Nameles: An intelligent system for Real-Time Filtering of Invalid Ad Traffic

Antonio Pastor*
Universidad Carlos III de Madrid

Matti Pärssinen
Aalto University

Patricia Callejo
IMDEA Networks Institute
Universidad Carlos III de Madrid

Pelayo Vallina
IMDEA Networks Institute
Universidad Carlos III de Madrid

Rubén Cuevas
Universidad Carlos III de Madrid

Ángel Cuevas
Universidad Carlos III de Madrid

Mikko Kotila
Botlab

Arturo Azcorra
IMDEA Networks Institute
Universidad Carlos III de Madrid

ABSTRACT

Invalid ad traffic is an inherent problem of programmatic advertising that has not been properly addressed so far. Traditionally, it has been considered that invalid ad traffic only harms the interests of advertisers, which pay for the cost of invalid ad impressions while other industry stakeholders earn revenue through commissions regardless of the quality of the impression. Our first contribution consists of providing evidence that shows how the Demand Side Platforms (DSPs), one of the most important intermediaries in the programmatic advertising supply chain, may be suffering from economic losses due to invalid ad traffic. Addressing the problem of invalid traffic at DSPs requires a highly scalable solution that can identify invalid traffic in real time at the individual bid request level. The second and main contribution is the design and implementation of a solution for the invalid traffic problem, a system that can be seamlessly integrated into the current programmatic ecosystem by the DSPs. Our system has been released under an open source license, becoming the first auditable solution for invalid ad traffic detection. The intrinsic transparency of our solution along with the good results obtained in industrial trials have led the World Federation of Advertisers to endorse it.

ACM Reference Format:
Antonio Pastor, Matti Pärssinen, Patricia Callejo, Pelayo Vallina, Rubén Cuevas, Ángel Cuevas, Mikko Kotila, and Arturo Azcorra. 2019. Nameles: An intelligent system for Real-Time Filtering of Invalid Ad Traffic. In *Proceedings of the 2019 World Wide Web Conference (WWW'19), May 13–17, 2019, San Francisco, CA, USA*. ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3308558.3313601

1 INTRODUCTION

Online advertising is a major social and economic driver of the so-called *Information Society*. First, online advertising sponsors free

---

*Corresponding author: anpastor@it.uc3m.es

offerings of essential services to billions of users, such as Online Search Services, Map Services, and Social Media. Second, the market volume of online advertising reached, only in the US, $88 B in 2017 with an inter-annual growth rate of 21% [30]. Third, online advertising represents an important source of jobs. For instance, recent studies have estimated that 1 M direct and 6 M indirect jobs were associated with online advertising in the EU-28 workforce in 2015 [38]. Fourth, online advertising represents the fundamental source of income of the companies at the forefront of technological innovations such as Google or Facebook [2, 13]. Therefore, it is in the best interest of everyone (citizens, governments and the private sector) to guarantee the sustainable growth of this business. However, this sustainability is in jeopardy due to several factors. Arguably, the most important is the high volume of *invalid ad traffic*, i.e., delivered ads not shown to humans. It is estimated that every year trillions of delivered ad impressions are not watched by humans leading to losses of tens of billions of dollars for advertisers [18, 28].

Unfortunately, the identification and filtering of invalid ad traffic has not been properly addressed so far due to two fundamental reasons. First, a rapidly increasing fraction of ad transactions occur through a programmatic ecosystem, where a chain of intermediaries automatically connects advertisers willing to show ads and publishers owning the inventory (websites, mobile apps) to show those ads. This automatic process makes the detection of invalid traffic complex. Second, intermediaries in programmatic advertising receive a commission for each delivered ad, regardless if it is invalid or not. Then, it is well accepted the idea that invalid traffic only harms the interests of advertisers, which pay for the cost of the invalid ad impressions. Intermediaries in the supply chain get a commission for each served invalid impression and then they do not have direct monetary incentives to fight invalid traffic effectively.

Specialized companies referred to as verification vendors (e.g., IAS [20], DoubleVerify [11], White Ops [42]) have emerged offering opaque proprietary solutions for the identification of invalid traffic. Previous research has shown that even simple attack vectors can defeat these opaque defenses [8, 26]. These solutions do not properly address the concerns of advertisers, which have become

increasingly vocal about the uncertainty of the quality of programmatic media transactions [12, 39, 41] and the lack of transparency in the ecosystem [4, 9].

To meet the demands of advertisers, in this paper, we present the first open source (and thus auditable) solution for the detection of invalid ad traffic in programmatic advertising.

