On Analyzing Third-party Tracking via Machine Learning

Alfonso Guarino, Delfina Malandrino, Rocco Zaccagnino, Federico Cozza and Antonio Rapuano
Department of Computer Science, University of Salerno, Via Giovanni Paolo II, 84084 Fisciano (SA), Italy

Keywords: Privacy, Third Party Tracking, Machine Learning.

Abstract: Nowadays, websites rely on services provided by third party sites to track users and offer personalized experiences. However, this practice threatens the privacy of individuals through the use of valuable information to create a digital personal profile. The existing client-side countermeasures to protect privacy, exhibit performance issues, mainly due to the use of blacklisting mechanisms (list of resources to be filtered out).
In this paper, we study the use of machine learning methods to classify the thirdy-party privacy intrusive resources (trackers). To this end, we first downloaded (browsing Alexa's Top 10 websites for each category like sports, shopping etc.) a dataset of 1000 web resources split into *functional* and *tracking*, and then we identified suitable metrics to distinguish between the two classes. In order to evaluate the effectiveness of the proposed metrics we have compared the performances of several machine learning models based on supervised learning among the most used in literature. As a result, we obtained that the Random Forest can classify functional and tracking resources with an accuracy of 91%.

1 INTRODUCTION

One of the main goal of websites is to provide online personalized experiences. To this end, a common practice is to rely on services provided by third party companies to monitor daily activities of Internet users. Any interaction enabled and mediated by ICT, is recorded and stored by third party companies (3rd-party, from now on) to create a fingerprint of our digital identity: social network activities, Google searches, credit card purchases, Netflix preferences, health-related information, or searches on online shops. Specifically, 3rd-party companies (e.g., aggregators, analytics, marketing companies, commercial data brokers etc.) monitor Web users in several ways: advertisements, analytics functionalities, social widgets, Web bug, standard HTTP headers, etc (Malandrino and Scarano, 2013). Such monitoring is enabled by stealth 3rd-party trackers embedded in the web pages, for purposes such as targeted advertising (Interactive Advertising Bureau (IAB), 2019), predicting trends, generating enormous benefits at the expense of users' privacy (Binns et al., 2018).

Nowadays, privacy violation caused by 3rd-party tracking has become a pervasive problem (Krishnamurthy and Wills, 2010), and a huge effort has been made to protect users' privacy against online tracking. Among these, anti-tracking technology based on *blacklists* is most effective (Mayer and Mitchell, 2012). Many commercial privacy preserving tools (Adblock[1], DoNotTrackMe[2], Ghostery[3]) are based on blacklists. They generate blacklists offline and block requests to the URLs if it is included in the blacklist. The main disadvantage of such a technique is the maintainance and performance (D'Ambrosio et al., 2017). In fact, this method highly depends on the records in the blacklist. which needs to be updated regularly because a tracking company can adopt new domains to track users (Pan et al., 2015).

In this paper, we propose the use of machine learning methods to classify trackers on the Web. Machine learning is based on methods that can learn from data without relying on rules-based programming[4]. As explained in (Domingos, 2012), *"it can figure out how to perform important tasks by generalizing from examples in a dataset"*. In this work we focus on the classification of every type of 3rd-party tracking resource. To this end, we first dowloaded a set of Web resources by browsing Alexa's Top 10 websites[5] for each category (shopping, sports, etc.), second we enginnered a set of metrics suitable to distinguish be-

[1]https://chrome.google.com/webstore/detail/adblock
[2]https://addons.mozilla.org/it/firefox/addon/donottrackplus/
[3]https://www.ghostery.com
[4]https://www.mckinsey.com
[5]https://www.alexa.com/topsites

Guarino, A., Malandrino, D., Zaccagnino, R., Cozza, F. and Rapuano, A.
On Analyzing Third-party Tracking via Machine Learning.
DOI: 10.5220/0008972005320539
In *Proceedings of the 6th International Conference on Information Systems Security and Privacy (ICISSP 2020)*, pages 532-539
ISBN: 978-989-758-399-5; ISSN: 2184-4356

tween functional and tracking resources; those metrics are based on HTTP traffic. In order to evaluate the effectiveness of the proposed metrics, as proposed in (Rzecki et al., 2017a; Cosimato et al., 2019; Deeba et al., 2016), we have compared the performance of several machine learning models based on supervised learning among the most used in literature, i.e., Naive Bayes (Maron, 1961), Random Forest (Breiman, 2001), MultiLayer Perceptron (Rumelhart et al., 1988), C4.5 (Quinlan, 2014), and Support Vector Machine (Wang, 2005). As best result, the Random Forest can classify functional and tracking resources with an accuracy of 91%.

