Computer and Information Science; Vol. 13, No. 2; 2020
ISSN 1913-8989 E-ISSN 1913-8997
Published by Canadian Center of Science and Education

Using Deep Learning to Block Web Tracking

Jianyi Wang[1]

[1] Bloomfield Hills, United States

Correspondence: Jianyi Wang, Bloomfield Hills, Michigan, United States.

Received: March 2, 2020 Accepted: April 16, 2020 Online Published: April 22, 2020

doi:10.5539/cis.v13n2p27 URL: https://doi.org/10.5539/cis.v13n2p27

Abstract

Most websites that users browse every day utilize trackers that can identify who users are and what activities they conduct online. Although there are benefits to these trackers, they also raise considerable privacy concerns. This research study examines the issue of how to identify web trackers and how to protect users from being tracked. Specifically, this study investigates how AI Deep Learning technologies can be leveraged to identify and stop trackers. The main idea of AI Tracker Blocking is that AI can detect the small differences between tracker server domains and non-tracker server domains. For instance, a tracker domain may appear like this: analytics.xxx.bid, whereas a non-tracker domain may be like this: mail.xxx.com. Although it is likely impossible to block all trackers, results from this study indicate that there are new ways to identify them using Deep Learning.

Keywords: artificial intelligence, internet privacy, web tracker, web tracker identification

1. Introduction

1.1 The Side Effects of Web Tracking

Broadly defined, web tracking (Michael Alan Pogue, Laura Allison Werner, Ralf I. Pfeiffer, Pratima Gupta, Yong Zhang, & James Andrew Clark) is the practice of identifying and collecting user information online. There are many types of web tracking. For example, websites can track your IP address and correlate the IP address to your browsing history, or they could set up a digital cookie on your computer that saves your browsing data. Typically, third-party web service providers label every visitor who accesses their webpage and save their browsing histories. They can record what users have searched, the webpages users have visited, and even the words users have typed by running a JavaScript file in the background that saves all of this information into the database on the webserver. They then set up a cookie on the user's browser that is linked to the user's data stored in the database. For example, websites that use cookies usually have a notification that states "this website uses cookies to improve your experience," which implies web tracking. This kind of tracking could only be done by third party web service providers since only the end-user and the website that is being accessed can run a javascript file.

Oftentimes web tracking is beneficial since the third-party web service providers can tell visitors the pages they have browsed upon request. For an online shopping website, web tracking is necessary because most users will want to see what items they have recently viewed; for a news website, users often want to see the news that they previously viewed. By collecting user data, web service providers can also learn more about individuals who visit their sites to provide them with better-tailored online services. However, numerous privacy concerns arise from the practice of web tracking. For example, third-party web service providers sometimes sell individuals' data to other companies (e.g., advertisers), resulting in unwanted advertising phone calls.

1.2 Limitations of Current Solutions

Currently, there are many solutions to solve privacy concerns raised by web tracking, but these strategies have considerable shortcomings. Using proxy servers such as PPVPN (L. Andersson, & T. Madsen, 2005) is one often-used approach. Proxy servers are like tunnels that connect one computer or server to another. When the user is using a proxy server, all of his request traffic goes through the tunnel toward the webserver and the webserver sends data back through the tunnel to the computer. One benefit of using a proxy is that it can hide the user's real IP address. Some people think that using a proxy could avoid most of the aforementioned privacy concerns, and they might think that it would be a good idea to use a proxy all the time, but the problem is that once users stop using the proxy, their real IP will be exposed. And if they don't stop using the proxy, there is no

difference between whether using a proxy or not.

Browser plugins are another method used to block web trackers. They attempt to stop suspicious requests before any data is exchanged by comparing the requested domain to a domain list that records many of the trackers. Although browser plugins can block most of the trackers on the list, their major shortcoming is that they cannot block trackers that are not on the list. Because most of the tracker websites are fairly sophisticated, they change their domain names over time, which considerably limits the effectiveness of browser plugins. As such, using browser plugins only is definitely not enough to block all trackers. Also, since the browser plugins are written in JavaScript and websites can run JavaScript, websites are able to detect whether their users are using anti-tracker plugins or not and stop providing services for users who are using these plugins. To address this issue, tracker blocking has developed to function through the hosts file, which is similar to browser plugins but undetectable by the website. However, this method still does not meet the aforementioned "practical" requirement of being able to distinguish the unmet domains. We urgently need a new approach to stop these web trackers.

1.3 Related Works

The graph-based ML approach that identifies advertising and tracking resources (U. Iqbal, P. Snyder, S. Zhu, B. Livshits, Z. Qian, & Z. Shafiq, 2020) is too complicated because it needs browser kernel level support that usually needs to modify the kernel, and it doesn't mention anything about comparing load times on websites that don't have any trackers. It may be accurate but definitely not that useful.