Prior to designing our solution, we have revisited the common idea that advertisers are the only stakeholders affected by invalid ad traffic in programmatic advertising. We present an economic model based on real financial reports of Demand Side Platforms (DSPs)–a key intermediary in the programmatic advertising ecosystem–and realistic assumptions on the operational set-up of DSPs, which provides initial evidence that invalid ad traffic seems to impact the business model of DSPs negatively. This finding suggests, contrary to the conventional wisdom, that DSPs may have strong incentives to filter invalid ad traffic. Our analysis concludes that post-bid (i.e., non real-time) detection of invalid traffic does not solve the problem for the DSPs. Instead, DSPs require a solution that can identify invalid traffic in real-time and at the level of individual ad transactions. Moreover, DSPs handle up to tens of billions of ad transactions per day, a factor imposing demanding computational performance constraints to the invalid traffic detection problem.

The main contribution of this paper is *Nameles*, an open source invalid ad traffic detection system that operates in real-time at the level of individual requests. Therefore, it meets the requirements of both advertisers and DSPs. Nameles identifies anomalous ad requests patterns of domains using an algorithm based on Shannon entropy. Nameles has been built in accord with the latest version of the OpenRTB specification [23] and is able to handle up to 500 k ad transactions per second, adding a total delay of less than 3 ms to each ad transaction. As a result, it can be seamlessly integrated into the programmatic supply-chain as a solution for the DSPs. We have applied Nameles on a stream of ~1.8 B daily bid request in two periods of two months separated one year from each other, observing that (on average) 20% of the daily ad traffic can be safely considered invalid.

Nameles' code is publicly available (under an open source license) and then auditable by anyone. This intrinsic transparency, along with the good detection performance shown by the system in extensive industry trials, has led the main global advertiser trade-body, the World Federation of Advertisers (WFA), to endorse Nameles.

2 BACKGROUND

In this section, we describe the process of serving an ad in programmatic advertising. We do it with an example of a user visiting a website, but the process is the same for ads shown in mobile apps.

A user connects to a website offering some ad spaces. Each ad space is typically leased by the website's publisher to an ad network or a Supply Side Platform (SSP), which upon the user's connection generates an ad request. This ad request may be forwarded through several intermediaries until it reaches the Ad Exchange. This part of the process is known as the *sell-side* and is represented by Step 1 in Figure 1. The ad request includes the domain name, IP address of the device, the User Agent, user's cookie(s), etc. Upon the reception of the ad request, the Ad Exchange initiates an auction which represents the *buy-side* of the programmatic process.

The bidding process of this auction is standardized by the Open-RTB protocol [23]. First, the Ad Exchange processes the information included in the ad request to generate a bid request whose format is specified by the OpenRTB standard [23]. For simplicity, in this paper, we will consider that a bid request includes the IP address receiving the ad and the domain name selling the ad space. Each bid request is sent to the Demand Side Platforms (DSPs) registered in the Ad Exchange. A DSP is an intermediary where advertisers, or their agencies, configure their programmatic advertising campaigns. Therefore, upon the reception of a bid request a DSP checks if the request meets the configuration parameters of any of its advertising campaigns and if so, it creates a bid response including, among other information, the price the advertiser is willing to pay for this ad space. Note that the bid responses to a given bid request have to be received by the Ad Exchange within 120 ms [15]. The Ad Exchange runs an auction based on the received bid responses and informs all the participant DSPs about the selected winner bid. The bidding process is represented by Steps 2-4 in Figure 1. To finalize the programmatic process the Ad Exchange coordinates the delivery of an URL from where to retrieve the ad, which is immediately downloaded by the browser and shown to the user. This is represented by step 5 in Figure 1.

From a business perspective, each bid event corresponds with an opportunity to place an online advertisement on a web page for the advertiser, and an opportunity to monetize an ad placement for the publisher. Based on their respective commission percentages (a reference of them obtained from insights from the industry is presented in Figure 1), the intermediaries are compensated every time a bid is successfully transacted, and an ad is displayed as a result. However, the advertiser only benefits when the traffic associated with the transaction is valid. This business model is open to fraudulent activity [18, 28, 29] (e.g., a publisher monetizing visits to a website coming from bots) whereas it seems not to offer the right incentives to intermediaries to identify and filter invalid ad traffic. Note that, according to various industry guidelines [14, 41], *invalid traffic is defined to correspond with those bid events where displaying an ad would not have any potential for advertising effect and the advertiser would lose its investment without getting anything in return.*

Verification companies (e.g., IAS, Double Verify or White Ops) have emerged recently offering proprietary opaque solutions for filtering invalid traffic. However, the lack of transparency on the used techniques makes it difficult to assess their actual capabilities. Indeed, recent works have demonstrated inefficiencies in these solutions [8, 26] and different studies attribute billions of dollars wasted in invalid traffic every year [18, 28, 29].

