Ad Blocking Whitelist Prediction for Online Publishers

Shuai Zhao[1], Achir Kalra[2], Chong Wang[3], Cristian Borcea[1], and Yi Chen[1]

[1]New Jersey Institute of Technology, Newark, NJ, USA
[2]Forbes Media, NJ, USA
[3]S&P Global, NY, USA
sz255@njit.edu, akalra@forbes.com, chong.wang@spglobal.com, borcea@njit.edu, yi.chen@njit.edu

***Abstract*—The fast increase in ad blocker usage results in large revenue loss for online publishers and advertisers. Many publishers initialize counter-ad-blocking strategies, where a user has to choose either whitelisting the publisher's web site in their ad blocker or leaving the site without accessing the content. This paper aims to predict the user whitelisting behavior, which can help online publishers to better assess users' interests and design corresponding strategies. We present several techniques for personalized whitelist prediction for a target user and a target web page. Our prediction models are evaluated on real-world data provided by a large online publisher, Forbes Media. The best prediction performance was achieved using the gradient boosting regression tree model, which also demonstrated robustness and efficiency.**

***Index Terms*—online advertising, user behavior prediction, ad blocking, gradient boosting regression tree**

I. Introduction

Digital technologies and the Internet have dramatically changed the content publishing industry. Today, most content on the Internet is free, and the publishers make profit through digital advertising, the primary revenue source [1], [2]. Online advertising generated an earning of over \$100 billion in 2018 [3].

As figure 1 shows, there are three stakeholders in free web publishing. The users view free content and "pay" with their attention for ads displayed in the web pages. The publishers spend money to generate content and receive ad revenue. The 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 good-quality content for free. Publishers can reach out to a much broader audience than ever before, with potential to have a far-reaching impact. Advertisers deliver ads that are targeted to individual users with potential benefits of enhancing user shopping experience and achieving better marketing effectiveness with lower cost [4].

Excessive ads, however, can be annoying. More and more users opt to use ad blockers, which are tools (typically a browser plugin) that remove ads while a user is reading online content. According to a 2018 report, ad blocker usage has increased by 30% in 2017 and was expected to cause a loss of \$35 billion globally in online advertising revenue by 2020. Over 40% of the US users have used ad blockers [5].

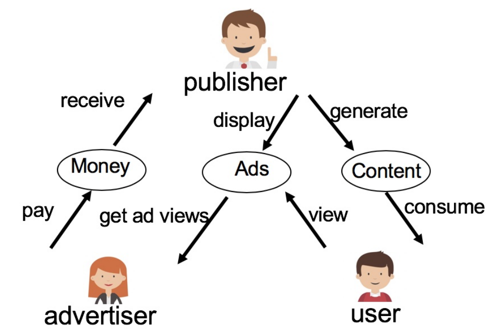

Fig. 1. Stakeholder relations in ad-supported free web publishing ecosystem. Publishers are intermediaries between advertisers and users

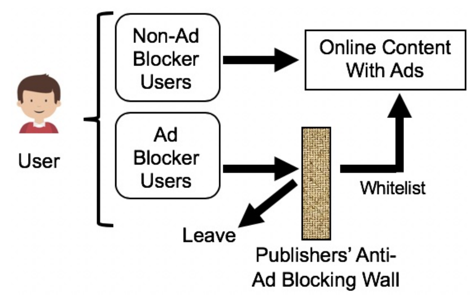

Fig. 2. Example of counter-ad-blocking strategy used by online publishers

The broad usage of ad blockers causes a severe threat to publishers' revenue. As a result, many online publishers are struggling to produce content with a tighter budget and find it difficult to survive in business. In face of significant revenue loss, online publishers launch counter-ad-blocking methods. A recent study found that counter-ad-blocking scripts were used by more than 30% of the 1,000 most popular domains [6].

The most popular counter-ad-blocking strategy is illustrated in Figure 2. When an ad blocker is detected, a publisher pops up a message requesting the user to turn off or pause the ad blocker, i.e. *whitelist* the publisher's website or the specific page the user intends to view (see Figure 3). If the user rejects the request, s/he is forbidden access to the content. However, this explicit counter-ad-blocking wall can be irritating to many users. According to the dataset used in this paper, more than 60% of the ad blocker users choose not to whitelist, and leave the website. This is a loss-loss situation: publishers lose potential users and the corresponding revenue, while users are not able to view the intended content. The ongoing ad blocking "battle" between the ad blocker users and the publishers can break the ad-supported free web ecosystem.