1.4 The Approach of Artificial Inteligence

The main point of this work is to use AI Deep learning (LeCun, Y., Bengio, Y. & Hinton, 2015) technologies to identify a more secure, efficient and practical approach to stop web trackers. By "secure" I mean that the method should not be discovered by the websites while able to stop the connection between trackers. By "efficient," I mean that the time spent on one Http request using the method divided by the time spend on one Http request not using the method should be as small as possible. Accuracy is the percentage of domains that are correctly classified over the number of domains. Unmet domains are domains that are neither on the tracker list nor the non-tracker list. By "practical", the new method should have an accuracy higher than 50% with unmet domains (the accuracy of current solutions has 50% accuracy since they cannot tell if a domain is a tracker domain or not if that domain is not on the list that they use). My hypothesis is that the new approach with AI Deep Learning should be able to block most of the trackers better than current blocking methods and also block unmet tracker domains while allowing connection with unmet non-tracker domains.

2. Method

2.1 The Reason for Choosing Deep Learning

Deep Learning is a new method of creating powerful AI models that have the ability to learn and capture certain subtle features of the data. Multi-layer perceptron (MLP) is an artificial neural network with a trend structure that maps a group of input vectors to a group of output vectors. An MLP can be viewed as a directed graph consisting of multiple node layers, each fully connected to the next. In addition to the input node, each node is a neuron (or processing unit) with a nonlinear activation function. A supervised learning method called a backpropagation algorithm is often used to train MLP. MLP is a generalization of perception, which overcomes the disadvantage that perceptron cannot recognize linearly inseparable data. For AI tracker blocking, MLP is the best fit because tracker blocking is basially decision making and finding the best fit "line" in a higher dimension, this best fit line would be able to separate most of the the tracker domains out of the non-tracker domains.

2.2 Using Word to Vector (word2vec)

To block web trackers, the AI tracker blocker should be handling all the requests from the computer. And to implement that, I wrote a system proxy using python and it is able to handle all the requests and get the domain of the website that the computer is trying to connect to. The AI tracker blocker should then turn the domain names on the tracker list and the non-tracker list into a big matrix that could be used to train the model. In the step of data processing, domain names will be split by the symbol ".". The domain www.xxx.com will be split into a list that looks like ['www', 'xxx', 'com'] and the last part ['com'] will be removed since the suffix does not matter much. Although it is obvious for a human to tell which domain belongs to which, AI needs one more process: word to vector (Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, & Jeffrey Dean, 2013) that turns a domain name to a format that a computer can recognize. And then the AI model should start training, and training is basically adjusting the weights of the neural network and find the best fit line on a higher dimention.

2.3 Training Procedures

2.3.1 Experimental Setup/Approach

First, three different datasets were prepared: the training set, the testing set and the validation set. Each dataset was composed of one tracker list and one non-tracker list (otherwise the weight of the tracker domains and non-tracker domains will be different). The training set was prepared to train the model. The testing set was used to calculate the accuracy of the model while training (finding the best fit line). And the validation set was used to finally measure how well the model blocks web tracking.

2.3.2 Model Training

To collect data for training dataset, I created a data loader that loads one domain at a time from each list and turns them into a (179090,) matrix through word2vec then sends the matrix to the MLP network. I used Peter Lowe's Ad and tracking server list (Peter Lowe, 2019), which is a list of websites to block that most web browser plugins use. As for the non-tracker list for the tracker list of the training set and Alexa ranking top 100,000 for the non-tracker list of the training set. Alexa ranking is a system that counts the unique visitors of all website and page views. The higher a website's Alexa ranking is, the more popular the website is. The testing set was generated by a program that randomly selected 2000 domains from each list of the training set. Lastly, the validation set contains 200 domains (100 for each list) that were generated by me and some of the domains in the training set.

2.3.3 Model Evaluation

I created a proxy server using python and made it a system proxy so that all Http/Https traffic goes through the AI tracker blocking model. The proxy disconnects any connection to domains that have scored over 0.51. Every domain that has a score over 0.51 was considered as "true", and every domain that has a score below or equal to 0.51 was considered as "false". Usually for binary classification problems using sigmoid as the activation function, the boundary value should be 0.5. But because of overfitting, some unmet domains are classified as tracker domains since they have the score of 0.5062. To solve the problem, I manually add a bias and moved the cut-off value to 0.51. I also wrote a program that generates fake urls based a a set of rules to calculate the recall rate, precision and f-measure.