The paper is structured as follows: Section 2 provides the background of the paper, Section 3 is about related work in the field of Web privacy, Section 4 details our approach to classify 3rd-party trackers, Section 5 includes final remarks and future steps of this research.

2 BACKGROUND

Machine Learning. In the last decade, machine learning (Bishop, 2006) joined a number of emerging studies (Rzecki et al., 2017a; Rahwan et al., 2019; Butler et al., 2018; Camacho et al., 2018) in different science areas, ranging from chemistry (Wu et al., 2018) to law science (Hanke and Thiesse, 2017), to music (Cosimato et al., 2019).

One of the main techniques to use machine learning requires to train the model through a supervised learning. A supervised learning algorithm analyzes the training data and produces an inferred function, which can be used for mapping new samples. The classification (our problem in the following) is the task of approximating a mapping function (f) from input variables (X) to discrete output variables (y). The output variables are often called labels or classes. In our analysis, the mapping function that has to predict the label or class for a given observation, i.e., whether a Web resource can be classified as "tracking" or "functional".

Several machine learning models have been used for classification problems. We briefly describe the ones that we use in this paper. For further details about machine learning we refer the reader to (Hall et al., 2009). The models we use are implemented in Weka library (Hall et al., 2009): NaiveBayes classifier, Random Forest, Support Vector Machines, MultiLayer Perceptron, and J48 classifier.

- *NaiveBayes (NB)*: a simple probabilistic classifier based on applying Bayes' theorem with strong (naive) independence assumptions between the features (Maron, 1961).
- *Random Forest (RF):* is a supervised classification algorithm which consists of an ensemble of methods based on bagging (Breiman, 2001).
- *Support Vector Machines (SVM):* is a supervised learning model with associated learning algorithms (Wang, 2005).
- *MultiLayer Perceptron (MLP):* is a feedforward artificial neural network that exploits a supervised learning technique called backpropagation (Rumelhart et al., 1988) for training.
- *J48*: it is the Weka implementation (Quinlan, 2014) of the more famous C4.5 classifier (Quinlan, 2014).

There exist several metrics for evaluating scores of machine learning models. In this work we used the accuracy (Swets, 1988; Hossin and Sulaiman, 2015) like in (Cosimato et al., 2019; Rzecki et al., 2017b; Kovács et al., 2016; Shaw et al., 2020). Informally, accuracy is the fraction of classifications our model got right; formally it is defined as

$$accuracy = \frac{(tp+tn)}{(tp+tn+fp+fn)}$$

where tp, fn, fp, and tn are the number of true positives, false negatives, false positives and true negatives, respectively.

Web Tracking and HTTP Header. A typical webpage consists of several Web-components, e.g., JavaScript programs, Flash-content, images, CSS, etc. When a user opens a website in a Web browser, the fetched webpage always generates several other HTTP(S) connections to download additional components of the webpage. These components can be downloaded from the website visited by the user or downloaded from other 3rd-party domains. Here, we focus on one type of Web-component which is loaded from 3rd-party domains through the previously mentioned HTTP requests.

HTTP header fields are components of the header section of request and response messages in the Hypertext Transfer Protocol (HTTP). They define the operating parameters of an HTTP transaction. A few fields can contain also comments (i.e., in User-Agent, Server, Via fields), which can be ignored by software[6]. Many field values may contain a key-value pair separated by equals sign. HTTP headers contain different fields that can be used to track users'

[6]https://tools.ietf.org/html/rfc7230#section-3.2.6

identity. For instance: (i) *Referer*: indicates what resource requested an element of the Web page (Malandrino and Scarano, 2013); (ii) *From*: used when the requested resource has to confirm the identity of an account, it can contain user's email; (iii) *User-Agent*: it contains information about the user's machine, browser, and language; (iv) *Cookie*: indicates the value of cookies previously set in order to store user's information such as browsing habits.

3 RELATED WORK

Existing Privacy Mechanisms. Although Web tracking has garnered much attention, no effective defense system has been proposed. In (Roesner et al., 2012) authors proposed a tool called *ShareMeNot*, but it can only protect users against social media button tracking, a small subset of threats. Disabling script execution[7] provides protection at the cost of pages failing to open or render properly (Krishnamurthy et al., 2011). The Do Not Track (DNT) header requires tracker compliance and cannot effectively protect users from tracking in reality (Gervais et al., 2017). The most effective method to defend against 3rd-party tracking is based on blacklists, and most commercial anti-tracking tools Ghostery, and AdBlock are based on blacklists.