3 ECONOMIC IMPACT OF INVALID TRAFFIC IN DSP COMPANIES

In this section, we refute the argument that advertisers are the only stakeholders in the programmatic ecosystem negatively affected by invalid traffic [27]. To this end, we provide qualitative and quantitative economic analyses that support how, under realistic assumptions, invalid traffic negatively impacts the profitability of DSP companies.
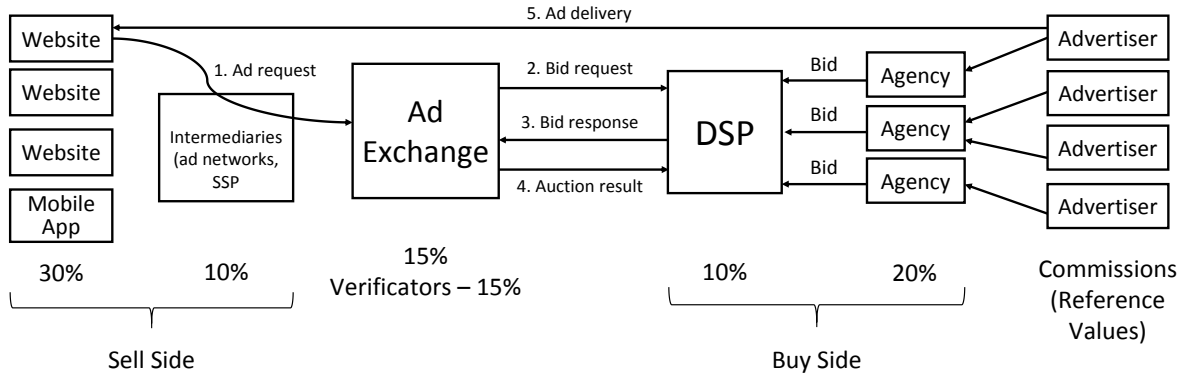
Figure 1: Overview of the programmatic advertising ecosystem operation.

-**Qualitative Analysis:** We investigated seven publicly listed DSP companies through their annual income statements and found that only one company had a positive net income. Moreover, depending on the DSP company, variable costs ranged from 30% to 50% of the revenue. Variable costs represent a proportion of total costs that vary as a function of revenue. These findings indicate that the low profitability of analyzed DSPs has a strong correlation with variable costs.

The DSP win-rate [9] is defined as the fraction of won bids out of all auctions. Regardless if an auction the DSP is hosting results in a win or not, the DSP bears the cost for facilitating that auction. Then, the inverse of the win-rate indicates how much a DSP company accumulated variable costs that yield no economic revenue. While valid bids represent a real opportunity for advertisers that provides an intangible value even to lost valid bids, in the case of invalid bids there is not a real opportunity and thus, lost invalid bids contribute exclusively to increase DSPs costs.

Based on interviews we conducted with DSP companies, we conclude that the DSP win-rate is typically between 5% to 20%. An individual advertiser win-rate has been shown to be in the range 0.1% to 1% [44], and ad exchanges (and ad networks) *fill-rate* (i.e., the fraction of successfully completed auctions) is commonly below 40% [35]. Consequently, we assert that there is an oversupply of programmatic ad inventory. This supports the economic viability of invalid traffic filtering so that even after removing the invalid ad traffic there will be still enough ad inventory available for DSPs.

In summary, all the above objective facts show that filtering invalid traffic would contribute to reducing the accumulated variable costs of DSPs without affecting the availability of ad inventory and as a result would lead to improving profitability and valuation of a DSP company.

Finally, to maximize the profitability of the DSP company, invalid bid requests should be identified in real time in the pre-bid stage, so that variable costs incurred by processing such invalid bids are minimized since the processing of the bid is stopped in the first step of the procedure.

-**Quantitative Analysis:** Net Present Value (NPV) model is the tool of choice for financial forecasting because it considers the time value of money, and provides a concrete metric to financial decision-makers, such as investors, for evaluating investment against the

Table 1: Impact of invalid traffic filtering to economics of DSPs.

| | Enterprise Value (EV) | | | Net Present Value (NPV) | | |
| | No filtering | Filtering | | No filtering | Filtering | |
		Max [F=0,23; P=0]	Min [F=1; P=1]		Max [F=0,23; P=0]	Min [F=1; P=1]
DSP-1	10,544	19,002	-3,269	4,421	8,020	-1,979
DSP-2	2,516	5,194	-3,518	536	1,672	-2,213
DSP-3	3,254	3,833	-1,147	1,237	1,481	-706
DSP-4	1,184	2,973	-2,376	133	892	-1,498
DSP-5	1,702	2,648	-819	634	1,035	-506
DSP-6	1,354	2,342	-1,005	445	863	-628
DSP-7	1,595	2,262	-2,118	310	592	-1,337
Average	3,163	5,468	-2,036	1,101	2,079	-1,267

predicted return [3]. Finance theory endorses an investment if NPV is positive and higher than NPV of an alternative investment [3]. In addition to the NPV, we evaluated Enterprise Value (EV) [21], a useful variant of the NPV, that takes into account cash flows beyond the forecasted time window. Positive NPV and EV values are reached when the cash inflows exceed cash outflows [3]. NPV and EV are widely used as decision-making tools for planning purchases, mergers or acquisitions [3].

