Personalized Dynamic Counter Ad-Blocking Using Deep Learning

Shuai Zhao, *Member, IEEE* Michael K. Chen, Cristian Borcea, *Member, IEEE* and Yi Chen, *Member, IEEE*

Abstract—The fast increase in ad-blocker usage has resulted in significant revenue loss for online publishers. To mitigate this, many publishers implement the Wall strategy, where an adblock user is asked to whitelist the intended webpage. If the user refuses, the result is a loss-loss situation: the user is denied access to content, and the publisher cannot receive revenue. An alternative strategy, called AAX, is to show only acceptable ads to users. However, acceptable ads generate less revenue than regular ads. This article proposes personalized counter ad-blocking that dynamically chooses a counter ad-blocking strategy for individual users. To implement it, we propose a novel deep learning-based whitelist prediction model. Adblock users predicted to whitelist a page receive the Wall strategy; the others receive the AAX strategy. The proposed Deep Ad-Block Whitelist Network (DAWN) for whitelist prediction captures page characteristics, user interests in pages and their sensitivity to ads, reflected in historic behavior, using a deep learning mechanism. Furthermore, DAWN leverages multi-task learning on whitelist prediction and dwell-time prediction to boost performance. DAWN's effectiveness is validated on a real-world dataset provided by Forbes Media. The experimental results demonstrate the advantages of the proposed counter ad-blocking policy over existing policies on revenue generation and user engagement.

Index Terms—Online advertising, Ad-blocking, User behavior, Deep learning, Revenue, User engagement, Personalization

1 Introduction

Digital technologies and the Internet have dramatically changed the content publishing industry. Today, most content on the Internet is free, and the primary revenue source for publishers is digital advertising [1], [2]. Online advertising generated over $300 billion in 2019 [3]. As Figure 1 shows, there are three stakeholders in free online publishing. Users view free content and "pay" with their attention for ads displayed on the web pages. Publishers spend money to generate content and receive ad revenue. Advertisers pay publishers for displaying ads and receive user's attention on the ads. The ad-supported web publishing ecosystem provides opportunities to all three parties. Users can receive high-quality content for free. Publishers can reach out to a much broader audience than ever before. Advertisers deliver ads that are targeted to individual users, with potential benefits of enhancing user shopping experience and achieving higher marketing effectiveness with lower cost.

However, excessive ads can be annoying. More and more users opt to use ad-blockers, which are tools (typically browser plugins) that prevent ads from being rendered on a web page. As shown in Figure 2, there is a wide range of browser extensions available to end users, with Adblock Plus as the most popular ad-blocker in 2021. According to a report by DigiDay, over 40% of all US users have used ad-blockers and their usage was expected to result in a loss of around $35 billion in the global online advertising revenue [4].

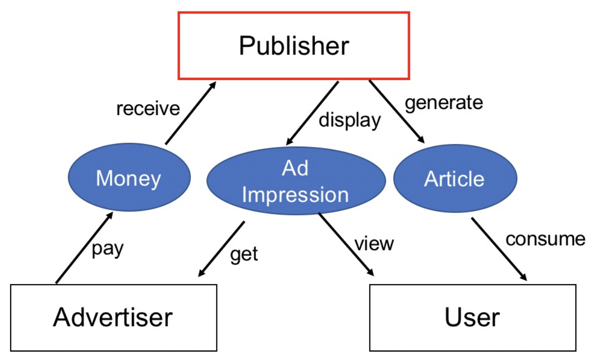
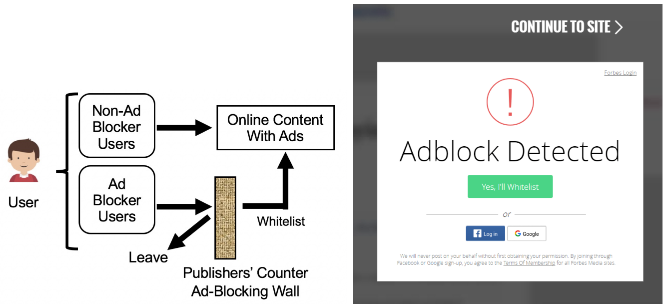


Fig. 1. Stakeholder relationships in an ad-supported web publishing ecosystem.

Fig. 2. Examples of ad-blockers

In the face of significant revenue loss, online publishers launch counter ad-blocking strategies. A previous study found that counter ad-blocking scripts were used by more than 30% of the 1,000 most popular domains [5]. The most popular counter ad-blocking strategy, as illustrated in Figure 3(a), is the Wall strategy. When an ad-blocker is detected, a publisher shows a dialog popup requesting the user to turn off or pause the ad-blocker, i.e. *whitelist* the publisher's

- *S. Zhao and Y. Chen are with Martin Tuchman School of Management, and C. Borcea is with Department of Computer Science, New Jersey Institute of Technology. Email: {sz255, borcea, yi.chen}@njit.edu. Y. Chen is the correspondence author.*
- *M. Chen is with Raffles Institution, and the New Jersey Institute of Technology. Email: michaelchenkaijie2004@gmail.com.*



(a) Workflow of the Wall strategy. (b) Screenshot of the Wall on the Forbes website.

Fig. 3. Illustration of the counter ad-blocking Wall strategy.

website or the specific page that the user intends to view (Figure 3(b)). If the user rejects the request, access to the content will be denied. However, such a forceful strategy is irritating to many. In fact, more than 60% of the ad-blocking users choose not to whitelist [6]. This results in a loss-loss situation: users are unable to view the intended content, and publishers lose a nontrivial amount of user traffic, which has long-term implications on readership and brand awareness. The ongoing ad-blocking "battle" between the adblock users and the publishers can break the free ad-supported online publishing ecosystem.

An alternative counter ad-blocking strategy is to use the acceptable ads exchange (AAX) [7]. An agreement is signed by the publisher and the ad-blocking company, where publishers are allowed to display a limited number of ads to adblock users, adhering to guidelines set by the Acceptable Ads Commission [1]. With this strategy, all adblock users can access the content with the acceptable ads. However, ad revenue per user, per page largely decreases due to restrictions on ad placement (e.g., location of ads on a web page and the size of ads). Furthermore, to enroll in the AAX program, large publishers need to pay about 30% of the revenue generated by AAX [8], [9]. These restrictions produce significant financial challenges for publishers, who are struggling to produce high-quality free content with a large decrease on ads revenue.

As we can see, the Wall strategy fails to account for users with high ad sensitivity, while the AAX strategy assumes all users have high ad sensitivity, which is not necessarily the case. Both strategies result in significant revenue loss of publishers.

In this article, we propose a policy that uses personalized counter ad-blocking, based on the inferred user interests in the intended pages and their sensitivity to ads. Given the popularity of the Wall and AAX strategies, we propose the following policy: if we can accurately predict whether or not a user will whitelist an intended page, then we use the Wall strategy to those who will, and use the AAX strategy to those who will not. Here, a *policy* can use multiple strategies under different conditions, where a strategy is atomic. The simplest, and the most common, form of a policy is to use a single strategy for all users.

The core of the proposed personalized counter ad-blocking policy lies in effective whitelist prediction. The key technical challenges for whitelist prediction include how to extract features relevant to whitelisting behavior from large amounts of mostly unstructured data and how to model the interaction between user behavior and the page properties.

To address these challenges, we formulate the whitelist prediction as a recommender system problem and propose a novel deep learning based framework, called Deep Ad-Block Whitelist Network (DAWN) model. We utilize deep learning for its strength in detecting intricate patterns and its strong representation ability [10] to capture user features and page characteristics. We make a conjecture that whitelist behavior largely depends on a user's interest in the page they intend to view and their ad sensitivity. For user features, we consider both the user historic interactions with pages (e.g. time spent per visit, number of clicks), the ad sensitivity reflected by the user past whitelist decisions, and the user profile (e.g. operating system). We also consider page characteristics (e.g. topic, freshness) and the context of the visit. We design a novel attention [11]-like activation unit to capture the interactions among previously visited pages, historical actions, and the intended page. In addition, a residual connection is added for historical actions to the final output in order to avoid the gradient vanishing problem and strengthen the global influence to whitelist behavior.

Furthermore, we observe that users who are willing to whitelist a page likely spend relatively longer time on the page. We propose to co-learn the whitelist prediction and the dwell-time prediction in order to enhance the model learning ability on parameter training, inspired by the use of multitask learning on page recommendation [12].

The effectiveness of DAWN is demonstrated by experiments with real-life user behavior data collected from Forbes Media, a large US news publisher, with about 67,000 events for 34,000 users over three consecutive months. This article also presents the use a DAWN-based policy for personalized counter ad-blocking in practice. We analyze how to set the decision threshold in DAWN and its impact on ad revenue and user engagement. We compare the proposed DANW-based policy with three existing policies used in practice: Wall-only, AAX-only, and Random Assignment of users to Wall or AAX. Data analysis shows that the DAWN-based policy is able to generate higher revenue than the Wall-only policy and the AAX-only policy, and also achieves higher user engagement than the Random Assignment policy.

The main contributions of the work are summarized below.