Existing Machine Learning-based Privacy Mechanisms. Recently, a number of works focused mainly on advertisement content (Kushmerick, 1999; Orr et al., 2012; Bhagavatula et al., 2014) or more in depth on tailoring the advertisement on the Web (Haddadi et al., 2010; Toubiana et al., 2010; Guha et al., 2011; Liu and Simpson, 2016; Parra-Arnau et al., 2017; Parra-Arnau, 2017). Specifically, in (Kushmerick, 1999) authors, for the first time, employed machine learning techniques to block online advertisements, using the C4.5 classification scheme. In (Orr et al., 2012) authors trained a classifier for detecting advertisements being loaded via JavaScript code. In this study, they manually labeled advertisement-related JavaScript code. Authors in (Bhagavatula et al., 2014) presented a technique for detecting advertisement resources exploiting the k-nearest neighbors classification (Peterson, 2009) leveraging on the EasyList (blacklist used by AdBlock). More recently, authors in (Parra-Arnau et al., 2017) proposed a Web-browser extension, which enables users to configure their own access policies to enforce a smart blocking over advertising platforms. Authors evolved the Web-browser extension in (Parra-Arnau, 2017) by adding a functionality that blocks or allows advertisements depending on the economic compensation provided by the adverting platforms to the users.

The main difference with previous studies in literature is that we focus on classifying tracking behaviours from functional behaviours independently from the resource (analytic, tracking, advertisement).

Other works have focused, like us, on classifying via machine learning not only advertising content but JavaScript programs and other kind of resources. For instance, from one hand, in (Ikram et al., 2017) authors propose a One-Class SVM classifier to distinguish between functional and tracking JavaScript programs. The classifier is feed with semantic representation of the JavaScript programs, i.e., the semantic information associated with the abstract sintax tree of the JavaScript code, and is capable to provide up to 97% accuracy in the classification task. The main difference with this work is that we aim to develop our solution in a client-side browser extension, thus we need to reduce at minimum the computation time, while in (Ikram et al., 2017) the main aim was to develop the best possibile classifier, without accounting the performance in the real usage. Our features are easier to compute than the ones computed in (Ikram et al., 2017). From the other hand, in (Iqbal et al., 2020), authors proposed a novel Chromium patch to face every kind of trackers and advertisement content in the Web. The tool shows good performances also in real usage with 2.3% crashes and 5.9% major issues. The main difference with this work is that we aim to provide a browser extension for the most used Web browsers, i.e., Google Chrome, Mozilla Firefox instead of creating a novel one by modifying the source code of Chromium. So, in a future perspective, we expect a larger audience of customers.

4 OUR APPROACH FOR CLASSIFYING WEB RESOURCES

In this section, we describe a novel machine learning approach to automatically detect 3rd-party tracking activities while browsing the Web. We remark that such a approach can be used to overcome the problems of having to manually update blacklist. Our classification problem is a binary classification problem on Web resources where positive samples are *tracking* requests and negative samples are *functional* ones. In the following, we describe in details the three main

[7]https://addons.mozilla.org/zh-CN/firefox/addon/noscript/

steps of our approach (see Figure 1):

- *Data collection*: by means of automatic browsing sessions we navigate a number of Web pages from Alexa's Top 10, meanwhile we trace the HTTP traffic (Section 4.1).
- *Features extraction and labeling*: *(1)* a set of suitable features has been defined and engineered from the data collected, in order to build the dataset used for the training and testing of the machine learning models, *(2)* the dataset has been manually labeled (Section 4.2).
- *Validation and testing*: the dataset has been split into a train set and a test set; a *k*-fold cross-validation has been performed on the train set to validate several well-known machine learning models previously described; once validated, machine learnign models have been tested on the test set (Section 4.3).

4.1 Data Collection

By using Selenium webdriver (Avasarala, 2014) we automatically browse Alexa's Top 10 Web pages for each of the Alexa's categories[8], i.e., *adult, arts, business, computers, games, reference, regional, science, shopping, society, health, home, kids and teens, news, recreation, sports and world*. We then employ Fiddler (Lawrence, 2012) and Wireshark (Orebaugh et al., 2006) to monitor and capture HTTP traffic while browsing such websites. Moreover, each captured resource has been downloaded. Specifically, in this phase we collected 1000 Web resources.

4.2 Features Extraction and Labeling

In this section, we provide details about the features engineered from the data collected (Section 4.1) and we explain how samples have been labeled.