We compute NPV and EV for two scenarios; without invalid traffic filtering (Scenario A), and with filtering (Scenario B). The timeframe of the analysis is eight years. NPV and EV are computed based on five key factors:

1) Annual growth rates. In our analysis, they are based on the industry average of seven publicly listed DSPs' annual and quarterly income statements between 2012-2015 [16] and are the same for both scenarios.

2) Rate of return r. We use $r = 20\%$, which is a typical value for investments made into new products [3, 22], for both scenarios.

3) Invalid traffic filtering rate F. We consider $F = 0$ for Scenario A and F between 0 and 100% for Scenario B.

4) Revenue penalty P (as a dependent factor of F). We have selected the parameters of the penalty function to make the penalty increase in an exponential manner, such that the penalty is low until $F = 20 - 30\%$ and it spikes after this point until F reaches 100% where all traffic is filtered. This function has been carefully constructed so that a low penalty is imposed for filtering rates up to the average reported fraction of invalid traffic from different studies [10, 43].

5) Long-term cash flow growth rate G, which is set-up to 2% in both scenarios [25].

Our results show that there are NPV and EV gains for the DSP when the filtering rate increases from zero towards $F = 23\%$. Filtering invalid traffic beyond $F > 23\%$ first results in diminishing benefit and eventually drives a decline in revenue for the DSP. These results confirm our hypothesis that filtering invalid traffic (at a reasonable rate) improves DSPs profitability due to a reduction in variable costs. Table 1 shows the minimum and maximum values of NPV and EV, which correspond to [F;P] values of [0,23;0] and [1;1] in Scenario B, respectively. We observe that at the optimal filtering rate, NPV and EV in Scenario B increase (on average) 1,72 and 1,89 times in comparison with Scenario A, respectively.

Finally, we would like to note that the results obtained in this section represent a reference example based on realistic assumptions and should be interpreted as such.

4 DATASET

The dataset used in this paper includes a daily sample of incoming bid requests stream data collected in two different periods one year apart and each spanning two months between Dec 2016-Jan 2017 and Dec 2017-Jan 2018, respectively.

The data is from one of the largest DSPs with a significant global presence. The data consist of desktop and mobile bid events, for video, banner and in-app inventory. In particular, each daily sample includes between 1.7-1.9 Billion actual bid requests issued on that date from ~50 Ad Exchanges. These bid requests are associated (on average) to ~150 M IP addresses and ~900 k domains per day. The dataset includes the following information per bid request: a unique identifier, the IP address[1] and User Agent of the device initiating the ad request and the Web Domain or Mobile Application ID selling the ad space. For simplicity, we refer to both Web Domains and Mobile Apps as *Domains* along the paper.

5 SYSTEM REQUIREMENTS, DESIGN AND IMPLEMENTATION

In this section we describe *Nameles*, an open source system for the detection of invalid ad traffic that operates in real time and at the level of individual bid requests, thus meeting the requirements of DSPs and advertisers.

5.1 System's Functional Requirements

1. Scalability: DSPs typically handle tens of billions of bid requests per day. This maps into peaks of hundreds of thousands of bid request per second. Nameles must be capable of handling these high rates of bid requests.

2. Delay: The bid response to a given bid request has to be received by the Ad Exchange within 120 ms [15]. Hence, the delay introduced by Nameles should be limited to a few milliseconds in order to minimize the impact in the overall bidding process delay.

3. Accuracy in invalid traffic identification: Providing 100% guarantee that a bid request is invalid (or not) is not feasible. Instead, it is more reliable providing a score indicating the likelihood that a
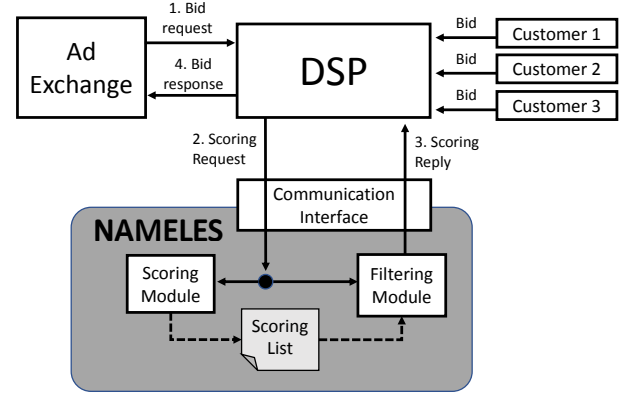
---

[1]We anonymize the IP address before any processing.