This paper studies the prediction of user whitelisting behavior. The findings of our studies can help online publishers to better assess users' needs and to design counter-ad-blocking strategies to boost revenue, contributing to sustaining an economically viable free web publishing model.

If the publisher is able to accurately predict that an ad blocking user will refuse to whitelist, instead of blocking access, the publisher may choose to allow the user to view a page with fewer ads that are deemed acceptable by ad blockers[1]. Even though this approach will generate less revenue than showing the web page with normal ads, it is better than losing such users and not generating any revenue. More importantly, allowing the users to access the content may make them come back to the website and agree to whitelist in the future, if they find the content valuable. On the other hand, if the publisher predicts that a user will whitelist, it can continue with the regular web pages, without compromising ad revenue.

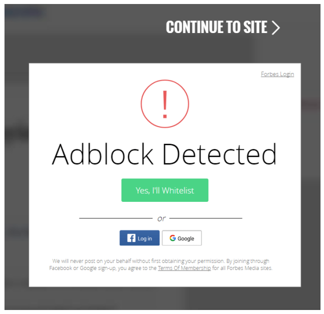

Fig. 3. The counter-ad blocking wall on the Forbes website. Ad blocker users can only access the page content if they whitelist the Forbes website

The challenge in predicting user whitelisting behavior is how to model the interaction between users and pages, and how to extract features relevant to whitelisting behavior. There are both categorical and numerical features, and combining these types of features is key to feature engineering. In this paper, we present several models that, given a target ad block user and a web page, predict the likelihood of the user to whitelist. The models are evaluated on a real-world dataset provided by Forbes Media. Our gradient boosting regression tree model achieves the best prediction accuracy due to its strong ability to model categorical and numerical variables together. The empirical evaluation also demonstrates that whitelist prediction can be done in real-time when a user attempts to visit a web page, without significant delays that compromise user experience.

To summarize, our main contributions are:

- To the best of our knowledge, this is the first work on user whitelist behavior prediction using data analytics.
- We have implemented predictive models that achieve good prediction accuracy.
- The extensive empirical evaluation on a real-world dataset verified the effectiveness of our proposed models.

The rest of the paper is organized as follows. Section II discusses the related work. Section III presents the proposed models for whitelist prediction. Experimental results are discussed in Section IV. Section V discusses the future work and concludes the paper.

II. Related Work

We are unaware of any work that addresses ad blocker whitelist prediction. Therefore, this section discusses recent works related to this topic.

In [7], [8], the authors summarize ad blocking and counter-ad-blocking techniques. Generally, ad blocker 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. However, if the URL belongs to a whitelisted site, the ad blocker does not take any action. Hence, if a website or a web page is whitelisted in the ad blocker, their ads will be displayed.

To detect the presence of ad blockers, publishers can utilize "baits". The "baits" are fake ads inserted in web pages such that ad blockers will attempt to block them. This counter-ad-blocking technique checks whether these baits are blocked [8]. If yes, the publishers may show an counter-ad-blocking wall. To defend against counter-ad-blocking, users crowdsource rules through Github [2] in order to avoid blocking the "baits", and thus to escape from the ad blocker detection techniques. This ad blocking "battle" is ongoing, and its associated methods keep evolving.

Our previous work performed a study of the factors that influence ad blocker usage based on a large-size dataset obtained from Forbes Media [9]. We used data analytics to identify several factors that are correlated with ad blocker usage, such as gender and age group. Shiller et al. [10] explored the impact of ad blocker usage on site-level traffic. Utilizing data from Alexa's website ranking, the authors find 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. [11] studies the effects of ad blocker usage on user engagement with the web. The authors use propensity score matching to reduce the bias between the treatment group and the control group. They conclude that ad blocking has a positive impact on user engagement with the web (i.e., dwell time, page views). In other words, ad blocker users tend to stay longer and have more engagement with pages compared with non-ad blocker users.