Features. The choice of the set of features is one of the most significant step during the definition of a machine learning system because features must describe, in the best possibile way, the input samples. In our case, given we resource, the set of features has to describe the information about the HTTP traffic associated with. Features engineered are:

1. *Number of Set-Cookie* included in the HTTP request associated to a single resource: a huge number of Set-Cookie is used to send different information to more tracking domains (see Figure 2).
2. *Number of Cookies* sent by a resource at the moment of the request: tracking, analytics or advertisement resources usually send more cookies to send different user's information (see Figure 3 for an example).
3. *Length of Sent Cookies*: the 80% of tracking cookies contains more than 35 characters (Li et al., 2015), unlike the ones provided by functional resources. This behaviour is justified by the quantity of information trasmitted to tracking domains.
4. *Cookie Duration*: in order to learn more and more user's preferences, a tracker has to observe users habits for a long time. In (Li et al., 2015) authors showed that the 90% of cases tracking cookies lasted more than 6 months.
5. *URL Length*: information can also be sent through HTTP GET method, instead of cookies. In this case, data are sent as parameters in the URL, increasing consequently its length.
6. *Number of Parameters in HTTP GET Request*: this feature is related to the URL lenght, indeed tracking, analytics or advertisement resources tend to send more data, thus parameters to share different user's information.
7. *Sent/Received Bytes Ratio*: in HTML pages, trackers insert some invisible resources, i.e., images or iFrames with a dimension of 0x0, 0x1, 768x0, etc. named Web bugs (Malandrino and Scarano, 2013). Such content does not have a functional aim for the webpage, it is actually used just to track the user's habit on the page. The Web bugs still send an HTTP request from the client to the server. Then the server takes the information about the user, embedded in the HTTP (cookie, referer, user-agent, and so on), and reply to the client with an empty frame or image. This result in a difference between sent bytes (information about the user) and the received bytes.

Labels. Samples have been manually labeled. Specifically, as proposed in (Lai, 2019), we asked to three experts in the privacy field to independently label the resources as functional (with label `0`) or tracking (with label `1`). We assigned the label according to the majority voting. Samples are evenly balanced between tracking and functional ones i.e., 500 has label functional while the latter 500 has tracking.

4.3 Validation and Testing

In this section we detail the validation and testing phase conducted by means of Weka (Hall et al., 2009).

[8]https://www.alexa.com/topsites/category

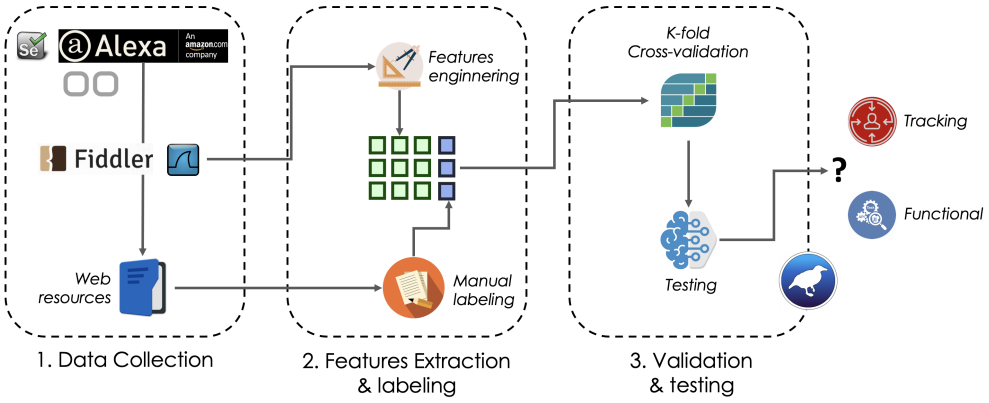

Figure 1: Steps of the approach for classifying Web resources.

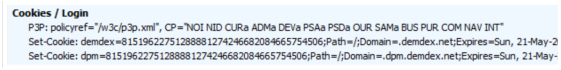

Figure 2: An example of tracking request which set different cookies from demdex domain, specifically from both *dpm.demdex.net* and *demdex.net*.

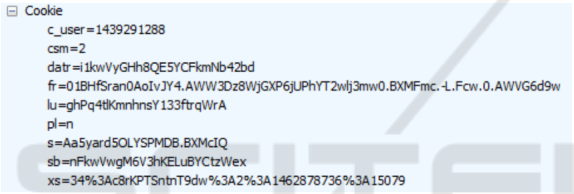

Figure 3: An example of several cookies set at request time by a tracker. A unique user id has been set (*c_user*) togheter with other information associated with him, like *datr* used to understand if the user access by multiple devices.

