7666**	0.6943	0.2186	**0.7047**

Table 2: Performance metrics on the purchase prediction Youchoose dataset for all algorithms.

5.2.2 Results on VM dataset - prospecting users pCVR prediction.

Results on binary classification. In this section we conduct experiments on a binary classification task predicting whether a user converted for any of the conversion rules set by the Advertisers 1 and 2. Overall, more prominent results are obtained compared to the public dataset on the proprietary dataset where temporal aspect plays a larger role in prediction (Table 3).

First of all, there are differences in task difficulty between the two advertisers, as predicting conversion for retail advertisers tends to be an easier task than predicting conversion for communication advertisers. Furthermore, all sequence modeling baselines outperformed RF algorithm, while DTAIN outperforms all baselines by a large margin on a majority of metrics and on both datasets. The large time window allows the time mechanism parameters to properly capture both initial and temporal distance impacts of each event. Moreover, as the time window is significantly larger, the events may repeat multiple times, and time mechanism will be able to select the most important events out of the redundant ones through the event's temporal distance impact and thus filter out the noise in the data. Both properties are considered crucial for prospective conversion modeling.

	ROC AUC	Accuracy	Precision	Recall
Advertiser 1				
RF	0.9323	0.8494	0.5223	0.8643
CNN	0.9408	0.8757	0.5773	0.8808
RNN	0.9436	0.8855	0.6010	0.8804
RNN+Attn	0.9424	0.8937	0.6231	0.8764
RNN+SelfAttn	0.9440	0.8846	0.6002	0.8691
DTAIN	**0.9519**	**0.9031**	**0.6478**	**0.8854**
Advertiser 2				
RF	0.8845	0.8330	0.1963	0.7469
CNN	0.9034	0.8225	0.1771	**0.8378**
RNN	0.8929	0.8474	0.1942	0.7902
RNN+Attn	0.8865	0.8525	0.1974	0.7735
RNN+SelfAttn	0.8943	0.8461	0.1921	0.7851
DTAIN	**0.9031**	**0.8857**	**0.2434**	0.7651

Table 3: Performance metrics on the proprietary user trails dataset for all algorithms and both advertisers.

Results on multi–task classification. Next, results for the multi–task classification setup (Table 4) are discussed, where prediction is made whether a user will convert for any of the three different conversion rules defined by advertiser 1. DTAIN shows the best performance on majority of metrics across the four tasks, always having the top performance at ROC AUC metric. This evaluation shows that the DTAIN model is overall the best among the chosen baselines once again. The DTAIN model was prominently the best approach for Task 1 (prediction if the user is not going to convert) which is very important for the bidding system to know if it should bid for a user or not and the Task 3.

5.2.3 Results on VM dataset - users' retargeting event prediction. Next, in terms of prospective DA, we evaluate the performance of all algorithms on task of predicting whether the first retargeting (thus not conversion) event will occur within 24 hours.

In order for the DSP to be attributed with a conversion, the DSP must show its ability to display ads to prospective users before they become retargeting users (before firing their first retargeting signal). In order to achieve this, DSPs need to estimate how likely is that the user will visit advertisers website in the near future and make sure to show advertiser's ad to the user before that event.

DTAIN model and all baselines are run on the retargeting event prediction dataset and results are shown in Table 5.

Compared to conversion prediction task we can see that the retargeting event prediction task is slightly easier for both advertisers and all baselines, and we suspect that this is due to the fact that there are simply more positive events in the dataset. There are $\sim 3.9M$ positive retargeting events vs $\sim 230K$ conversions for advertiser

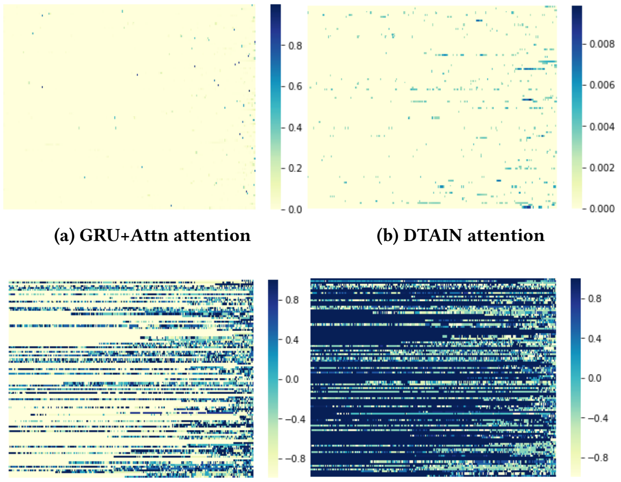
[4] https://github.com/yahoo/TensorflowOnSpark, accessed August 2020