Sinha et al. [12] propose a difference-in-difference method to measure the effect of the whitelist-or-leave counter-ad-blocking strategy, which we illustrated in Figure 2). The

[1]https://acceptableads.com/

[2]https://github.com/reek/anti-adblock-killer/

authors find that there is a negative impact of the whitelist-or-leave strategy on the overall user engagement, and consequently web traffic. This result reflects the need of better strategies on the publishers' side.

III. Whitelist Prediction

Problem Definition. *Given an ad blocker user and a web page, the goal is to predict whether the user will whitelist the entire site or the page that s/he intends to view when facing the counter-ad-blocking wall. The input includes user features, page features, and user engagement behavior in previous visits (if any).*

A. Data Collection

Building prediction models for whitelist behavior requires data from web publishers; specifically, it needs user browsing logs and ad blocker usage logs for all user visits. In this paper, we use 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 shows, if the website detects an ad-blocker, the website will pop up a message "Adblock Detected" to ask the user to whitelist Forbes (or the specific page the user intends to visit) in order to view the content. Once Forbes site or the page is whitelisted, users are given access to content. If a user refuses to whitelist, she will be prevented from viewing the intended content.

Each record in our dataset is a user session. A session records the pages the user intended to view, the pages actually viewed by the user, and user behaviors on pages. A user may visit several pages in a session. The sessions in which the user refused to whitelist, and subsequently left the site, contain only the pages the user intended to view.

Note that an ad blocking user can be blocked multiple times for several reasons: (i) the user did not whitelist the last time s/he attempted to visit the site; (ii) the user whitelisted a single session and then turned on the ad blocker again; (iii) the user whitelisted only one page, and then wants to access another page; (iv) the user re-installed or upgraded the ad blocker software and the whitelist data was lost.

B. Feature Extraction

The model input comprises of the following components:

- 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. OS includes both desktop OS (e.g., Windows, MacOS, Linux) and mobile OS (iOS, Android).
- The user's *geo-location*, which is detected from user IP addresses and provided at country granularity. We consider user geo-locations because this is the only explicit feature about users that can be easily obtained by publishers without violation of user privacy [13].
- The user's *traffic source*, which is the origin of a user's visit. There are three main traffic source categories: search engines traffic, direct traffic, or referring 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. Referring traffic counts those visitors that click a link on another site (e.g., social networking sites).
- *Page freshness* is the duration between the time the page was published on the website and the time the page was read. Page freshness is an important attribute of web resources and can benefit a series of time-sensitive applications about user behavior [14]. In our case, freshness is measured in days.
- *Page popularity* is the total number of visits received from all users for the page during our data collection period. It stands for the "hotness" of a page.
- *Page channel* and *section* of the article in a page are its topical categories defined by the publisher's website, e.g., finance and lifestyle. A channel can be considered as a high-level topic label of a page. A section is defined as a sub-channel at finer granularity.
- *Session number* is a counter of the user sessions on the web site. A higher session number indicates a higher user interest/loyalty for the web site.
- *Number of page views and number of actions* in the last session reflect the degree of user engagement and thus their interests with the website the last time she visited the site. The actions include clicks, scrolls, mouse selection, etc.
- Whether a user *whitelisted* in the last session records the most recent whitelisting behavior of the user. Recall that a user may have whitelisted the site multiple times in the past and then had the ad blocker turned on again. This features records the most recent behavior.
- Other page attributes in the Forbes article metadata are also taken into account: page type (e.g., "blog", "blogslide", or "trending activity"), whether a page is in a standard template type, whether the article is written by Forbes staff, and the number of user comments.

We do not have gender and age information for individual users. To protect user privacy, Forbes maintains aggregated group-level data for these features. Therefore, we cannot use them in our model.

C. Representing Categorical Variables

The geo-location is a categorical variable with a very large cardinality. It is difficult to be represented in a binary vector (i.e., one-hot encoding) because this binary representation dramatically increases the input feature space, and thus increases both the execution time and the memory consumption of the model. Also, it makes sense to consider geo-location as a single feature when analyzing the results, instead of using many dummy variables. Inspired by [15], we use the class probability $P(y=1|c_i)$, calculated from the training data, to represent c_i, where y is the targeted whitelist behavior and c_i is a country. This approach is more effective when the number of samples per category is relatively large. Another perspective to understand the above transformation is to construct a naive Bayes classifier, in which we calculate the naive Bayes output in terms of the per-input class probability $P(y=1|c_i)$ as the representation of the class.