We split (with a stratified approach) the dataset into: (1) the *train set*, obtained including the 80% of the samples (randomly chosen), and (2) the *test set*, obtained including the remaining 20% of the samples. Thus, the train set contains 800 samples (400 functional and 400 tracking), while the test set contains 200 samples (100 functional and 100 tracking).

We used a scaler to normalize the data according to the min-max technique,i.e.,

$$z_i = \frac{x_i - min(x)}{max(x) - min(x)}$$

where x_i is the true feature value for i-th sample, z_i is the normalized value, and $min()$ and $max()$ compute minimum and maximum value respectively for the feature x.

Then, we first validate the RF, J48, NB, MLP, SVM models (see Section 2) in a `k`-fold cross-validation phase on the train set, second we perform the testing phase on the test set (i.e., unseen samples). `k`-fold cross-validation is a resampling procedure employed to evaluate machine learning models on a limited samples. The function has a single parameter called `k` that specifies the number of groups that a given dataset (train set in our case) has to be split into. We set $k = 5$ and $k = 10$. Informally, the function takes the first of the `k` groups as validation set and the remaining `k-1` as train set (James et al., 2013).

Experiments have been conducted using `default` parameters in Weka for each model on a machine with `16 GB 2133 MHz LPDDR3` memory, and `Intel 2,8 GHz Intel Core i7 quad-core` processor. Results of these experiments can be found in Table 1. As we can see, oveall all the models except the Naive-Bayes classifier showed good performance (more than 85% accuracy in both validation and testing phases). Specifically, Random Forest model stands out with a 91% accuracy in the testing phase, and up to 92% average accuracy in the validation phase. In Figure 4 we show the confusion matrix for the Random Forest. The confusion matrix shows predicted labels (by RF) against true labels (in the test set), i.e., the number of tp, tn, fp, fn (see Section 2). Thus, the Random Forest classifier shows a precision, computed as $\frac{(tp)}{(tp+fp)}$, of about 88,8%, while a recall, computed as $\frac{(tp)}{(tp+fn)}$, of 95,2%. We observe that some functional resources (15) have been classified as tracking. The reason is that those resources were from Content Delivery Networks and exibits similar HTTP traffic as for tracking ones because at the same time provides functionalities and analytics to the website (Falahrastegar et al., 2014).

5 CONCLUSION

Privacy on the Web is a significative problem today, and as more technologies join the Internet as more threats to the users' rights researchers will have to

Table 1: Average (avg.) accuracy achieved by the machine learining models in the 10-fold (5-fold) cross-validation phase (*Validation*), and accuracy achieved in the testing phase on the test set (*Testing*). In **bold** the best result.

Model	Validation k=10		Validation k=5		Testing
	Avg. accuracy	*Std. Error*	*Avg. accuracy*	*Std. Error*	*Accuracy*
NB	0,77	0,03	0,76	0,05	0,79
RF	**0,92**	0,01	**0,92**	0,01	**0,91**
MLP	0,89	0,02	0,88	0,02	0,87
SVM	0,88	0,02	0,87	0,03	0,86
J48	0,89	0,01	0,90	0,01	0,87

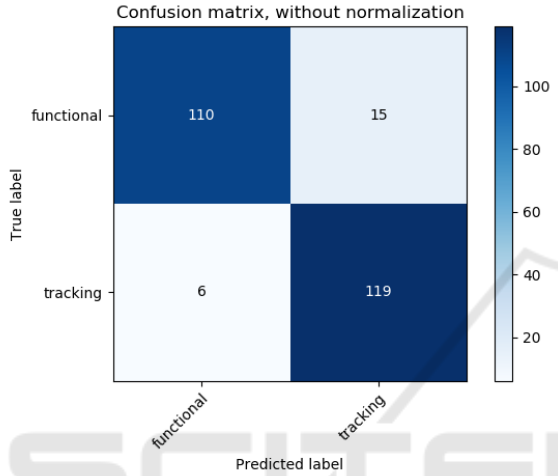

Figure 4: Confusion matrix for Random Forest model.