Figure 2: Programmatic Ecosystem Scheme with Nameles.

bid request is invalid. Therefore, our system must incorporate an accurate scoring algorithm.

5.2 System Design

In this subsection, we present a brief overview of the system functionality and how it is integrated with the programmatic advertising ecosystem. Then, we describe in detail each of the functional blocks forming Nameles.

5.2.1 Overview. Figure 2 depicts a high-level representation of Nameles functional blocks. Moreover, the figure shows how Nameles could be integrated into the programmatic ad delivery chain as an auxiliary service for DSPs. The only difference with respect to the current operation of a DSP would be that, as part of the pre-bid phase, the DSP makes a request to Nameles to provide a Confidence Score per bid request. To this end, the DSP sends a *scoring request* to Nameles (step 2 in Figure 2). The scoring request includes the following fields: bid request id (to allow mapping Nameles result to the corresponding bid request), IP address of the device associated with the ad request and the domain offering the ad space. This information is included in the bid requests as defined in the OpenRTB protocol standard [23]. The *scoring request* is delivered to two independent modules of Nameles: the *Scoring* module and the *Filtering* module.

Because the DSP has limited information about a bid request to determine if it is invalid or not, we propose to aggregate all bid requests from a domain and use statistical analysis to determine the level of confidence of a domain. This approach provides statistically robust Confidence Scores for domains since they are computed from a sample of (at least) hundreds of bid requests. Then, Nameles assigns to the bid requests from a domain the Confidence Score of such domain. The *Scoring Module* is responsible for computing the Confidence Score for domains present in the bid requests received by the DSP. Moreover, it groups the domains in four different Confidence Classes. This information is summarized in the *Scoring List*.

The *Filtering* module is responsible for classifying in real-time each received *scoring request*. To this end, it retrieves the domain id from the *scoring request* and obtains the domain's Confidence Score and Confidence Class from the *Scoring List* introduced above.

After that, it creates a *scoring reply* to be sent to the DSP (Step 3 in Figure 2). This reply includes the following information: bid request id (extracted from the corresponding scoring request), the domain Confidence Score, and the domain Confidence Class.

Finally, the *Communication Interface Module* handles the communication between the DSP and Nameles.

5.2.2 Communication Interface Module. This module manages the delivery of scoring requests from the DSP to Nameles and scoring replies in the opposite direction. We have opted to use a parallel pipeline communication structure. In particular, the DSP creates two queues: a sending queue used for pushing scoring requests to Nameles and a receiving queue for pulling scoring replies from Nameles in return. Nameles sets up a number of worker processes, which connect to the sockets associated with both queues. These workers pull scoring requests from the sending queue and forward them to the Scoring and Filtering modules. The resulting scoring replies are pushed by the workers to the receiving queue of the DSP.

This parallel pipeline communication structure is highly scalable, being able to handle streams of hundreds of thousands of requests per second with processing delays below 3 ms (see Section 6.2).

5.2.3 Scoring Module. The goal of the scoring module is to produce a *Scoring List* of domains to be used by the *Filtering* module. This list is updated daily. Since Nameles operates in real-time, the list used at day d is obtained from a prediction algorithm applied on the historical *Confidence Score* values of domains at days $d - 1$, $d - 2$, $d - 3$, ... This module implements three different algorithms to produce the *Scoring List*: one to compute the Confidence Score of each domain, a second to compute the Confidence Classes, and a third to derive the *Scoring list* to be used at day d based on historical information. Next, we describe each of these algorithms.

- **Confidence Score computation:** A DSP can reconstruct the traffic pattern associated with a given domain X by analyzing the distribution of a number of requests across the IP addresses included in the bid requests associated to X. This is the fundamental signal used by our algorithm. Skewed distributions, where most bid requests come from just a few IP addresses, are for obvious reasons suspicious[2] and thus domains presenting such traffic patterns should be assigned low Confidence Scores. Instead, legit traffic patterns correspond to more homogeneous distributions of bid requests across IPs and domains presenting such distributions should receive high Confidence Scores.

We compute the Shannon Entropy [33] of the distribution of bid requests across IP addresses for each domain. The Shannon Entropy summarizes in a single value the level of determinism of a distribution and ranges between 0 (all bid requests to a domain come from a single IP address) and $log_2(n)$ (the bid requests are homogeneously distributed across the n IP addresses making ad requests to the domain).