1) To the best of our knowledge, this is the first work that designed a policy for personalized counter ad-blocking. The core of this policy is to use machine learning techniques for whitelist prediction.
2) We propose DAWN, a deep learning model that makes effective whitelist predictions. We design a novel attention mechanism to represent both historical page visits and actions on pages. We also conduct co-learning of whitelist prediction and dwell time prediction in order to enhance the model learning ability on parameter training.
3) We evaluated DAWN using real-life ad-blocking user behavior data collected from a large US news publisher for three consecutive months. Experimental re-

1. https://acceptableads.com/committee/, Accessed on 05/2021

sults demonstrate that DAWN significantly outperforms comparison systems in whitelist prediction.
4) We discuss how to use DAWN-based policies in practice, setting the decision threshold appropriately for optimizing revenue and for encouraging user engagement. Data analysis shows that DAWN-based policies outperform existing policies used by publishers.

The rest of the article is organized as follows. Section 2 discusses related work. Section 3 presents the overview of the personalized counter ad-blocking policy that uses whitelist prediction. Section 4 presents the proposed model for whitelist prediction, whose experimental results are discussed in Section 5. Section 6 discusses the practical use of personalized counter ad-blocking in a real-world context. Section 7 summarizes the article and discusses future work.

2 Related Work

This section presents a literature review on ad-blocking and recommender systems.

2.1 Ad-Blocking and Counter Ad-Blocking Mechanisms

In [8], [13], the authors summarize the existing ad-blocking and counter ad-blocking techniques. Generally, ad-blocking tools (e.g., AdBlock Plus) handle ads based on matching filter rules with filter lists. If a filter rule matches a URL that is marked as an ad, the ad-blocker will prevent the web browser from requesting the URL, unless the URL belongs to a whitelisted site. In other words, the ads will be displayed only if a website or a web page is whitelisted in the ad-blocker. As part of their counter ad-blocking efforts, publishers utilize "baits" to detect the presence of ad-blockers. The "baits" are fake ads inserted in web pages, such that ad-blockers will attempt to block them. If the baits are blocked, it means the user is using an ad-blocker, and the ads in a page will not be displayed [8]. Then, the publisher may impose a counter ad-blocker Wall, which requires the users to whitelist the intended page or the site for content access. To defend against counter ad-blocking, users crowdsource rules through Github [2] in order to avoid blocking the "baits", and therefore to escape from the ad-blocker detection techniques. This ad-blocking "battle" is ongoing, and it keeps evolving.

2.2 Ad-Blocking and Counter Ad-Blocking Studies

A study that analyzed the gender and age distribution of ad-blocking users found that males under 34 years old are most likely demographic group to use ad-blockers [14]. Shiller et al. [15] explored the impact of ad-blocker usage on site-level traffic. Utilizing data from Alexa's website ranking, the authors found that each additional percentage increase of ad-blocker visitors reduces the traffic by 0.67% over 35 months on a site. Based on their calculation, the revenue declines by 20% if the ad-blocking rate is 12% because the relation between traffic and revenue is not linear and it is moderated by the website quality. The work done by Miroglio et al. [16] studied the effects of ad-blocker usage on user engagement with the web. The study concludes that ad-blocking has a positive impact on user engagement with the web (i.e., dwell time, page views). In other words, ad-blocking users tend to stay longer and have more engagement with pages compared with non-ad-blocker users.

There are also studies [7], [17] on how to measure the effect of counter ad-blocking strategies on user behavior. The major finding is that counter ad-blocking strategies have a selective influence on users with varying degrees of loyalty to the website. These studies provide insights on the reasons of ad-blocker usage and the impact ad-blocking and counter ad-blocking on user behavior.

There are only a few studies on counter ad-blocking strategies. Aseri et al. [18] proposed a theoretical model for selective-gating, where p fraction of randomly selected ad-blocker users are gated, which is functionally similar to the Wall strategy, while the rest of users are given ad-free access to the website. The focus is to determine the value of p for revenue optimization in the long run, based on the relative strength of a website's network effect and its proportion of ad-blockers. Network effect refers to the additional utility derived by a user due to other agents consuming the same good [18]. The stronger the networking effect or larger the proportion of ad-blockers in the total user base, the lower p should be. Our study has a different setting, where a user is treated either with Wall or AAX. Instead of a random assignment, we select users to be treated with Wall based on the predicted probability of whitelisting, in order to optimize revenue generation. The proposed user selection also achieves better user engagement than a random assignment. Furthermore, we validate the proposed model and demonstrate its practical benefits using real-life data. Zhao et al [6] used a gradient boost regression tree model for whitelist prediction. In our study, a novel deep neural network model is proposed, which achieves better performance in whitelist prediction, as show in Section 5. Furthermore, Zhao et al [6] does not study how to use whitelist prediction in a counter ad-blocking policy in practice or analyze its impact to revenue or user engagement.

2.3 Recommender Systems and Click-Through Prediction

Whitelist prediction is related to recommender systems, which predict user behavior based on historical behavior. Recommender systems in business settings are used to understand the preferences of users, typically using historical behavioral data, in order to make recommendations of products tailored to each individual user. Such systems are prominent in e-commerce [19], social networks [20], crowdsourcing platforms [21], and many other contexts. Based on objective functions, recommender systems are divided into three types: list-wise recommendation, pair-wise recommendation, and point-wise recommendation [22]. List-wise recommendation aims to rank a list of items, pair-wise recommendation compares the items in pair, and point-wise recommendation predicts the likelihood of a given <user, item> pair.

Given a user and an intended page to visit, the whitelist prediction is related to point-wise recommendation, in

2. https://github.com/reek/anti-adblock-killer/, Accessed on 05/2021

which the most popular application is click-though rate (CTR) prediction. CTR prediction aims to predict ad click or purchase behavior given a <user, ad> pair [23]. Collaborative filtering (both item-based [24] and user-based [25]), content-based recommendation, and matrix factorization are early techniques that have achieved wide success in the field of click-through prediction [26]. Recently, the structure of the CTR prediction models has evolved from shallow learning to deep learning. Deep learning-based models are used to improve the quality of recommendations, such as Wide&Deep [27] and DeepFM [28]. Deep Interest Networks (DINs) aim to capture a given user's diverse interests and likely actions with regards to a product by using the historical behaviors of the user. It improves prediction accuracy over previous Embedding methods by adaptively learning a representation of user interests from historical behaviors with respect to a particular product, through the state-of-the-art attention-based mechanisms in the model architecture [23].

There are several research gaps identified from the existing literature. First, while there is much research on the effect of ad-blocking and counter ad-blocking, there are limited studies on the design of effective policies that publishers can use to handle counter ad-blocking. As discussed earlier, despite of the two most related work [6], [18] (Section 2.2), it is lack of studies of using whitelist prediction in a counter ad-blocking policy, or analyzing the impact of such a policy on revenue generation or user engagement. Second, it's lack of studies on investigating advanced machine learning techniques, in particular, deep learning models, for user whitelisting behavior prediction. This study will address the following questions: How can publishers effectively handle counter ad-blocking, with dual goals of optimizing revenue and user engagement? Can counter ad-blocking policies be developed to dynamically take into consideration individual users' characteristics reflected in their historic behavior?

3 Personalized Counter Ad-Blocking

In this work, we propose a policy for personalized counter ad-blocking that dynamically chooses different strategies on different users, based on their interests and sensitivity to ads reflected in their historic behavior.

Given the popularity of the Wall strategy and the AAX strategy, we propose the policy illustrated in Figure 4. We consider user attributes, user actions on web pages in their visits to the site in the past, as well as the attributes of those pages to develop a predictive model about the likelihood of a user to whitelist an intended page. User actions and attributes are collected using web analytics services (e.g., Google Analytics). The attributes of pages visited by users can either be pulled from a publisher's content management system (CMS) or scraped from the web page itself. Based on the whitelist prediction, the Wall strategy is assigned to a target user if the user is predicted to whitelist, and the AAX strategy is assigned otherwise.

Evidently, the accuracy of whitelist prediction plays a crucial role in the success of implementing the proposed policy. We will present a deep-learning model, DAWN, for

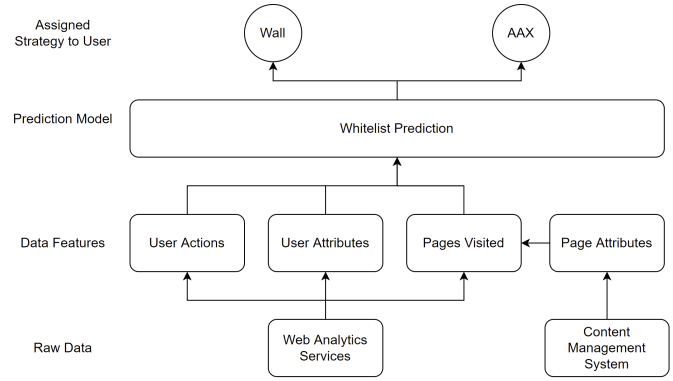


Fig. 4. A personalized counter ad-blocking framework.

whitelist prediction in Section 4 and its empirically evaluation in Section 5. Then, Section 6 will analyze proposed policy, used in conjunction with DAWN. Specifically, we will study the effect of different decision thresholds of DAWN on revenue and user engagement to provide a practical solution, and compare the DAWN-based personalized counter ad-blocking policy with policies that use only the Wall strategy or only the AAX strategy. Let us note that more complex personalized counter ad-blocking policies can be devised based on whitelist prediction. Our aim in this article is to demonstrate that even a simple policy works better in terms of revenue and user engagement than the current policies used in the publishing industry.