face (Hassan and De Filippi, 2017; De Filippi and Hassan, 2018; Larsson, 2018): not only privacy violation issues, for instance the ones related to the use of mobile technologies (Razaghpanah et al., 2018) or Internet of Things (Lin and Bergmann, 2016), but also issues related to the job protection (Lettieri et al., 2019), for instance the concerns about Uber (Berger et al., 2018) or Amazon Mechanical Turk (McInnis et al., 2016), child protection (Kumar and Sachdeva, 2019), and so on. In this paper we studied a part of those threats, in particular the 3rd-party tracking to derive whether it is possible to distinguish between functional and tracking resources to provide privacy against threats online. Existing privacy tools mainly rely on blacklisting mechanisms which are limited because such lists must be manually updated and maintained. The experiments with machine learning to classify Web resources based on HTTP traffic features showed positive results, with the Random Forest classifier able to classify functional and tracking resources with an accuracy of 91%. Our result shows therefore a new perspective in embedding the use of machine learning in a privacy tool such as AdBlock or Ghostery, i.e., a client-side application. The idea is to develop a browser extension powered by machine learning which is capable to filter out tracking resources, and fill a customized blacklist accordingly. The blacklist will be customized because each user has its own browsing habit, therefore its blacklist will include only the threats the user is exposed to. In this way, we reduce the time needed for the machine learning computation for each Web resource that has already been processed by the tool. Such privacy tool can include also visual analytics techniques (Guarino et al., 2019; Lettieri et al., 2018; Keim et al., 2008; Wang, 2018) to give the user the right level of awareness while browsing, for instance by showing the graph of the HTTP requests performed (and blocked) during the browsing session.

REFERENCES

Avasarala, S. (2014). *Selenium WebDriver practical guide*. Packt Publishing Ltd.

Berger, T., Chen, C., and Frey, C. B. (2018). Drivers of disruption? estimating the uber effect. *European Economic Review*, 110:197–210.

Bhagavatula, S., Dunn, C., Kanich, C., Gupta, M., and Ziebart, B. (2014). Leveraging Machine Learning to Improve Unwanted Resource Filtering. In *Proceedings of the 2014 Workshop on Artificial Intelligent and Security Workshop*, AISec.

Binns, R., Lyngs, U., Van Kleek, M., Zhao, J., Libert, T., and Shadbolt, N. (2018). Third party tracking in the mobile ecosystem. In *Proceedings of the 10th ACM Conference on Web Science*, pages 23–31. ACM.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. springer.

Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.

Butler, K. T., Davies, D. W., Cartwright, H., Isayev, O., and Walsh, A. (2018). Machine learning for molecular and materials science. *Nature*, 559(7715):547.

Camacho, D. M., Collins, K. M., Powers, R. K., Costello, J. C., and Collins, J. J. (2018). Next-generation machine learning for biological networks. *Cell*, 173(7):1581–1592.

Cosimato, A., De Prisco, R., Guarino, A., Lettieri, N., Malandrino, D., Sorrentino, G., and Zaccagnino, R.

(2019). The conundrum of success in music: playing it or talking about it? *IEEE Access*, pages 1–10.

D'Ambrosio, S., Pasquale, S. D., Iannone, G., Malandrino, D., Negro, A., Patimo, G., Scarano, V., Spinelli, R., and Zaccagnino, R. (2017). Privacy as a proxy for Green Web browsing: Methodology and experimentation. *Computer Networks*, 126:81–99.

De Filippi, P. and Hassan, S. (2018). Blockchain technology as a regulatory technology: From code is law to law is code. *arXiv preprint arXiv:1801.02507*.

Deeba, F., Mohammed, S. K., Bui, F. M., and Wahid, K. A. (2016). Learning from imbalanced data: A comprehensive comparison of classifier performance for bleeding detection in endoscopic video. In *2016 5th International Conference on Informatics, Electronics and Vision (ICIEV)*, pages 1006–1009. IEEE.

Domingos, P. M. (2012). A few useful things to know about machine learning. *Commun. acm*, 55(10):78–87.

Falahrastegar, M., Haddadi, H., Uhlig, S., and Mortier, R. (2014). The rise of panopticons: Examining region-specific third-party web tracking. In *International Workshop on Traffic Monitoring and Analysis*, pages 104–114. Springer.

Gervais, A., Filios, A., Lenders, V., and Capkun, S. (2017). Quantifying web adblocker privacy. In *European Symposium on Research in Computer Security*, pages 21–42. Springer.

Guarino, A., Lettieri, N., Malandrino, D., Russo, P., and Zaccagnino, R. (2019). Visual analytics to make sense of large-scale administrative and normative data. In *2019 23rd International Conference Information Visualisation (IV)*, pages 133–138. IEEE.