Advertiser 1	ROC AUC	Accuracy	Precision	Recall
Task 1				
RF	0.9211	0.8197	0.9689	0.8107
CNN	0.9382	0.8821	0.9735	0.8832
RNN	0.9517	0.8888	**0.9777**	0.8874
RNN+Attn	0.9472	0.8917	0.9753	0.8933
RNN+SelfAttn	0.9482	0.8806	0.9767	0.8783
DTAIN	**0.9523**	**0.9001**	*0.9774*	**0.9015**
Task 2				
RF	0.8992	0.7603	0.0701	0.9185
CNN	0.8713	0.7407	0.0633	0.8889
RNN	0.8888	0.7757	0.0718	0.8785
RNN+Attn	0.8961	0.7734	0.0728	0.9032
RNN+SelfAttn	0.8868	0.7695	0.0698	0.8758
DTAIN	**0.8993**	**0.7697**	**0.0721**	**0.9083**
Task 3				
RF	0.8893	0.7426	0.1874	0.9039
CNN	0.8867	0.7880	0.2154	0.8733
RNN	0.9032	0.7992	0.2271	0.8882
RNN+Attn	0.9071	0.8137	0.2385	0.8693
RNN+SelfAttn	0.9022	0.8001	0.2263	0.8757
DTAIN	**0.9113**	**0.8032**	**0.2318**	**0.8946**
Task 4				
RF	0.8899	0.7525	0.2282	0.8952
CNN	0.8879	0.8034	0.2674	0.8554
RNN	0.9038	0.8096	0.2778	**0.8815**
RNN+Attn	0.9031	0.8132	0.2803	0.8704
RNN+SelfAttn	0.9033	**0.8203**	**0.2873**	0.8608
DTAIN	**0.9084**	0.8146	0.2833	0.8805

Table 4: Performance metrics on the proprietary user trails dataset prediction of different conversion tasks for advertiser 1.

	ROC AUC	Accuracy	Precision	Recall
Advertiser 1				
RF	0.9210	0.8778	0.9481	0.8929
CNN	0.9393	0.8961	0.9582	0.9070
RNN	0.9477	0.8973	0.9633	0.9033
RNN+Attn	0.9452	0.9006	0.9614	0.9097
RNN+SelfAttn	0.9503	0.8935	0.9640	0.8976
DTAIN	**0.9745**	**0.9294**	**0.9772**	**0.9316**
Advertiser 2				
RF	0.9112	0.8963	0.9540	0.9188
CNN	0.9551	0.9107	0.9716	0.9189
RNN	0.9438	0.9030	0.9676	0.9132
RNN+Attn	0.9466	0.9078	0.9688	0.9181
RNN+SelfAttn	0.9402	0.8979	0.9660	0.9084
DTAIN	**0.9746**	**0.9274**	**0.9821**	**0.9290**

Table 5: Performance metrics on the proprietary user trails dataset for all algorithms predicting if the retargeting event will occur in the next day.



(a) GRU+Attn attention

(b) DTAIN attention

(c) DTAIN θ_{e_i}

(d) DTAIN μ_{e_i}

Figure 4: Heat maps of events attentions scores for 100 randomly sampled converters for Advertiser 2.

1 and $\sim 5M$ positive retargeting events vs $\sim 52K$ conversions for advertiser 2 on the drawn user sample.

Like in the previous setups, we observe that sequence models both with and without attention mechanisms outperform RF, and also we see that DTAIN outperforms all baselines by an even larger margin than on the conversion prediction dataset thus strengthening its position as a powerful algorithm for prospective advertising.