4 A Deep Learning-Based Whitelist Prediction Model

Formally, the whitelist prediction problem is defined as: Given an ad-blocking user U_i and an intended article P_j they want to access, the goal is to predict WL_{ij}, which denotes whether the user will whitelist (i.e., turn off or pause the ad-blocker) to view the article when facing the counter ad-blocking Wall.

To solve this problem, we develop the Deep Adblock Whitelist Network (DAWN) model for whitelist prediction. As discussed in Section 1, designing a whitelist prediction model poses several challenges due to the large data complexity and potentially hidden relationships. The section starts with a brief description of the DAWN architecture, continues with a discussion on feature engineering, describes the modeling of historical user behavior, and ends with the objective function using multi-task learning.

4.1 Model Architecture

The framework of DAWN is illustrated in Figure 5. It has four input types: user profile features, user historical behavior features, the intended page features, and the context features. The main difficulty is learning the heterogeneous behavior patterns of different users. This is similar with the problem faced by recommender systems, since user interest plays a key role in the whitelist behavior and user historical behaviors are important in modeling user interests.

DAWN leverages some ideas from DIN [23], a state-of-the-art work in e-commerce product recommendation, and

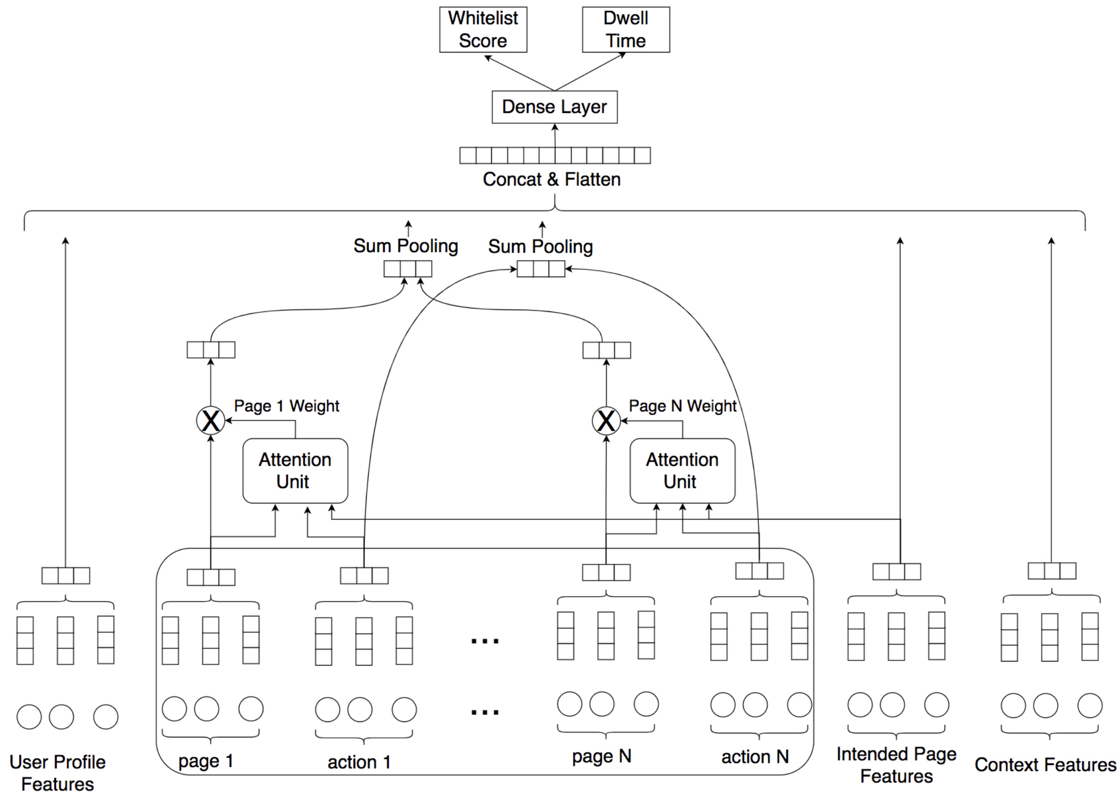


Fig. 5. The architecture of the proposed Deep Adblock Whitelist Network (DAWN).

enhances them with several novel contributions. First, while DIN only considers historical page information, DAWN also accounts for user actions (e.g., mouse clicks) on the web pages. We design a novel attention unit to represent the interaction between historical behavior and the current intended page. Attention has been shown to be effective in discovering intricate patterns in sequential modeling and flexible in learning the influence of historical data of different lengths [11]. Second, to address the gradient vanishing problem and strengthen the influence of historical behavior on output prediction, we add a residual connection [29] to propagate the historical action embeddings to the pooling layer. Third, we use multitask learning to increase the learning ability. Specifically, we observe that users who are willing to whitelist a page likely have a strong interest in the intended page and will spend a relatively longer time on the page. Therefore, DAWN learns the whitelist prediction and dwell time prediction models together.

4.2 Feature Engineering

The input of DAWN is composed of four feature types: user profiles, user historical behaviors, intended page, and context.

- *User Profile Features*
  - The user's *operating system (OS)* and *browser* information. The usage of ad-blockers is related to the user's expertise and familiarity with IT. The OS and the browser can indicate the user's level of expertise in IT. The OS includes both desktop OSs (e.g., Windows, MacOS, Linux) and mobile OSs (iOS, Android).
  - The user's *geo-location*, at the country level, which is detected from the user IP addresses. We consider user geo-locations because this is the only explicit feature about users that can be easily obtained by publishers without violating user privacy [30].
- *User Historical Behavior Features*
  - *Dwell time, number of clicks, and whitelist decision* of a user's historical interactions with the website. These features represent the most direct way of capturing the whitelist behavior.
  - *Session id* is a count of the sessions that a user had on the website in the past. A higher session number indicates a higher user interest/loyalty for the web site.
- *Page Features.*[3]
  - *Page id and article id* are considered as input features to learn the latent representation of ids and make more personalized predictions.
  - *Article popularity*, or articles "hotness", is the total number of visits received by an article from all users until the current visit.
  - *Article channel* is its topical category defined by the publisher's website, e.g., finance and lifestyle. A channel can be considered as a high-level topic label of an article.
  - *Article freshness* is the duration between the time the article was published on the website and the time the page was read. Article freshness is an important

3. We consider features of both historical pages and the intended pages

attribute of web resources and can benefit a series of time-sensitive applications about user behavior [31]. In our case, freshness is measured in days.
  - *Contributor* of an article is the author of that article. A contributor has their unique writing style and is proficient in certain topics.
- *Context Features*
  - The *traffic source* of the current visit, which is the origin of a user's visit. There are three main traffic source categories: search engines traffic, direct traffic, or referral traffic. Search engine traffic comes from visitors clicking on links in a page with search results. Direct traffic represents those visitors that type the URL in the browser or click on a bookmark or link in email, SMS, etc. Referral traffic counts those visitors that click a link on another site (e.g., social networking sites).
  - *Date & Time* of a given user's current visit. This is recorded once the web page loads in the user's browser.

The majority of these features are categorical variables, and deep learning excels at representing such variables. For discrete numerical variables (i.e., dwell time, number of actions, article freshness, popularity), we follow the common practice in deep learning for recommender systems [32], [33] of discretizing these features into categorical variables, then applying an embedding layer to capture their latent representation.

4.3 Modeling the Influence of Historical Behavior

The basic DIN framework performs as a weighted sum pooling to adaptively calculate the influence driven by historical page data. DAWN extends this attention-like mechanism in DIN by adding the modeling of actions on pages, and it separates the historical influence in two parts, i.e., historically visited pages and user actions on these pages. The user affinity representation $\mathbf{v}_U(P)$ given an intended page P to view is defined as follows.

$$\begin{aligned}\mathbf{v}_U(P) &= f(\mathbf{v}_P; \mathbf{e}_1^P, ..., \mathbf{e}_H^P; \mathbf{e}_1^A, ..., \mathbf{e}_H^A)\\ &= \sum_{j=1}^{H} a(\mathbf{v}_P, \mathbf{e}_j^A, \mathbf{e}_j^P)\mathbf{e}_j^P\\ &= \sum_{j=1}^{H} a(\mathbf{v}_P, (\mathbf{e}_j^A \odot \mathbf{e}_j^P))\mathbf{e}_j^P\\ &= \sum_{j=1}^{H} w_j \mathbf{e}_j^P\end{aligned} \tag{1}$$

where $\{\mathbf{e}_1^P, ..., \mathbf{e}_H^P\}$ and $\{\mathbf{e}_1^A, ..., \mathbf{e}_H^A\}$ are the lists of embedding vectors of historical pages and actions, respectively. $\mathbf{v}_P$ is the embedding vector of the intended page. In this way, $\mathbf{v}_U(P)$ varies for different users. $\odot$ is the Hadamard product. $a(\cdot)$ is a feed-forward network with output as the activation weight in order to capture the three-way interaction.