We use the same method to process channel and section features. For the traffic source and OS, we use the one-hot encoding because they have significantly fewer categorical values.

D. Normalizing Raw Data

The data range of freshness is large, from several days to thousands of days. As the freshness value increases, the marginal impact of freshness decreases. Thus, we consider a log transformation to normalize the freshness feature. Here we add 10 to $freshness$ to avoid having negative numbers.

$$freshness_{normalized} = log_{10}(10 + freshness) \quad (1)$$

Similarly, we normalize the popularity feature as follows:

$$popularity_{normalized} = log_{10}(10 + popularity) \quad (2)$$

E. Prediction Models

We use four models for whitelist prediction: Logistic Regression, Random Forest, Multilayer Perceptron (MLP), and gradient boosting regression tree (GBRT). For the sake of brevity, we briefly describe the first three, which are widely used in industry, and then focus on the GBRT model, which is more sophisticated and performs the best in our experiments.

Logistic Regression is a classic model for binary classification, and we developed it because whitelist prediction can be considered a classification problem. 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. MLP is a neural network model with two layers (100 neurons, 8 neurons) with Relu activation, and we use the binary cross-entropy as the loss function.

We select GBRT as the fourth model because of its ability to combine categorical and numerical variables by splitting nodes based on discrete and continuous feature values. It also has low memory consumption and fast execution time. This is important because our prediction must work well in real-time: publishers predict the whitelist behavior of an incoming user when the request arrives and have to make a decision immediately regarding its counter-ad-blocking strategy to avoid delays that may affect user experience.

GBRT converts weak learners together into a strong one. In GBRT, the instances that were misclassified by the previous learners are given higher weights in the current learner; therefore, subsequent learners give more focus to them during training. Figure 4 shows an example of GBRT. In each decision tree f_i, each training instance $\boldsymbol{x_i}$ is classified into one leaf. The example has three instances (i.e., users). Each leaf is

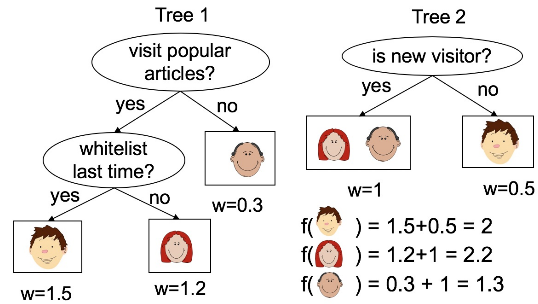

Fig. 4. An example of GBRT model. The final predicted probability to whitelist is the sum of all decision trees

associated with a weight w that stands for the prediction score. GBRT builds decision trees sequentially and updates the corresponding leaf weights of previous trees. Each decision tree is a weak learner and has limited depth (e.g., 3-8) to minimize its complexity. After building the trees, GBRT sums up the predictions of all the trees as the final prediction [16].

$$\hat{y_i} = \sum_{t=1}^{T} f_t(\boldsymbol{x_i}) \quad (3)$$

Here T denotes the maximum number of trees, and $f_t(\boldsymbol{x_i})$ is the prediction score of t^{th} tree for the data point $\boldsymbol{x_i}$.

Unlike random forest using bagging ensemble methods, which build trees in parallel, GBRT builds new trees sequentially and optimizes the residual of the last iteration. Hence the predicted score at iteration t, $\hat{y_i}^{(t)}$ is given as:

$$\hat{y_i}^{(t)} = \sum_{t=1}^{T} f_t(\boldsymbol{x_i}) = \hat{y_i}^{(t-1)} + f_t(\boldsymbol{x_i}) \quad (4)$$

The objective function for training GBRT at iteration t is:

$$J^{(t)} = \sum_{i=1}^{n} L(y_i, \hat{y_i}) + \sum_{t=1}^{T} \Omega(f_t) \quad (5)$$

Here n is the number of training samples, $L(\cdot)$ denotes a loss function, and $\omega(\cdot)$ refers to the regularization term to punish complexity and overfitting. We use binary cross-entropy as the loss function $L(\cdot)$. Given formula 4, the objective function can be written as:

$$J^{(t)} = \sum_{i=1}^{n} L(y_i, \hat{y_i}^{(t-1)} + f_t(\boldsymbol{x_i}) + \sum_{t=1}^{T} \Omega(f_t) \quad (6)$$

Currently there are three implementations of GBRT: XGBoost [17], LightGBM [18], and CatBoost [19]. We select XGBoost [17] for our implementation because it is robust and widely used.