Shannon entropy has been successfully used in the field of anomaly detection [24, 40]. However, in our case, it has an important limitation because it does not consider the volume of bid requests, but just the shape of the distribution of bid requests. This limitation

---

[2]For instance, a domain receiving most of its visits from scrapers or from other types of bots associated with fraudulent ad traffic.
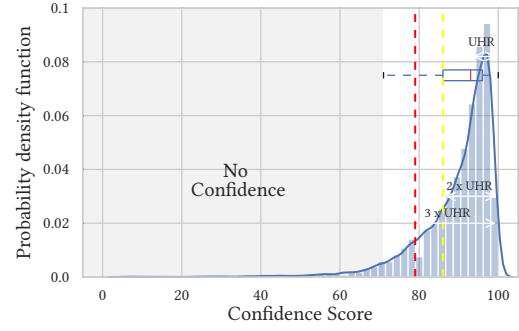
Figure 3: Distribution of Confidence Score (CS) values for domains with more than 500 bid requests at January 15, 2018.

prevents the direct comparison of domains with different traffic volumes. For instance, a domain with 5 bid requests uniformly distributed across 5 IPs would have the same Entropy value (2.32) than a domain with 5000 bid requests homogeneously distributed across 5 IPs. While the first domain is just an unpopular domain, the second one is highly suspicious, having a high number of daily visits distributed evenly across a small number of IPs.

To address this limitation, we propose a normalization process that takes into account the volume of bid requests associated with a domain X. In essence, we compute the ratio of the Shannon entropy and the binary logarithm of the total number of bid requests ($C(X)$) and scale the resulting value to a normalized range between 0 and 100. This normalized entropy score is the *Confidence Score* (CS) assigned to domains by Nameles, and its formal expression is:

$$CS(X) = 100 \left(1 - \frac{\sum_{i=1}^{n} C(x_i) \, log_2\,(C(x_i))}{C(X) \, log_2\,(C(X))}\right) \tag{1}$$

To get an intuition on the effect of this normalization process, we can consider the toy example mentioned above. The domain with 5 bid requests from 5 IP addresses would have a high CS equal to 100 whereas the domain with 5000 bid requests would have a low CS equal to 19.

- **Computation of the Confidence Classes:** We first analyzed the probability distribution function of the CS values across domains in our daily datasets. Figure 3 shows this distribution for a specific day. Note that other days in our dataset showed similar distributions. We observed a skewed distribution concentrated in the high CS values with a long tail towards low CS values. This distribution indicates that most domains present homogeneous traffic patterns (represented by high CS) while as we move towards lower values of CS, we find domains with infrequent (i.e., statistically unlikely) traffic patterns offering lower confidence.

To define the Confidence Classes, we use two different unsupervised statistical methods that divide the distribution in 4 ranges each representing a single Confidence Class:
- *Outlier detection method*: This method identifies outlier CS values based on the definition of traditional outliers [31], i.e., $CS(X) < 25\ percentile - 1.5 \times IQR$. Nameles uses this expression to define the threshold for the *No Confidence* Class including domains with an extremely deterministic and infrequent traffic pattern.

- *Dispersion method*: We defined intermediate Confidence Classes between the one formed by outliers and the one composed by the mass of legit domains. To this end, we use the Upper Half Range[3] (UHR) of the distribution as our dispersion metric and define two new thresholds as $max(CS) - 2 \times UHR$ and $max(CS) - 3 \times UHR$. Based on these thresholds we defined the following Confidence Classes:

- *Low Confidence Class*: formed by domains whose CS falls in the range $max(CS) - 3\,UHR > CS \geq 25\,percentile - 1.5\,IQR$.
- *Moderate Confidence Class*: formed by domains whose CS falls in the range $max(CS) - 2\,UHR > CS \geq max(CS) - 3\,UHR$.
- *High Confidence Class*: formed by domains whose CS falls in the range $CS \geq max(CS) - 2\,UHR$.

Figure 3 shows the defined Confidence Classes for the Confidence Score distribution of the dataset of Jan 15, 2018.

- **Predicting the Scoring List:** The Scoring list used at day d has to be inferred from a prediction algorithm applied to the historical Confidence Score values of domains at days $d - 1$, $d - 2$, $d - 3$, . . . We refer to the estimated CS value of a domain X included in this list as $CS_d^*(X)$. To define the prediction algorithm, we first studied the stationary properties of the temporal series of CS values of domains across the 62 days forming each of our datasets. This analysis revealed that CS values present high stationarity, with 40% of the domains in our dataset being strictly stationary (with a 90% confidence interval), as reported by the Augmented Dickey-Fuller test [32]. The analysis of the autocorrelation and partial autocorrelation functions for these domains revealed that in general, only the CS of the previous day ($CS_{d-1}(X)$) contributes significantly to the prediction of $CS(X)$ at day d. Then, the best predictor is $CS_d^*(X)$ = $CS_{d-1}(X)$ and the Scoring List to be used at day d is formed by the $CS_d^*(X)$ of the different domains in our dataset.

As a result of the application of the three described algorithms, the Scoring Module produces each day a Scoring List that includes both the Confidence Score and the Confidence Class for each individual domain.