The details of the activation unit $a(\cdot)$ are illustrated in Figure 6. The inputs are historical actions and the associated visited pages' embedding in each historical timestamp, and the intended page embedding. The output is a scalar weight to represent the historical influence to the intended page.

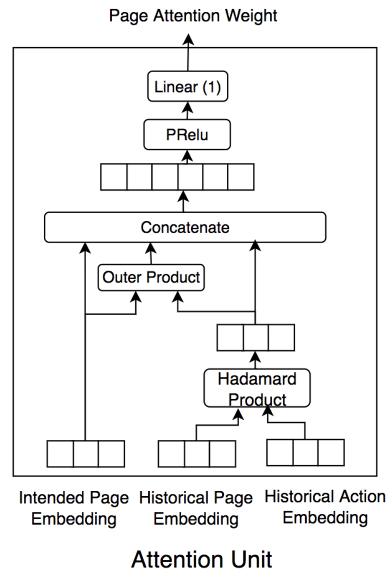


Fig. 6. The structure of attention unit, to represent the interaction between historical behavior and the current intended page.

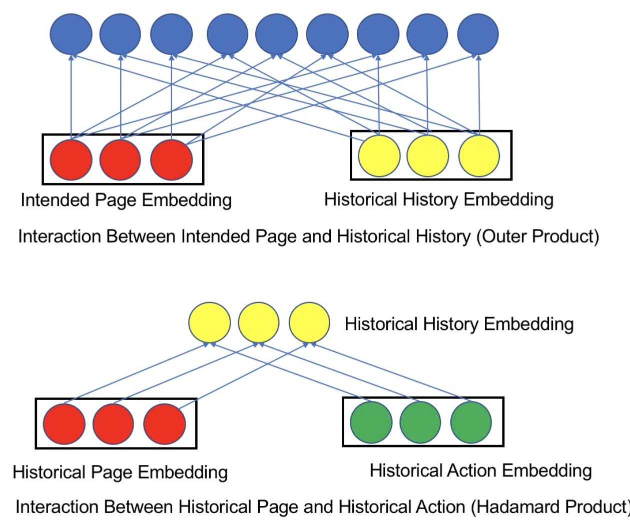


Fig. 7. Examples of two types of products used in the attention unit: outer product (top) and Hadamard product (bottom).

As shown in Figure 6, we consider two types of embedding products: Hadamard product and outer product. First, the Hadamard product between the historical page embedding and the historical action embedding is applied to capture the overall historical influence. Then, in order to represent the influence of historical behavior, we utilize the outer product between the current intended page embedding and the learned historical embedding. Here the outer product serves as data augmentation to create latent pair-feature variables. Figure 7 illustrates the two types of products between latent embeddings. PReLU [34] is a commonly used activation function, and it adds a learnable parametric factor upon the original ReLU function to enable representation flexibility. The main difference compared with the self-attention mechanism in Transformer [11] is that DAWN has a *three-way* attention among <*intended page, past page, past action*> to compute the attention weights of the past behavior on the intended page. Unlike our attention mechanism, self-attention uses query and key embeddings to compute the attention weights on values and it cannot easily handle such a three way interaction.

4.4 Objective Function

Multitask learning is a machine learning methodology which aims to solve multiple tasks simultaneously. It helps to alleviate data scarcity by extracting useful information from other related tasks [35]. Whitelisting is not a frequent event, since users often choose not to whitelist. On the other hand, we can easily capture the dwell time of a user in a page. Dwell time is useful in quantifying content relevance to a particular user [36]. Intuitively, a user tends to whitelist pages that are most relevant to their interests. Therefore, we propose to use multitask learning to simultaneously predict whitelist behavior and dwell time of a user in an intended page.

Specifically, DAWN combines whitelist prediction and dwell time prediction to compute its final optimization goal. For the whitelist prediction, the objective function is the binary cross-entropy function:

$$L_{wl} = -\frac{1}{N}\sum_{(\mathbf{x},y_{wl})\in\mathcal{S}} y_{wl} log p(\mathbf{x}) + (1-y_{wl}) log(1-p(\mathbf{x})) \tag{2}$$

in which N is the number of predictions, y_{wl} is the ground truth of the whitelist label and $p(\mathbf{x})$ is the predicted likelihood to whitelist. In the inference phase, we use a decision threshold $T \in (0,1)$ as a cutoff to convert the predicted likelihood to whitelist $p(\mathbf{x})$ to a binary prediction label: whitelist or not. We will discuss how to set the value of T in Section 6.

Dwell time is discretized into categorical bins due to its long tail characteristic. Given the ordinal order of such categorical bins, it is not suitable to be modelled as a multi-class classification problem. Instead, we minimize the loss of transformed categories of dwell times using the mean square error, defined as follows:

$$L_{dt} = -\frac{1}{N}\sum_{(\mathbf{x},y_{dt})\in\mathcal{S}} (y_{dt}-\hat{y}_{dt})^2 \tag{3}$$

where y_{dt} is the ground-truth dwell time categorical bin and $\hat{y}_{dt}$ is the predicted one. The final loss is the combinations of the two aforementioned losses.

$$L = \frac{1}{1+\alpha}L_{wl} + \frac{\alpha}{1+\alpha}L_{dt} \tag{4}$$

in which α is the weight of the whitelist prediction loss.

5 Experimental Evaluation of Whitelist Prediction

We first describe the experimental data collection, the experimental settings, the evaluation metrics, and the comparison methods. Then, we present the experimental results, along with the analysis of the impact of multitask learning, historical behavior size, and different data features on the prediction results.

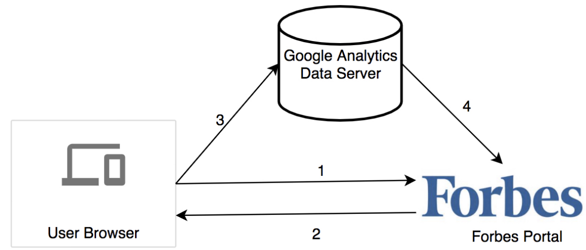


Fig. 8. Data collection platform.

5.1 Data Collection

Building prediction models for whitelist behavior requires user historical activity data, including whether an ad-blocker is used, and content attributes of any given page from the publisher's content database. We collected data from Forbes Media for three consecutive months. Each web page is an article written by a contributor to Forbes Media. We use a JavaScript program to detect the existence of ad-blockers, and discard data from non-ad-blocking users.

As Figure 3(b) shows, if the website detects an ad-blocker, the website will pop up a message "Adblock Detected", asking the user to whitelist the Forbes site (or the specific page the user intends to visit) in order to view the content. Users who refuse to whitelist will be prevented from viewing the intended content. Each record in our dataset is a user visit, which contains all the pages viewed or intended to be viewed by the user, along with the user actions on these pages.

Figure 8 illustrates the data collection procedure. The sequence of interactions are as follows: (1) the user browser requests an article from Forbes news portal, (2) the Forbes web server responds by sending the requested article together with a small JavaScript program from a 3rd party server, i.e., Google Analytics server, (3) the Javascript program will record each user interaction with the article pages and store them in Google Analytics. In other words, Google Analytics collects user behavior data from Forbes. (4) the data stored by Google Analytics can be accessed by Forbes via a user interface or data pipeline. Visitors are differentiated via HTTP cookies, and the data contain no personal information about each visitor.

5.2 Experimental Settings

The dataset contains 34,000 adblock users, with a total of 67,000 ad-blocker-detected events, in which the whitelist ratio is around 20%. The user behavior is time-ordered. Instead of using the usual cross-validation procedure with randomized allocation of events across data splits, we split the dataset into training and testing dataset by time with a ratio of 80:20. Only returned users who have appeared in the training dataset period are included in the test dataset. DAWN is implemented using Tensorflow version 2.3.0 and is built upon the DeepCTR library [37]. Based on empirical observation of model performance, the dimensions of page embeddings and action embeddings are both set to 25, and the dimension of user embeddings is set to 20. We consider the last five historical visits for each prediction (i.e. $H = 5$). The training goal is to minimize the defined loss, and we adapt an Adam optimizier with a 0.001 learning rate. We set the batch size to 32, and by default we report the average performance over five runs.

5.3 Evaluation Metrics

We use the following two metrics to evaluate the predictive models.

Logistic Loss (LogLoss): It is widely used in probabilistic classification. It gives high penalty to a method for being both confident and wrong. Lower values are better.

Area Under Curve (AUC): It is defined as the area under a receiver operating characteristic (ROC) curve. If a classifier is good, the true positive rate will increase quickly and the area under the curve will be close to 1. Higher values are better.

We choose AUC and LogLoss as metrics instead of accuracy and F1 score because they are not influenced by specific decision thresholds T to discretize the final prediction score into positives or negatives. Also, they are better metrics if the class distribution is highly imbalanced. We will discuss how to set threshold T in Section 6.