IV. Experiment Evaluation

The goals of our experimental evaluation are: (1) evaluate the prediction accuracy results of the four predictive models on a large scale, real-world dataset; (2) evaluate if the best model in terms of accuracy can provide prediction in real-time.

A. Settings

The data collection is described in Section III-A. Since we need user past engagement information to make predictions, the datasets contains all the sessions of returning users (a total of 33,000 ad-blocking-detected sessions). We also observe that the whitelist ratio in the ground truth data is 40%. We utilize the 10-fold cross validation and randomly split the data into training-validation-test datasets with a ratio of 8:1:1. We use randomized search instead of grid search for fast tuning of the following major hyper parameters: 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, subsample ratio of columns when constructing each tree.

B. Metrics

We use the following metrics to assess our 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.

F1 Score: It is the harmonic average of precision and recall. F1 score reaches its best value at 1 (perfect precision and recall) and worst at 0.

The F1 score calculation is influenced by the decision threshold, which turns a predicted probability into a binary value. Thus, we compute the F1 score of each model by varying the decision threshold to obtain the highest score. AUC and LogLoss are not influenced by the threshold. They are better metrics if the class distribution is highly imbalanced.

C. Results

Prediction Performance. Table I shows the performance of the four models using the best threshold for each of them. XGBoost performs the best in all three metrics, and overall it exhibits good whitelist prediction performance. Figure 5 plots the ROC of the models, and it shows that XGBoost is the best at every point.

TABLE I
EVALUATION OF WHITELIST PREDICTION MODELS

Model	Model Performance		
	Log Loss	AUC	F1_core
Logistic Regression	0.3560	0.8458	0.6745
Random Forest	0.3653	0.8678	0.6875
Multilayer Perceptron	0.3342	0.8813	0.6996
XGBoost	**0.3201**	**0.8919**	**0.7153**

Figure 6 depicts the density distribution of the prediction probabilities for three models. Unsurprisingly, we find there is a clear gap between two blocks. This means there is strong confidence to whitelist for some (user, page) pairs. Our analysis shows that this happens for loyal users who whitelisted the site or some pages before and have high user engagement.

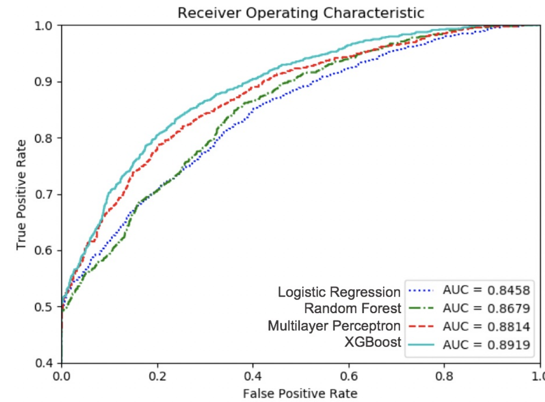

Fig. 5. ROC curve and AUC score for our models

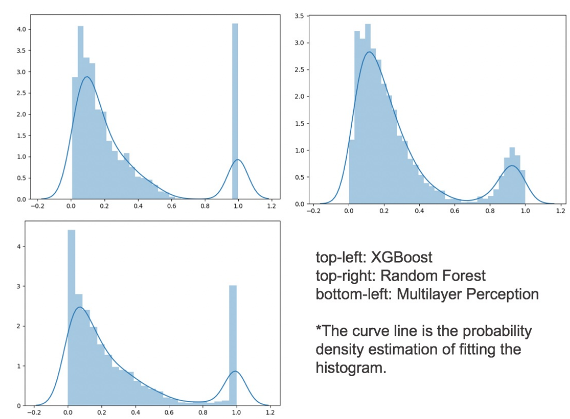

Fig. 6. The density distribution of the prediction probabilities for 3 models. The X axis is the prediction score, and the Y axis is the probability density

Compared with MLP and Random Forest, XGBoost predicts more values closer to 1 or 0. In other words, XGBoost is more confident in giving clear outputs. Blunt confident and wrong prediction will be penalized by the LogLoss model. However, since XGBoost has also the best LogLoss values among the models, we conclude that its confidence is correct and reliable.