5.2.4 Filtering Module. This module processes in real-time each scoring request received from the Communication Interface module. In particular, it extracts the domain from the scoring request and searches for the $CS_d^*(X)$ and the Confidence Class associated with the domain in the Scoring list. As a result of this process, the Filtering Module generates a *scoring reply* message including the following information: Bid Request ID (obtained from the corresponding scoring request) and the domain's CS and Confidence Class. The scoring reply is sent to the DSP through the Communication Interface module. Note that if the domain extracted from the scoring request is not present in the Scoring list, the scoring reply has the following content <bid request id, NULL, NULL>.

5.3 System Implementation

In this subsection, we describe our implementation of Nameles that meets the performance and scalability requirements defined in Subsection 5.1.

5.3.1 The Communication Interface and Filtering module. The Communication Interface and the Filtering modules address different functional aspects of Nameles, and thus we have described them separately in Section 5.2. In our Nameles prototype, we use an integrated implementation of these two functional modules for efficiency purposes. We implement the parallel pipeline communication structure described in Section 5.2.2 on top of ZeroMQ [19] (a highly scalable distributed messaging system) using the existing Java bindings for this purpose. On the Nameles side, we use 6 workers that in addition to taking care of the pull and push communication functions, implement the filtering process. Each worker is an independent process, which has an independent copy of the Scoring List hash table produced by the Scoring Module allocated in RAM. Moreover, each worker pulls independently scoring requests from the DSP's sending queue. For each scoring request, it extracts the domain, obtains the CS and Confidence Class associated with the domain from the Scoring List hash table, creates the scoring reply and pushes it to the DSP's receiving queue.

5.3.2 The Scoring Module. The Scoring Module implements a temporary hash table including the number of bid requests associated with each pair <domain, IP>. For each new bid request, the counter of the tuple <domain, IP> included in the bid request is increased by 1. At the end of every day, the resulting hash table includes the needed information to compute the Confidence Score for each domain as well as the thresholds to define the different Confidence Classes. For this purpose, we store this temporary table into a PostgreSQL database and use different PostgreSQL functions and Java scripts to obtain the CS and the Confidence Class of each domain. The final result of the process is the Scoring List, which is stored in a hash table using as a key the domain id and as value the tuple <CS, Confidence Class>. This table is transferred to the "Communication Interface and Filtering" module to be used in the real-time filtering of bid requests.

6 PERFORMANCE EVALUATION OF THE SYSTEM

We have deployed a realistic experimental set-up to confirm that our Nameles prototype meets the requirements defined in Section 5.1.

6.1 Experimental Set-up

The experimental set-up replicates a production set-up used by a large-scale DSP. In particular, we use three servers in our setup for Nameles. The first server plays the role of the DSP. This server uses the real stream of bid requests from our dataset to produce a stream of scoring requests to Nameles. The rate of scoring requests is a configurable parameter so that we can perform stress tests by using significantly higher rates of bids per second than the ones reflected in our dataset. The second server deploys the "Communication Interface and Filtering" module of our Nameles prototype. It receives the stream of scoring requests from the DSP server and processes it to obtain the scoring replies. In addition, this server forwards the scoring requests to a third server, which implements the "Scoring" module.

The server emulating the DSP is a Dell PowerEdge R710 with 16 cores and 48 GB of RAM. The servers implementing the "Communication and Filtering" and the "Scoring" modules are similar,
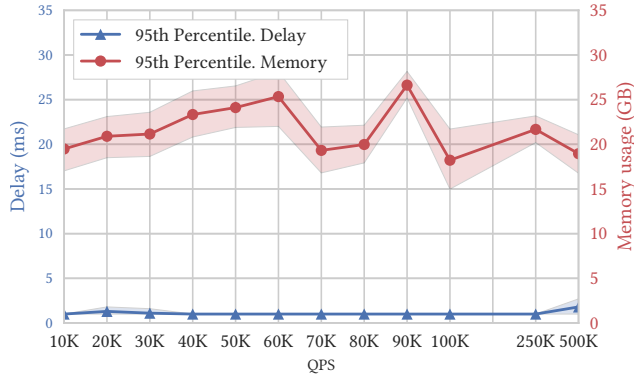
Figure 4: 95 percentile of delay and memory consumption for the filtering process at different input request rates.

a Dell PowerEdge R730xd with 24 cores and 64 GB of RAM. The servers are connected to a conventional 1 Gbps Ethernet switch. In the context of common use in the Adtech industry, the resources employed in our prototype can be considered commodity hardware.

6.2 Scalability and Processing Delay

6.2.1 Scoring List computation time. A critical aspect of the scalability of Nameles resides in its ability to produce the Scoring List in a short time. Specifically, given that the Scoring List is updated daily, the computation process must guarantee that the new list is ready before the expiration of the previous one, i.e., in less than 24 h. We have measured the computation time for every daily dataset, including between 1.7-1.9 B bid requests, and confirmed that it is always shorter than 4 hours. Hence, Nameles meets the scalability requirements for this critical process.