Guha, S., Cheng, B., and Francis, P. (2011). Privad: Practical privacy in online advertising. In *USENIX conference on Networked systems design and implementation*, pages 169–182.

Haddadi, H., Hui, P., and Brown, I. (2010). Mobiad: private and scalable mobile advertising. In *Proceedings of the fifth ACM international workshop on Mobility in the evolving internet architecture*, pages 33–38. ACM.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. (2009). The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18.

Hanke, J. and Thiesse, F. (2017). Leveraging Text Mining for the Design of a Legal Knowledge Management System. In Ramos, I., Tuunainen, V., and Krcmar, H., editors, *ECIS*.

Hassan, S. and De Filippi, P. (2017). The expansion of algorithmic governance: From code is law to law is code. *Field Actions Science Reports. The journal of field actions*, (Special Issue 17):88–90.

Hossin, M. and Sulaiman, M. (2015). A review on evaluation metrics for data classification evaluations. *International Journal of Data Mining & Knowledge Management Process*, 5(2):1.

Ikram, M., Asghar, H. J., Kaafar, M. A., Mahanti, A., and Krishnamurthy, B. (2017). Towards seamless tracking-free web: Improved detection of trackers via one-class learning. *Proceedings on Privacy Enhancing Technologies*, 2017(1):79–99.

Interactive Advertising Bureau (IAB) (2019). Interactive Advertising Bureau (IAB) and PricewaterhouseCoopers (PwC) US. Internet Advertising Revenue Report. https://www.iab.com/news/iab-advertising-revenue-q1-2019/.

Iqbal, U., Snyder, P., Zhu, S., Livshits, B., Qian, Z., and Shafiq, Z. (2020). Adgraph: A graph-based approach to ad and tracker blocking. In *Proc. of IEEE Symposium on Security and Privacy*.

James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013). *An introduction to statistical learning*, volume 112. Springer.

Keim, D. A., Mansmann, F., Schneidewind, J., Thomas, J., and Ziegler, H. (2008). Visual analytics: Scope and challenges. In *Visual data mining*, pages 76–90. Springer.

Kovács, I., Mihéltz, K., Kránitz, K., Juhász, É., Takács, Á., Dienes, L., Gergely, R., and Nagy, Z. Z. (2016). Accuracy of machine learning classifiers using bilateral data from a scheimpflug camera for identifying eyes with preclinical signs of keratoconus. *Journal of Cataract & Refractive Surgery*, 42(2):275–283.

Krishnamurthy, B., Naryshkin, K., and Wills, C. (2011). Privacy leakage vs. protection measures: the growing disconnect. In *Proceedings of the Web*, volume 2, pages 1–10.

Krishnamurthy, B. and Wills, C. E. (2010). Privacy leakage in mobile online social networks. In *Proceedings of the 3rd Wonference on Online social networks*, pages 4–4. USENIX Association.

Kumar, A. and Sachdeva, N. (2019). Cyberbullying detection on social multimedia using soft computing techniques: a meta-analysis. *Multimedia Tools and Applications*, pages 1–38.

Kushmerick, N. (1999). Learning to Remove Internet Advertisements. In *Proceedings of the Third Annual Conference on Autonomous Agents*, AGENTS '99, pages 175–181.

Lai, M. (2019). *On Language and Structure in Polarized Communities*. PhD thesis.

Larsson, S. (2018). Law, society and digital platforms: Normative aspects of large-scale data-driven tech companies. In *The RCSL-SDJ Lisbon Meeting 2018" Law and Citizenship Beyond The States*.

Lawrence, E. (2012). *Debugging with Fiddler: The complete reference from the creator of the Fiddler Web Debugger*. Eric Lawrence.

Lettieri, N., Guarino, A., and Malandrino, D. (2018). E-science and the law. three experimental platforms for legal analytics. In *JURIX*, pages 71–80.

Lettieri, N., Guarino, A., Malandrino, D., and Zaccagnino, R. (2019). Platform economy and techno-regulationexperimenting with reputation and nudge. *Future Internet*, 11(7):163.

Li, T.-C., Hang, H., Faloutsos, M., and Efstathopoulos, P. (2015). Trackadvisor: Taking back browsing privacy from third-party trackers. In *International Conference*

on Passive and Active Network Measurement, pages 277–289. Springer.

Lin, H. and Bergmann, N. (2016). Iot privacy and security challenges for smart home environments. *Information*, 7(3):44.

Liu, Y. and Simpson, A. (2016). Privacy-preserving targeted mobile advertising: Formal models and analysis. In *Data Privacy Management and Security Assurance*, pages 94–110. Springer.