3. Results

3.1 Metrics

There are seven metrics presented for each set of results: recall rate, precision, f-measure, true-positive rate, false-positive rate, accuracy, efficiency. Recall rate is the percentage of the predictions that are classified as true were true. Precision is the percentage of the predictions that were true are classified as true. And f-measure is the mean of the recall rate and precision. These three metrics should be as close to 100% as possible. The full results are presented in Table 1.

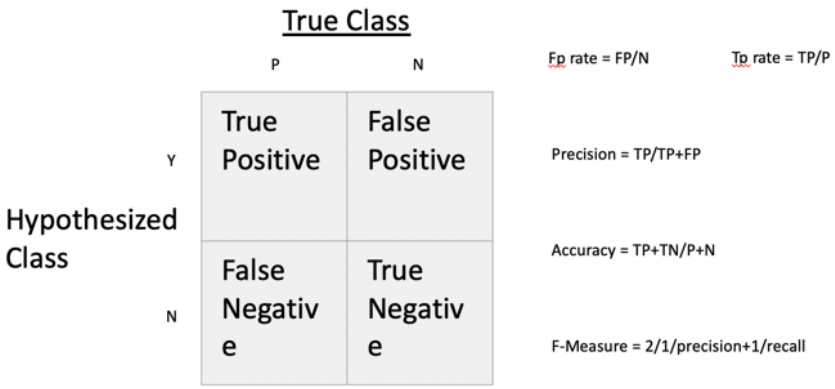

Figure1. The Definition of the Results

3.2 Accuracy

The first set of results is based on the testing set. 2000 existing domains have been tested. And the second set of

results is based on unmet domains generated by me. 5040 non-existing domains have been tested.

Table 1. Detailed Result

Result\Dataset	Testing	Validation
Recall Rate	98.20%	73.45%
Precision	99.09%	100.00%
F-Measure	98.64%	84.70%
True-Positive	98.20%	73.45%
False-Positive	0.89%	0.00%

Results gathered using a program that generates random domains and accessible domain names under a rule.

3.4 Efficiency

The average time accessing google without the proxy server is about 181.96 millisecond. And the average time accessing google using the proxy is 208.65 millisecond. Which means 5.49 requests/second without the proxy and 4.97 requests/second with the proxy. The difference is only 26.69 millisecond.

4. Discussion

4.1 Accuracy Analysis

The result indicates the AI Tracker blocking approach is promising; the model has an amazing accuracy on the traning set. For the validation set, it has a precision of 100%, which indicates that it will not block any non-tracker normal websites that are not on the list. It also has a recall rate of 73.45%, which means it is able to correctly identify 73.45% of the unmet tracker domains. For some of the domains that normally would not be on the list, the model is still able to differentiate them while the browser plugins based on tracker lists cannots: it is able to distinguish 84.60% of the domains that were made up by me. Although it is difficult to determine the accuracy of a browser plugin based on tracker lists since they can only block the domains on the list, the AI tracker blocking method is able to do what a browser plugin can do and is able to differentiate more unmet domains.

4.2 Efficiency Analysis

The AI approach is also very efficient; it has an efficiency of 98%, which means for a URL that usually takes 500ms to access, using this AI tracker blocking model will cause a 10ms time increase.

4.3 Conclusion

The evidence suggests that my AI Tracker blocking approach is a secure, efficient and practical approach for to addressing the problem of web tracking. Furthermore, AI tracker blocking is able to not only work with domain names but also the URI (such as /xxx/xxx/xxx?xxx=xxx), which should improve the accuracy as well as range the model. For these reasons, believe AI tracker blocking will become one of the most efficient and efficacious way to stop web trackers, and it deserves greater attention from researchers and technology companies.

References

Andersson, L., & Madsen, T. (2005). *Provider Provisioned Virtual Private Network (VPN) Terminology.* https://doi.org/10.17487/rfc4026

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature, 521,* 436-444. https://doi.org/10.1038/nature14539

Michael, A. P., Laura, A. W., Ralf, I. P., Pratima, G., Yong, Z., & James, A. C. (1997). *Web site client information tracker.* Retrieved from https://patents.google.com/patent/US6112240A/en

Peter Lowe. Peter Lowe's Ad and Tracking List. Retrieved August 10, 2019, from https://pgl.yoyo.org/adservers/serverlist.php

Tomas, M., Ilya, S., Kai, C., Greg, C., & Jeffrey, D. (2013). Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems* (pp. 3111-3119). Lake Tahoe, Nevada.

U. Iqbal, P., Snyder, S., Zhu, B., Livshits, Z. Q., & Shafiq, Z. (2000). Adgraph: A graph-based approach to ad and tracker blocking. In *Proc. of IEEE Symposium on Security and Privacy*.

Notes

Note 1. All codes are available on Github