5.2.4 Attention analysis and interpretation. To tap into the explainability of the models we randomly selected a hundred converters and analyzed attentions of their events for the communications advertiser. We compare DTAIN model primarily against the GRU+Attn model, which has shown properties of explainability in the past [7]. From Fig. 4a it can be seen that GRU+Attn model assigns attentions across the users trails, highlighting not only events that happened close to conversion which is a desirable property for prospective advertising. The DTAIN model has a slightly different mechanism of attention as time plays a significant role in allowing information from different signals to be passed through the network. As discussed in Section 3, key parameters θ_{e_i} and μ_{e_i} have interesting interpetability properties. To show this, we plot scores of both the key parameters in Fig. 4c, 4d, and the attentions from the attention block in Fig. 4b. Interestingly, we can see that there are plenty of high positive values of θ_{e_i} and high negative values of μ_{e_i} further away from the end of sequences, in addition to the expected ones closer to the end of it. This means that DTAIN is capturing both long term as well as short term patterns and controls which event signals fully pass through the rest of the network.

These interesting findings allow us to use the attention scores for explainabilty to the advertisers by providing insights into both long- and short-term patterns and important events that they can further use to improve their creatives and advertising strategies.

5.2.5 Retargeting vs Prospecting pCVR estimation discussion. We finally compare the *pCVR* estimation tasks for retargeting and prospecting options. As we discussed before, the two tasks are targeting-wise mutually exclusive, the two prediction tasks, however, could be directly compared thanks to the trail cutting strategy. We have, thus, run standard retargeting experiment including retargeting events a day before conversion and compared AUC's of the retargeting and prospecting tasks for all algorithms. Figure 5 shows percentage of changes going from retargeting task to prospecting. As we can observe, in a majority of algorithms for both advertisers we see an AUC drop of few percentages implicating the increased difficulty of the latter task. More importantly, with the loss of retargeting features the AUC has not dropped significantly which is the key milestone result for productizing prospecting audiences product. The AUCs don't uncover the real differences, thus, we looked into the top events as scored by the temporal attention. We found that for the prospecting case, algorithms focused on a wider specter of events including commercial email receipts, content read, searched queries, etc., while for the retargeting case, algorithms focused almost exclusively on browsing through the advertisers website for a majority of users, seldom selecting other events such are email receipts or content reads. We can conclude that the dominant retargeting signals have biased algorithms away from useful signals that were exploited in the prospecting dataset and that did not necessarily occurred close to conversion point. Our final analysis shows the there really exist a difference between two, seemingly similar tasks, reinforcing our dedication to the prospective modeling.

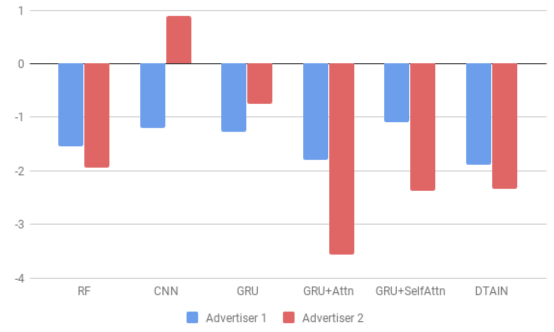


Figure 5: Percentage of AUC drop for estimating pCVR from retargeting to prospecting tasks for two advertisers.

6 CONCLUSIONS AND FUTURE WORK

In this study we proposed a sequence based approach for modeling conversion prediction based on users' activity trails that leverages both the sequence and temporal information of events collected from many data sources. We proposed a new way to model temporal information for prospective conversion prediction that preserves ability of interpretation, and finally we showed that the DTAIN model outperforms baselines that represent state-of-the art on both public and proprietary datasets. However, as the data is collected from many data sources, and different events may repeat often or periodically there is still noise in data heterogeneity that the algorithms need to address properly, thus developing novel techniques to address these concerns will be the next steps.

REFERENCES

[1] S. K. Arava, C. Dong, Z. Yan, A. Pani, et al. Deep neural net with attention for multi-channel multi-touch attribution. *arXiv preprint arXiv:1809.02230*, 2018.

[2] T. Bai, S. Zhang, B. L. Egleston, and S. Vucetic. Interpretable representation learning for healthcare via capturing disease progression through time. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 43–51. ACM, 2018.