5.4 Comparison Methods

Logistic Regression (LR): Since the whitelist prediction can be considered as a classification problem, we developed a logistic regression model for comparison. The input variables are the same as those used in DAWN. LR models categorical features using one-hot encoding.

Random Forest: Random Forest is an ensemble machine learning method that uses multiple decision trees and reduces the prediction variance by averaging the prediction of individual decision trees. In order to achieve the best model performance, we use grid search on some key hyperparameters: the maximum depth of the tree, the minimum number of samples required to split an internal node, the minimum number of samples required to be at a leaf node, and the number of features to consider when looking for the best split.

XGBoost: XGBoost combines weak learners together into a strong one. In this model, instances that were misclassified by the previous learners are given higher weights when training the current learner. This was used in previous work for whitelist prediction [6]. There are more hyperparameters in XGBoost than Random Forest. Therefore, we use randomized search instead of grid search for fast tuning of the following major hyperparameters: learning rate, maximum depth of a tree, minimum child weight of further partition, minimum loss reduction required to make a further partition on a leaf node of the tree, L2 regularization term on weights, and the subsample ratio of columns when constructing each tree.

Deep Interest Network [23] (DIN): The comparison with DIN serves as an ablation study to examine the benefits of the proposed customized attention unit and the multitask learning objective. For a fair comparison, we use exact the same setting with DAWN on the training, such as the same optimizer and the same batch size. In addition to the original DIN, which uses the attention method, we also create a variation of DIN that replaces the attention method with a simple mean pooling [38], denoted at *DIN with mean pooling*, to examine the advantage of different attention mechanisms.

TABLE 1
Whitelist prediction performance of various models

Model	Model Performance	
	Log Loss	AUC
Logistic Regression	0.4060	0.7387
Random Forest	0.3995	0.7651
XGBoost	0.3903	0.7736
DIN with mean pooling	0.3979	0.7846
DIN	0.3904	0.7848
DAWN	**0.3722**	**0.8084**

5.5 Prediction Results: DAWN vs. Comparison Methods

Table 1 shows that DAWN outperforms the comparison methods. Overall, we observe that the deep learning approaches of both DAWN and DIN work better than the other methods. This demonstrates the strong pattern recognition and representation with neural networks in predicting whitelist behavior. The results show that DIN's variant with mean pooling instead of attention has an inferior performance to DIN and DAWN. This demonstrates the benefits of the attention mechanism for both DAWN and DIN. Furthermore, DAWN works the best and obtains significantly better performance than the original DIN, its variation, and the other methods.

In particular, DAWN gains 0.0236 in AUC compared to DIN. Note that the work in which DIN was proposed [23] stated that 0.001 absolute AUC gain is significant and worthy of model deployment for CTR prediction in commercial advertising systems with hundreds of millions of requests. Since whitelist prediction has similar characteristics with CTR prediction and shares the same scale of score sensitivity and traffic volume, the improvement brought by DAWN will provide significant benefits in practice.

This validates the effectiveness of our novel contributions in DAWN to model the historical behavior using the customized attention mechanism and the co-learning of whitelist prediction and dwell-time prediction. The prediction scores are in the [0, 1] interval, where 0 means negative and 1 means positive. To better illustrate the distribution of prediction scores Figure 9 plots the ROC of all models. As we can see, the performance of DAWN, represented by the black solid line, is superior to all the other methods.

5.6 Effect of Multitask Learning

In order to show the effect of co-learning with dwell time and to determine the best value for hyperparameter α, we conduct a hyperparameter search for α. We observe that generally dwell time loss is much larger than whitelist prediction loss because the former is measured by mean squared error and the latter is measured by binary output

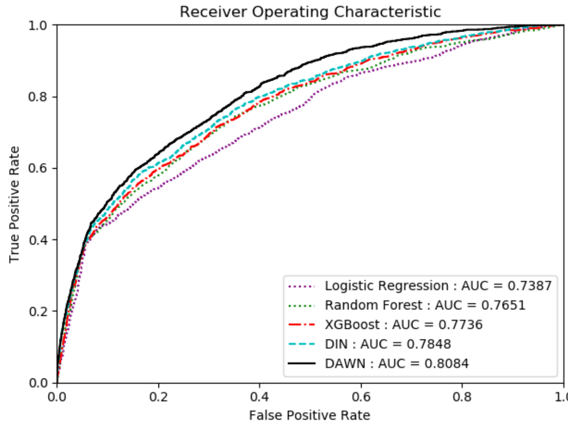


Fig. 9. ROC curve of whitelist prediction of various models.

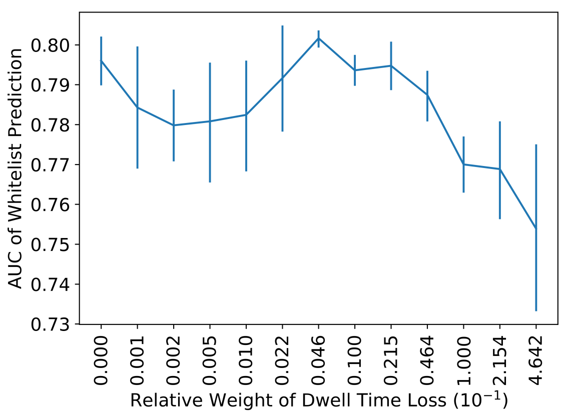


Fig. 10. The effect of weight parameter α in multi-task learning on DAWN's whitelist prediction performance.

loss. Since our main goal is to boost the performance of whitelist prediction, the weight of dwell time loss should not be large. Therefore, we choose α between $(0, 0.5)$ spaced evenly on a log scale. The model runs five times for each value of α to compute the average time and the confidence interval. The results is presented in Figure 10.

When $\alpha = 0$, DAWN performs only the whitelist prediction task, without the dwell-time co-learning. The higher performance achieved by DAWN demonstrates the effectiveness of its attention mechanism compared to the standard attention mechanism used in DIN. As α increases from 0 to 0.0002, the whitelist prediction performance degrades. It indicates that a very low weight for the dwell-time task might add noise to the whitelist task, thus resulting in inferior performance. As α further increases, we observe an inverted U shape for the whitelist prediction performance, with the peak being achieved when $\alpha = 0.0046$. Further, we notice that the confidence interval range has a roughly negative relationship with the model performance. With the optimal value of α, the co-learning between whitelist prediction and dwell-time prediction brings benefits to not only the whitelist prediction performance but also to the model stability.

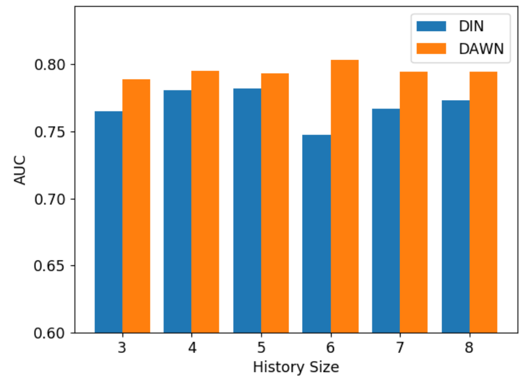


Fig. 11. DAWN's whitelist prediction performance for users of different historical data sizes.

5.7 Influence of Historical Data Size

DAWN uses user historical behavior to predict whitelist behavior. Next, we study the influence of historical behavior size on the performance of whitelist prediction. We compare DAWN with DIN, which was shown to perform the best amongst all comparison methods. The results are illustrated in Figure 11, in which the X axis shows the maximum number of historical pages used by the model for all users, and the Y axis shows the AUC score of the model performance. We noticed that 98% of users have no more than 8 history visits in the three consecutive months in the dataset. Therefore, the history size is cut off at 8, as this value covers most users. The results demonstrate that more historical behavior data leads to higher prediction performance. However, the data can overfit the model when the history size increases too much. Specifically, DIN's best performance is achieved when the history size is 5 whereas DAWN's performance peaks when the history size is 6. As can be seen, DAWN always outperforms DIN. It also takes better advantages of more historical data to further improve the performance, demonstrating the effectiveness of the proposed attention mechanism to model the influence of historical behavior to the current visit.

6 DAWN-based Personalized Counter Ad-Blocking Policy

As discussed in Section 3, we propose a DAWN-based policy for personalized counter ad-blocking based on whitelist prediction. This section discusses how to use this policy in practice.

As discussed in Section 5.3, DAWN outputs a score between 0 and 1, as the predicted likelihood of the user to whitelist the current page. If a user's prediction score is larger than the decision threshold T specified by the publisher, the user is predicted to whitelist and will be treated with the Wall strategy; otherwise, the user will be treated with the AAX strategy. Thus, fewer users will be treated with the Wall strategy for higher values of T. Figure 12 shows the percentage of users that will be treated with the Wall strategy when the threshold T varies from 0.0 to 0.45. When $T = 0.0$, about 80% of users are treated with the Wall strategy, and the remaining users, who have the prediction

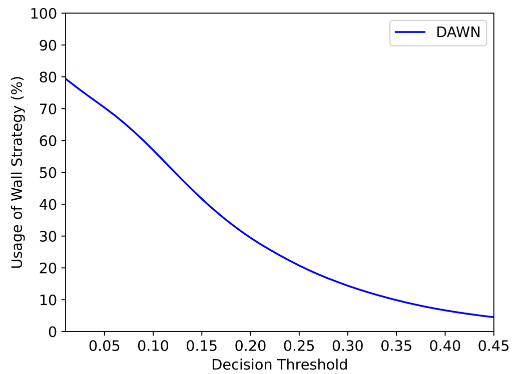


Fig. 12. The percentage of users assigned to the Wall strategy at varying values of the decision threshold T.