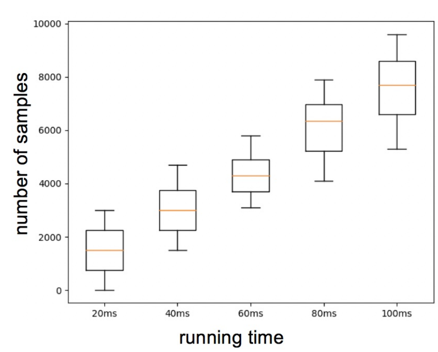

Fig. 7. Execution time of XGBoost

Real-time Performance. The prediction is made after the page was requested by the user and before the user engages with the page. The publishers want to make sure the latency effect of the prediction is not noticeable to users. Thus, small execution time of the model is critical for the model to be feasible in applications. We measure the execution time of the prediction made by XGBoost using a laptop with 4 2.3GHz CPU cores and 8GB memory. We execute 5 runs for each experiment with a batch size of 32. As shown in Figure 7, XGBoost can make about 8,000 predictions in 100ms. This is good enough for most publishers. If necessary, the performance can be improved by running the model on a powerful server.

V. Conclusions

We studied the problem of predicting whether an ad blocking user will whitelist a web page or a website that she intends to access when faced with the counter-ad-blocking wall. We implemented four prediction models and found that the gradient boosting regression tree model provides the best prediction performance. Further, this model works well in real-time. Whitelist prediction can be used to design personalized counter-ad-blocking strategies in order to increase the number of users who visit a publisher's web site and increase the revenue.

For future work, we plan to evaluate how our prediction models will be used in practice. The prediction may have false positive or false negative errors. The worst case scenario happens when there is false positive error: the model predicts that the user will whitelist, but the user does not. In this situation, the publisher will use the counter-ad-blocking wall and the user will choose to leave the website. As a result the publisher not only fails to earn revenue but also loses the user. On the other hand, if a user is predicted not to whitelist by mistake, the publisher may choose not to use counter-ad-blocking wall. Although in this case the publisher does not earn much revenue, at least the user is maintained. We plan to study how to determine the best decision threshold to optimize the results. Finally, we plan to perform feature analysis to find out which features influence the most the prediction. The results of this analysis can give managerial insights to publishers on designing personalized counter-ad-blocking strategies.

Acknowledgement

This work is partially supported by NSF under grant No. DGE 1565478, by the Leir Foundation, and by the National Institutes of Health under grant No. UL1TR003017. 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, "Iab internet advertising revenue report," https://www.iab.com/insights/iab-internet-advertising-revenue-report-conducted-by-pricewaterhousecoopers-pwc-2/, 2018.

[4] wikipedia, "Advertising," https://en.wikipedia.org/wiki/Advertising, accessed: 2017-11-7.

[5] 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.

[6] 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.

[7] 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.

[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] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] L. Yan, R. H. Wolniewicz, and R. Dodier, "Predicting customer behavior in telecommunications," *IEEE Intelligent Systems*, vol. 19, no. 2, pp. 50–58, 2004.

[16] J. H. Friedman, "Greedy function approximation: a gradient boosting machine," *Annals of statistics*, pp. 1189–1232, 2001.

[17] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. ACM, 2016, pp. 785–794.

[18] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "Lightgbm: A highly efficient gradient boosting decision tree," in *Advances in Neural Information Processing Systems*, 2017, pp. 3146–3154.

[19] A. V. Dorogush, V. Ershov, and A. Gulin, "Catboost: gradient boosting with categorical features support," *arXiv preprint arXiv:1810.11363*, 2018.