Malandrino, D. and Scarano, V. (2013). Privacy leakage on the web: Diffusion and countermeasures. *Computer Networks*, 57(14):2833–2855.

Maron, M. E. (1961). Automatic indexing: an experimental inquiry. *Journal of the ACM (JACM)*, 8(3):404–417.

Mayer, J. R. and Mitchell, J. C. (2012). Third-party web tracking: Policy and technology. In *2012 IEEE symposium on security and privacy*, pages 413–427. IEEE.

McInnis, B., Cosley, D., Nam, C., and Leshed, G. (2016). Taking a hit: Designing around rejection, mistrust, risk, and workers' experiences in amazon mechanical turk. In *Proceedings of the 2016 CHI conference on human factors in computing systems*, pages 2271–2282. ACM.

Orebaugh, A., Ramirez, G., and Beale, J. (2006). *Wireshark & Ethereal network protocol analyzer toolkit*. Elsevier.

Orr, C. R., Chauhan, A., Gupta, M., Frisz, C. J., and Dunn, C. W. (2012). An Approach for Identifying JavaScript-loaded Advertisements Through Static Program Analysis. In *Proceedings of the 2012 ACM Workshop on Privacy in the Electronic Society*, WPES '12.

Pan, X., Cao, Y., and Chen, Y. (2015). I do not know what you visited last summer: Protecting users from third-party web tracking with trackingfree browser. In *Proceedings of the 2015 Annual Network and Distributed System Security Symposium (NDSS), San Diego, CA*.

Parra-Arnau, J. (2017). Pay-per-tracking: A collaborative masking model for web browsing. *Information Sciences*, 385:96–124.

Parra-Arnau, J., Achara, J. P., and Castelluccia, C. (2017). Myadchoices: Bringing transparency and control to online advertising. *ACM Transactions on the Web (TWEB)*, 11(1):7.

Peterson, L. E. (2009). K-nearest neighbor. *Scholarpedia*, 4(2):1883.

Quinlan, J. R. (2014). *C4. 5: programs for machine learning*. Elsevier.

Rahwan, I., Cebrian, M., Obradovich, N., Bongard, J., Bonnefon, J.-F., Breazeal, C., Crandall, J. W., Christakis, N. A., Couzin, I. D., Jackson, M. O., et al. (2019). Machine behaviour. *Nature*, 568(7753):477.

Razaghpanah, A., Nithyanand, R., Vallina-Rodriguez, N., Sundaresan, S., Allman, M., Kreibich, C., and Gill, P. (2018). Apps, trackers, privacy, and regulators: A global study of the mobile tracking ecosystem.

Roesner, F., Kohno, T., and Wetherall, D. (2012). Detecting and defending against third-party tracking on the web. In *Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation*, pages 12–12. USENIX Association.

Rumelhart, D. E., Hinton, G. E., Williams, R. J., et al. (1988). Learning representations by back-propagating errors. *Cognitive modeling*, 5(3):1.

Rzecki, K., Pawiak, P., Niedwiecki, M., Sonicki, T., Lekow, J., and Ciesielski, M. (2017a). Person recognition based on touch screen gestures using computational intelligence methods. *Information Sciences*, 415-416:70 – 84.

Rzecki, K., Pławiak, P., Niedźwiecki, M., Sośnicki, T., Leśkow, J., and Ciesielski, M. (2017b). Person recognition based on touch screen gestures using computational intelligence methods. *Information Sciences*, 415:70–84.

Shaw, B., Suman, A. K., and Chakraborty, B. (2020). Wine quality analysis using machine learning. In *Emerging Technology in Modelling and Graphics*, pages 239–247. Springer.

Swets, J. A. (1988). Measuring the accuracy of diagnostic systems. *Science*, 240(4857):1285–1293.

Toubiana, V., Narayanan, A., Boneh, D., Nissenbaum, H., and Barocas, S. (2010). Adnostic: Privacy preserving targeted advertising. In *Proceedings Network and Distributed System Symposium*.

Wang, C. (2018). Graph-based techniques for visual analytics of scientific data sets. *Computing in Science & Engineering*, 20(1):93–103.

Wang, L. (2005). *Support vector machines: theory and applications*, volume 177. Springer Science & Business Media.

Wu, Z., Ramsundar, B., Feinberg, E. N., Gomes, J., Geniesse, C., Pappu, A. S., Leswing, K., and Pande, V. (2018). Moleculenet: a benchmark for molecular machine learning. *Chemical science*, 9(2):513–530.