TABLE 2
Confusion matrix of whitelist prediction

		Ground Truth	
		Whitelist	Leave
Prediction	Wall	True Wall (TW)	False Wall (FW)
	AAX	False AAX (FA)	True AAX (TA)

score equal to or less than 0.0, are treated with the AAX strategy. Next, we discuss how the choices of T impact the revenue (Section 6.1) and the user engagement (Section 6.2).

6.1 Impact on Revenue

This section discusses how to choose the threshold T in DAWN to optimize the revenue when using the DAWN-based policy for personalized counter ad-blocking in practice. We also compare the DAWN-based policy with the policies that use Wall on all ad-blocking users or AAX on all ad-blocking users.

To analyze the impact of the threshold T on revenue generation, we first discuss four result cases, as shown in Table 2, similar to the confusion matrix of a typical classification task. True Wall means that DAWN correctly predicts that a user will whitelist if treated with Wall. True AAX means that DAWN correctly predicts that a user will not whitelist if treated with Wall, and thus this user should be treated with AAX. True Wall and True AAX are desirable cases since the model correctly predicts the outcome and an optimal strategy is used. The opposite is true for False AAX and False Wall. In the False AAX scenario, an adblocker user is served AAX without asking to whitelist. However, in fact, the user would be willing to whitelist if the Wall strategy is used. Since AAX generates less revenue than regular ads after whitelist, such situation will result in less-than-optimal revenue. In the False Wall scenario, an ad-blocking user is served Wall, but the user refuses to whitelist, and consequently is forced to leave the website. This is a loss-loss situation for both the publisher who wishes to maximize the revenue and the user who wishes to access the content. Thus, False Wall is the worst case scenario which publishers should aim to avoid.

Next, we discuss the revenue generated in each of the four cases. Let R_1 denote the revenue generated by regular ads in a page, and R_2 denote the revenue generated by AAX ads in a page. The revenue generated in each of the four cases is:

- True Wall, i.e., Wall is displayed, then the ad-blocking user whitelists, and Regular ads are served: R_1
- True AAX, i.e., No Wall is displayed and the ad-blocking user goes to AAX directly, with AAX ads served: R_2
- False Wall, i.e., Wall is displayed, then the ad-blocking user refuses to whitelist, and consequently leaves the site: 0
- False AAX, i.e., No Wall is displayed, and the ad-blocker user goes to AAX directly, with AAX ads served (though the user would whitelist if facing the Wall): R_2

The expected ad revenue per page is calculated as follows:

$$revenue = \frac{n_{TW} \cdot R_1 + n_{TA} \cdot R_2 + n_{FA} \cdot R_2 + n_{FW} \cdot 0}{n_{TW} + n_{TA} + n_{FA} + n_{FW}} \tag{5}$$

Here $n_{()}$ is the number of visits in each corresponding case, True Wall (denoted as TW), True AAX (TA), False Wall (FW) and False AAX (FA).

The value of threshold T affects how many visits will fall into each of four cases. Hypothetically, if the threshold T is set to be larger than 1, all ad-blocker users will see AAX ads and generate revenue R_2 per page. On the other hand, if the threshold T is set to be 0, all ad-blocker users would be directed to the Wall strategy. Figure 12 shows the percent of users that will be treated with the Wall strategy when the threshold T varies.

We now discuss how to set the threshold T with the goal of maximizing the revenue. The optimal value of the threshold T depends on the ratio between R_1 and R_2. According to the discussions with our publisher collaborators, the ratio between R_2 and R_1 is typically smaller than 0.4. To help publishers set the threshold T, we plot Figure 13, where each dashed line corresponds to a ratio between R_1 and R_2, ranging from 0.1 to 0.4, where real data is used to determine the number of visits in each of four cases. The peak of each dashed line is achieved at the optimal threshold value.

Figure 13 also shows that as the ratio of AAX ads revenue and regular ads revenue increases, the optimal value of the threshold T also increases. For $R_2 = 0.1R_1$, $R_2 = 0.2R_1$, $R2 = 0.3R1$, and $R2 = 0.4R1$ the best threshold T is 0.07, 0.24, 0.44, and 0.44, respectively. It indicates that the proposed policy will direct more adblock users to AAX when AAX earns a relatively higher revenue.

Now we compare a DAWN-based policy with the policy that uses the Wall strategy on all users (denoted as “Wall") and the policy that uses the AAX strategy on all users (denoted as “AAX”). In the figure, the black solid line represents the revenue generated by Wall. There is a diamond point at the end of each dashed line, which represents the revenue generated by AAX. As we can see, the peaks of

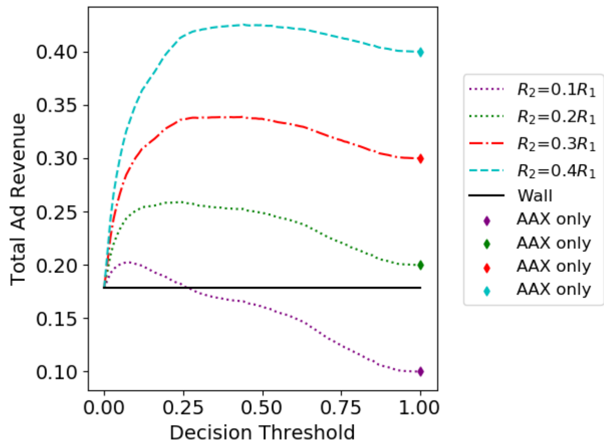


Fig. 13. Comparing the DAWN-based policy with the Wall strategy in terms of ad revenue at varying values of the decision threshold T.

the dashed lines are always above the black solid lines and the diamond points. This indicates that, when the threshold T is properly set, the proposed DAWN-based counter ad-blocking policy generates more revenue than the policies that use the Wall strategy or the AAX strategy on all the users.

6.2 Impact on User Engagement

One possible concern of a DAWN-based policy, which maximizes revenue, is that it may have a short-term outlook. A DAWN-based policy may increase revenue by assigning a large proportion of users to the Wall strategy because a whitelisted session generates more revenue than a session under AAX. Over-utilizing the Wall strategy may irritate users, who are averse to ads, by forcing them to whitelist and compromising user experience by showing regular ads, ultimately causing a fall in user engagement. This, in turn, leads to traffic reduction and revenue loss in the long run when adblock users choose not to return to the website. To study the effects of our proposed solution on user engagement, we conduct a study about the impact of the DAWN-based policy on user engagement by comparing it with a policy that randomly assigns an adblock user to Wall or AAX.

For the analysis, we collected data from the the same publisher who deployed a random assignment policy where each user has a 50% likelihood to be treated with the Wall strategy and a 50% likelihood to be treated with the AAX strategy. This is a different dataset from the one used in Section 5, which uses Wall policy on all the users in the study. This dataset consists of about 550,000 sessions and more than 200,000 unique visitors over 39 days. The format of the dataset is similar to the one used in Section 5.

Since the DAWN-based policy has yet to be deployed in production, we propose the following methodology to obtain the data that represent the impact of the DAWN-based policy on user engagement. Referring to Table 3, there are 4 types of users: A, B, C and D. User A is treated with Wall using the random assignment policy, but it is suggested to be treated with AAX by the DAWN-based policy. User B is treated with AAX using the random assignment policy, and it is also suggested to be treated with AAX by the DAWN-based policy. Users C and D are defined accordingly. Although we do not have user engagement behavior data under the DAWN-based policy, we propose to use the data of users B and C under the random assignment policy to study the DAWN-based policy, since these users are actually treated with the same strategy that the DAWN-based policy would choose. Note that the sample size is non-trivial, the subset of users B and C consists of about 250,000 sessions in total.

TABLE 3
Illustration of the methodology for comparing the DAWN-based policy with random assignment policy

	Random Assignment Policy	DAWN-Based Policy
User A	Wall	AAX
User B	AAX	AAX
User C	Wall	Wall
User D	AAX	Wall

To make a fair comparison between the DAWN-based policy and the random assignment policy, the number of users treated with Wall or AAX should be the same under either policy. We achieve this by choosing a subset of the whole dataset generated by the random assignment policy for comparison. More specifically, given a chosen value of decision threshold T (defined in Section 4.4), we denote the percentage of user sessions that are treated with Wall under the DAWN-based policy as w. Then, we randomly select a subset of data among all the data generated in the random assignment policy, such that $w\%$ of sessions are treated with Wall and the remaining $(1-w)\%$ of sessions are treated with AAX. This subset of the data, representing the effect of the random assignment policy, is compared with the subset of data corresponding to users B and C, representing the effect of the DAWN-based policy.