[3] A. Beutel, P. Covington, S. Jain, C. Xu, J. Li, V. Gatto, and E. H. Chi. Latent cross: Making use of context in recurrent recommender systems. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pages 46–54. ACM, 2018.

[4] X. H. Cao, C. Han, and Z. Obradovic. Learning a dynamic-based representation for multivariate biomarker time series classifications. In *2018 IEEE International Conference on Healthcare Informatics (ICHI)*, pages 163–173. IEEE, 2018.

[5] Y. Cui, R. Tobossi, and O. Vigouroux. Modelling customer online behaviours with neural networks: applications to conversion prediction and advertising retargeting. *arXiv preprint arXiv:1804.07669*, 2018.

[6] D. Gligorijevic, J. Stojanovic, A. Raghuveer, M. Grbovic, and Z. Obradovic. Modeling mobile user actions for purchase recommendations using deep memory networks. In *41st Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval*, 2018.

[7] D. Gligorijevic, J. Stojanovic, W. Satz, I. Stojkovic, K. Schreyer, D. Del Portal, and Z. Obradovic. Deep attention model for triage of emergency department patients. In *2018 SIAM International Conference on Data Mining (SDM 2018)*, 2018.

[8] J. Gligorijevic, D. Gligorijevic, I. Stojkovic, X. Bai, A. Goyal, and Z. Obradovic. Deeply supervised model for click-through rate prediction in sponsored search. *Data Mining and Knowledge Discovery*, Apr 2019.

[9] N. Grislain, N. Perrin, and A. Thabault. Recurrent neural networks for stochastic control in real-time bidding. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery &#38; Data Mining*, KDD '19, pages 2801–2809, New York, NY, USA, 2019. ACM.

[10] H. Jing and A. J. Smola. Neural survival recommender. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, pages 515–524. ACM, 2017.

[11] N. Karlsson. Control problems in online advertising and benefits of randomized bidding strategies. *European Journal of Control*, 30:31–49, 2016.

[12] Y. Li, N. Du, and S. Bengio. Time-dependent representation for neural event sequence prediction. *arXiv preprint arXiv:1708.00065*, 2017.

[13] H. B. McMahan, G. Holt, D. Sculley, M. Young, D. Ebner, J. Grady, L. Nie, T. Phillips, E. Davydov, D. Golovin, et al. Ad click prediction: a view from the trenches. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1222–1230. ACM, 2013.

[14] B. N., K. R., and M. S. A large scale prediction engine for app install clicks and conversions. In *International Conference on Information and Knowledge Management (CIKM)*, 2017.

[15] J. Pan, Y. Mao, A. L. Ruiz, Y. Sun, and A. Flores. Predicting different types of conversions with multi-task learning in online advertising. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1834–1842. ACM, 2019.

[16] W. Pei and D. M. Tax. Unsupervised learning of sequence representations by autoencoders. *arXiv preprint arXiv:1804.00946*, 2018.

[17] A. Rajkomar, E. Oren, K. Chen, A. M. Dai, N. Hajaj, M. Hardt, P. J. Liu, X. Liu, J. Marcus, M. Sun, et al. Scalable and accurate deep learning with electronic health records. *NPJ Digital Medicine*, 1(1):18, 2018.

[18] Y. Shan, T. R. Hoens, J. Jiao, H. Wang, D. Yu, and J. Mao. Deep crossing: Web-scale modeling without manually crafted combinatorial features. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 255–262. ACM, 2016.

[19] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[20] S. Zhai, K.-h. Chang, R. Zhang, and Z. M. Zhang. Deepintent: Learning attentions for online advertising with recurrent neural networks. In *22nd ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining*, pages 1295–1304. ACM, 2016.

[21] Y. Zhang, H. Dai, C. Xu, J. Feng, T. Wang, J. Bian, B. Wang, and T.-Y. Liu. Sequential click prediction for sponsored search with recurrent neural networks. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014.

[22] Y. Zhou, S. Mishra, J. Gligorijevic, T. Bhatia, and N. Bhamidipati. Understanding consumer journey using attention based recurrent neural networks. *KDD*, 2019.

[23] 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 *IJCAI*, pages 3602–3608, 2017.