User engagement is measured with three key performance indicators (KPIs) in a session, as shown in Equation 6: the number of hits (i.e., actions such as mouse clicks and scrolls), the number of page views, and the dwell time. All KPIs are available in the dataset and are important predictors of ad viewership and click-through rates (CTR), which is a major revenue metric for online publishers. On the one hand, each KPI plays a distinct role to help understand user behavior. The number of hits helps inform publishers about the interactions that a user has on pages; the number of page views shows how many articles the users are reading; the dwell time reflects the time span for each article. On the other hand, considering the Pearson correlation coefficients [39] of the three KPIs, as shown in Table 4, it is evident that they are correlated. This is intuitive because more hits often correlates to the user having a longer dwell time, and a longer dwell time often corresponds to the user viewing more pages.

For each KPI in a session n, we first perform a min-max normalization [40] among all the sessions to transform its value into the interval (0, 1]. Then we calculate the engagement score of the session n by taking the average of its three normalized KPIs values. Since all KPIs are important and

TABLE 4
Correlation matrix of three engagement key performance indicators

	Page Views	Hits	Dwell Time
Page Views	1.00		
Hits	0.62	1.00	
Dwell Time	0.27	0.49	1.00

yet correlated, they are assigned equal weights.

$$engagement_n = \frac{hits_n + views_n + dwellTime_n}{3} \quad (6)$$

The average engagement score across all sessions is used to compare the effect of the random assignment policy and the DAWN-based policy on user engagement. Figure 14 shows the percentage increase in user engagement score when a DAWN-based policy is used instead of the random assignment policy, across different values of the decision threshold T from 0.0 to 0.6. Recall that for each value of T, a $w\%$ of users are selected by the DAWN-based policy to be treated with the Wall strategy; the same percentage of users, w, are randomly selected by the random assignment policy to be treated with the Wall strategy.

First, we observe that for any target revenue and its corresponding threshold value, the DAWN-based policy consistently achieves higher user engagement than the random assignment policy. The advantage of the DAWN-based policy lies in its ability to assign Wall to users based on their likelihood of whitelisting. In other words, users selected by the DAWN-based policy for Wall treatment are more likely to whitelist and therefore have engagement, compared to users selected by the random assignment policy for Wall treatment. Second, we observe that as the threshold value increases, the general trend is that the advantage of DAWN-based policy decreases. This is because, as threshold T increases, the number of users assigned to the Wall strategy decreases (see Figure 12). For example, when a threshold of 0.6 is used, less than 1% of users are assigned the Wall.

Referring to Figure 13, we identify the optimal threshold values for different revenue ratios between regular ads and AAX ads (R_1 and R_2). We marked these threshold values on the curve in Figure 14 to show the relationship between revenue and engagement. As it can be seen, while a DAWN-based policy always leads to higher user engagement compared to a random assignment, the benefits are especially significant when AAX ads revenue is much less than the regular ads revenue ($R_2 = 0.1R_1$). In such a situation, a publisher is more likely to use Wall on a larger portion of users, where strategic user selection makes a big difference.

To summarize, as shown in the analysis in this section, the DAWN-based policy can generate more revenue compared to the Wall or AAX strategies alone. Given a revenue target, we can set the decision threshold in DAWN accordingly. At any chosen threshold, DAWN achieves better user engagement than randomly assigning users to the Wall.

6.3 Discussions of Model Deployment

When deploying the model into production, there are a few key considerations to be discussed as follows.

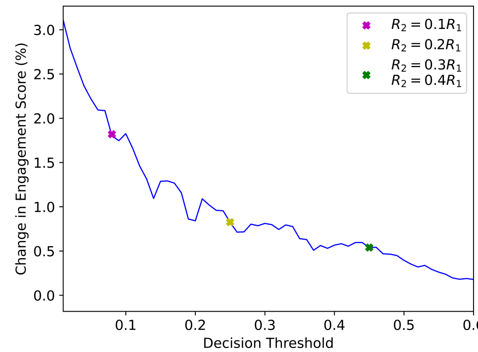


Fig. 14. Comparing the DAWN-based policy with the random assignment strategy in terms of engagement at varying values of the decision threshold T.

The data that are used in this work (as described in Section 4.2)) is available for typical online publishers, and thus the proposed model can be deployed into production. The user profile, user historical behavior, and context features are collected and supplied by Google Analytics [4], a free tool for any website owner to help them collect and analyze how visitors engage with their website. The page features are collected by publishers and are stored in their local database. All are commonly used data across publishers and, thus, the DAWN model can easily be adopted by publishers.

DAWN achieves fast training and inference time. The experiments presented in Section 5 were conducted using a laptop with four 2.3GHz CPU cores and 8 GB memory. It takes around 5-10 minutes to train the model to converge for our dataset. Also, it takes about 6ms on average to make a prediction on an incoming user request, which satisfies the latency requirements for serving web pages.

When deploying the model in production, with the increased number of users and visited pages, the model size gets larger and, thus, the inference latency also increases. To handle such a case, we first recommend the use of web servers with high computational power (such as AWS Inferentia [5]). Second, we suggest to conduct a part of the inference computation offline, which is a general practice adopted to deploy deep learning models in the industry. As mentioned in Section 4.2, all the input features are (or are converted to) categorical features and, then, the model applies embedding layers upon them to create the latent representation. These embedding layers of input features can be pre-computed offline, and then the feature embeddings can be stored in memory. During the real-time serving of web traffic, DAWN retrieves the feature embeddings directly from the memory, and then performs the attention computation in the later part of the model.

The model will be continuously updated based on the newly available data. We recommend to train the model daily. But if that brings computational burden to publishers with limited resource, the model can be updated less frequently based on the resource availability. After all, users typically shift their interests over a period of time rather on

4. https://analytics.google.com/
5. https://aws.amazon.com/machine-learning/inferentia/

a daily basis. We will explore existing techniques [41] in the future for efficient continual learning.

DAWN takes advantage of user historical behavior. Thus, it cannot be directly applied to users who do not have any recorded history, i.e. cold-start users, which is a problem similar the one encountered in typical recommender systems. One option to solve this problem is to serve the random assignment policy for first-time users. Another option is to select the policy based on only user profile features, page features, and context features by adapting existing work for the cold-start user problem [42]. Once users accumulate enough historical data, then, the system can switch them to the DAWN-based policy.

7 Conclusions and Future Work

This article proposed a personalized dynamic counter ad-blocking approach for online publishers, faced with a rapid increase in ad-blocker usage. The essence of the proposed approach is to choose counter ad-blocking strategies dynamically for each user, and to personalize these strategies according to user interests in the intended pages and sensitivity to ads. There are two widely used counter ad-blocking strategies: Wall strategy that requires users to whitelist for content access, and AAX strategy that serves acceptable ads instead of regular ads. We illustrated the proposed approach with a policy that uses the Wall strategy for the adblock users who are predicted to whitelist the web page/site, and the AAX strategy for the rest of adblock users.

The core of the proposed policy requires effective prediction of whether a user is willing to whitelist the intended page or site in the face of counter ad-blocking Wall. We developed an innovative deep learning model, DAWN, for whitelist prediction. DAWN uses an attention mechanism to capture historical page visit information, which comprehensively captures user heterogeneity. It further leverages multitask learning of related tasks, namely, whitelist prediction and dwell time prediction, in order to enhance the model learning ability on parameter training and boost the performance. Empirical studies on real-world data show that DAWN is highly effective in whitelist prediction.

The article also discussed how the decision threshold of DAWN impacts the ad revenue and user engagement. This provide insights to publishers on how to set the proper threshold value in practice to achieve good revenue and good user engagement. Furthermore, we compared the proposed DANW-based policy with three commonly used policies: Wall-only, AAX-only, and Random Assignment of users to Wall or AAX. The DAWN-based policy is able to generate higher revenue than the Wall-only policy and the AAX-only policy, and at the same time, achieves higher user engagement compared to Random Assignment. Although this article is based on the data provided by one publisher, given that data sources and settings are similar across online publishers, the proposed model and findings are applicable to online publishers in general.

More complex dynamic personalized policies based on DAWN can be devised in the future. Such policies can choose among other strategies as defined by a publisher toward a desired optimization goal, using the whitelist prediction made by DAWN.

Another future direction is to adapt the proposed DAWN model to other applications in point-wise recommendations. For instance, we can study the possibility of adapting DAWN to a dynamic personalized paywall erected on the publishers' websites. DAWN's ability to assign personalized strategies to users based on user historical behavior and the current intent can be useful in "paywall-like" settings, which are commonly found in online publishing industry.

Acknowledgement

This work is partially supported by NSF under grant No. DGE 1565478 and DGE 2043104, by a Leir Foundation endowment, a Faculty Seed Grant from the Henry J. and Erna D. Leir Research Institute for Business, Technology, and Society, and a Martin Tuchman'62 endowment at NJIT. Any opinions, findings, and conclusions expressed in this material are those of the authors and do not necessarily reflect the views of the funding agencies.

References

[1] M. McLellan, "Survey: Independent local online publishers rely heavily on ad revenue," http://www.knightdigitalmediacenter.org/blogs/mclellan/2014/05/survey-independent-local-online-publishers-rely-heavily-advertising-revenue.html, 2014.

[2] J. Holcomb and A. Mitchell, "Revenue sources: A heavy dependence on advertising," http://www.journalism.org/2014/03/26/revenue-sources-a-heavy-dependence-on-advertising/, 2014.

[3] IAB, "Global digital ad spending 2019," https://www.emarketer.com/content/global-digital-ad-spending-2019, 2019.

[4] J. Davies, "Uh-oh: Ad blocking forecast to cost $35 billion by 2020," https://digiday.com/uk/uh-oh-ad-blocking-forecast-cost-35-billion-2020/, 2017.

[5] E. Sweeney, "30% of top publishers adopt 'anti-ad-blockers,' new study finds," https://www.marketingdive.com/news/30-of-top-publishers-adopt-anti-ad-blockers-new-study-finds/516729/, 2018.

[6] S. Zhao, A. Kalra, C. Wang, C. Borcea, and Y. Chen, "Ad blocking whitelist prediction for online publishers," in *2019 IEEE International Conference on Big Data (Big Data)*. IEEE, 2019, pp. 1711–1716.

[7] S. Zhao, A. Kalra, C. Borcea, and Y. Chen, "To be tough or soft: Measuring the impact of counter-ad-blocking strategies on user engagement," in *Proceedings of The Web Conference 2020*, 2020, pp. 2690–2696.

[8] U. Iqbal, Z. Shafiq, and Z. Qian, "The ad wars: retrospective measurement and analysis of anti-adblock filter lists," in *Proceedings of the 2017 Internet Measurement Conference*. ACM, 2017, pp. 171–183.

[9] R. J. Walls, E. D. Kilmer, N. Lageman, and P. D. McDaniel, "Measuring the impact and perception of acceptable advertisements," in *Proceedings of the 2015 Internet Measurement Conference*. ACM, 2015, pp. 107–120.

[10] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[11] 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*, 2017, pp. 5998–6008.

[12] P. Jenkins, "Clickgraph: Web page embedding using clickstream data for multitask learning," in *Companion Proceedings of The 2019 World Wide Web Conference*, 2019, pp. 37–41.

[13] E. Pujol, O. Hohlfeld, and A. Feldmann, "Annoyed users: Ads and ad-block usage in the wild," in *Proceedings of the 2015 Internet Measurement Conference*. ACM, 2015, pp. 93–106.

[14] S. Zhao, C. Wang, A. Kalra, L. Vaks, C. Borcea, and Y. Chen, "Ad blocking and counter-ad blocking: Analysis of online ad blocker usage," in *Americas Conference on Information Systems*, 2017.

[15] B. Shiller, J. Waldfogel, and J. Ryan, "The effect of ad blocking on website traffic and quality," *The RAND Journal of Economics*, vol. 49, no. 1, pp. 43–63, 2018.

[16] B. Miroglio, D. Zeber, J. Kaye, and R. Weiss, "The effect of ad blocking on user engagement with the web," in *Proceedings of the 2018 World Wide Web Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 2018, pp. 813–821.

[17] A. R. Sinha, M. Macha, P. Maneriker, S. Khosla, A. Samdariya, and N. Singh, "Anti-ad blocking strategy: Measuring its true impact," in *Proceedings of the ADKDD'17*. ACM, 2017, p. 3.

[18] M. Aseri, M. Dawande, G. Janakiraman, and V. S. Mookerjee, "Ad-blockers: A blessing or a curse?" *Information Systems Research*, vol. 31, no. 2, pp. 627–646, 2020.

[19] Y. Pan and D. Wu, "A novel recommendation model for online-to-offline service based on the customer network and service location," *Journal of Management Information Systems*, vol. 37, no. 2, pp. 563–593, 2020.

[20] J. Guo, W. Zhang, W. Fan, and W. Li, "Combining geographical and social influences with deep learning for personalized point-of-interest recommendation," *Journal of Management Information Systems*, vol. 35, no. 4, pp. 1121–1153, 2018.

[21] J. Mo, S. Sarkar, and S. Menon, "Know when to run: Recommendations in crowdsourcing contests," *MIS quarterly*, vol. 42, no. 3, pp. 919–944, 2018.

[22] Z. Cao, T. Qin, T.-Y. Liu, M.-F. Tsai, and H. Li, "Learning to rank: from pairwise approach to listwise approach," in *Proceedings of the 24th international conference on Machine learning*, 2007, pp. 129–136.

[23] G. Zhou, X. Zhu, C. Song, Y. Fan, H. Zhu, X. Ma, Y. Yan, J. Jin, H. Li, and K. Gai, "Deep interest network for click-through rate prediction," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 1059–1068.

[24] Y. Pan, D. Wu, and D. L. Olson, "Online to offline (o2o) service recommendation method based on multi-dimensional similarity measurement," *Decision Support Systems*, vol. 103, pp. 1–8, 2017.

[25] Z. Zhang, Y. Kudo, and T. Murai, "Neighbor selection for user-based collaborative filtering using covering-based rough sets," *Annals of Operations Research*, vol. 256, no. 2, pp. 359–374, 2017.

[26] P. V. Kumar and V. R. Reddy, "A survey on recommender systems (rss) and its applications," *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 2, no. 8, pp. 5254–5260, 2014.

[27] H.-T. Cheng, L. Koc, J. Harmsen, T. Shaked, T. Chandra, H. Aradhye, G. Anderson, G. Corrado, W. Chai, M. Ispir *et al.*, "Wide & deep learning for recommender systems," in *Proceedings of the 1st workshop on deep learning for recommender systems*, 2016, pp. 7–10.

[28] H. Guo, R. Tang, Y. Ye, Z. Li, and X. He, "Deepfm: a factorization-machine based neural network for ctr prediction," in *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, 2017, pp. 1725–1731.

[29] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[30] C. Wang, S. Zhao, A. Kalra, C. Borcea, and Y. Chen, "Webpage depth viewability prediction using deep sequential neural networks," *IEEE Transactions on Knowledge and Data Engineering*, 2018.

[31] N. Dai and B. D. Davison, "Freshness matters: in flowers, food, and web authority," in *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2010, pp. 114–121.

[32] H. Lee, J. Im, S. Jang, H. Cho, and S. Chung, "Melu: Meta-learned user preference estimator for cold-start recommendation," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 1073–1082.

[33] P. Covington, J. Adams, and E. Sargin, "Deep neural networks for youtube recommendations," in *Proceedings of the 10th ACM conference on recommender systems*, 2016, pp. 191–198.

[34] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.

[35] K.-H. Thung and C.-Y. Wee, "A brief review on multi-task learning," *Multimedia Tools and Applications*, vol. 77, no. 22, pp. 29 705–29 725, 2018.

[36] X. Yi, L. Hong, E. Zhong, N. N. Liu, and S. Rajan, "Beyond clicks: dwell time for personalization," in *Proceedings of the 8th ACM Conference on Recommender systems*, 2014, pp. 113–120.

[37] W. Shen, "Deepctr: Easy-to-use,modular and extendible package of deep-learning based ctr models," https://github.com/shenweichen/deepctr, 2017.

[38] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

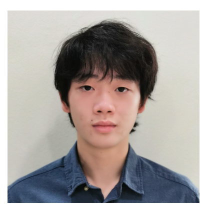
[39] O. J. Dunn, *Applied statistics: analysis of variance and regression (A Wiley publication in applied statistics)*. Wiley, 1974.

[40] L. Al Shalabi, Z. Shaaban, and B. Kasasbeh, "Data mining: A preprocessing engine," *Journal of Computer Science*, vol. 2, no. 9, pp. 735–739, 2006.

[41] M. Al-Ghossein, T. Abdessalem, and A. Barre, "A survey on stream-based recommender systems," *ACM Computing Surveys (CSUR)*, vol. 54, no. 5, pp. 1–36, 2021.

[42] P. B. Thorat, R. Goudar, and S. Barve, "Survey on collaborative filtering, content-based filtering and hybrid recommendation system," *International Journal of Computer Applications*, vol. 110, no. 4, pp. 31–36, 2015.

Shuai Zhao is currently an Applied Scientist at Amazon. This work was done while he was at NJIT. His research interests are data mining and machine learning for business applications. Shuai received a Ph.D. degree in business data science at New Jersey Institute of Technology (NJIT) in 2020.



Michael K. Chen is a student at the Raffles Institution and is also affiliated with NJIT. His research interests include the applications of data science and econometrics in management, marketing and mass communication.

Cristian Borcea is a Professor and the Associate Dean for Strategic Inititatives in the Ying Wu College of Computing at NJIT. He also holds a Visiting Professor appointment at the National Institute of Informatics in Tokyo, Japan. His research interests include: mobile computing & sensing; ad hoc & vehicular networks; and cloud & distributed systems. Cristian received his PhD from Rutgers University, and he is a member of ACM and IEEE.

Yi Chen is the Martin Tuchman '62 Chair and a Professor in Martin Tuchman School of Management with a joint appointment in the Department of Computer Science at NJIT. Her research interests include machine learning and natural language processing for business and healthcare applications. She received her Ph.D. degree from the University of Pennsylvania.