6.2.2 Delay and memory consumption of the filtering process. From the DSP's perspective, the filtering process starts when it sends a Scoring Request and finishes when it receives the corresponding Scoring Reply. The analysis of our dataset reveals an average and a peak rate of 22 k and 26 k requests per second, respectively. Then, our prototype must meet the next two requirements while processing scoring requests streams at the observed peak rate: not overflowing the memory of the server and offering a small delay to minimize its impact on the overall delay of the real-time bidding process.

We have evaluated the performance of our prototype for scoring request streams ranging from 10 k to 500 k queries per second (QPS). For each of the analyzed rates, we run stress tests of 5 minutes. During the tests, we measure the individual delay associated with the filtering process of each scoring request as well as the overall memory consumption of the filtering process. Figure 4 summarizes the performance of our Nameles prototype. The x-axis shows the different tested scoring request rates. The left y-axis and right y-axis show the 95-percentile filtering delay and 95-percentile memory consumption measured during the experiment for the different scoring request rates (QPS), respectively. Note that each stress test has been run 5 times. The line in the figure represents the average of 95-percentile values across the five experiments whereas the lighter color area shows the max and min 95-percentile values.

First of all, we observe that the system performance is quite stable across the different experiments and the observed variability in memory consumption is due to the instantaneous load of the server at the measurement moment rather than the QPS of the experiment. The results of the stress tests demonstrate that our Nameles prototype offers very high scalability performance. In particular, the 95 percentile of memory consumption and delay is lower than 28 GB and 3 ms for any of the considered QPS. These results prove that our filtering process scales to handle more than 20 B bid requests per day with a modest infrastructure, meeting the requirements of the largest DSPs such as Google, The Trading Desk and MediaMath.

6.3 Scoring Accuracy

In this subsection, we assess both the accuracy of our prediction algorithm and the accuracy of the Confidence Scores assigned to domains.

6.3.1 Accuracy of the prediction algorithm. For each daily dataset, we have computed the Root Mean Square Error (RMSE) of the difference between the predicted CS ($CS_d^*(X)$) and the actual CS ($CS_d(X)$) across all domains. The results indicate that the RMSE is smaller than 3 points in every case.

In addition, we have evaluated the misclassification rate of domains among Confidence Classes. We observe that the average misclassification rates across all days in our dataset are below 3% between any pair of classes. A careful analysis of the misclassified domains indicates that the classification errors are mainly associated with domains having a CS close to the threshold that separates two contiguous classes. This is also coherent with the fact that misclassifications between non-contiguous classes are negligible (< 0.3%).

6.3.2 Assessment of Confidence Score accuracy. The accuracy of the Confidence Score cannot be objectively evaluated. There are various continuously changing factors related to the invalid traffic problem; attack vectors, domain traffic profiles, and others. As a result, ground truth datasets for evaluating invalid traffic filtering solutions do not exist. Indeed, to the best of the authors' knowledge, previous related academic works [6, 37] rely on manual validation whereas proprietary commercial solutions do not offer any publicly available accuracy evaluation analysis.

In this subsection, we present for the first time an objective approach to (at least partially) assess the accuracy of invalid ad traffic detection tools. While we acknowledge our approach is far from ideal, it does not suffer from the severe limitations associated with manual validation such as: 1) limited scalability since only a handful of domains can be manually evaluated; 2) it suffers from human errors due to the lack of sufficient knowledge, lack of motivation, distractions, etc.

In particular, we propose a two-fold approach to validate the classification accuracy of Nameles:
(1) We compare the result of Nameles scoring with the following widely used ad tech metrics:
- **Bounce Rate:** This metric measures the fraction of sessions that only visit a single page in a domain. A low bounce rate is a strong indication of low-quality traffic.

Table 2: Value of external quality metrics associated with domains in each of the defined Confidence Classes in our dataset.

		No Conf.		Low Conf.		Moderate Conf.		High Conf.
Alexa Upstream traffic from Google and Facebook (%)	median	20	(-41%)	18.5	(-45%)	23.7	(-30%)	33.7
	IQR	21.05		20.19		24.19		32.84
Alexa Bounce rate (%)	median	41.8	(-27%)	40.9	(-29%)	35.3	(-39%)	57.5
	IQR	32.4		25.6		27.7		28.7
Alexa Search traffic (%)	median	8.1	(-35%)	7.7	(-38%)	5.5	(-56%)	12.5
	IQR	19.7		15.9		16.1		16.9
Alexa Total sites linking to the domain	median	9.2	(-75%)	131	(-62%)	256	(-27%)	348
	IQR	616		371		800		1,198
SimilarWeb Bounce rate (%)	median	51.5	(-12%)	38.8	(-34%)	34.9	(-40%)	58.6
	IQR	24.97		20.84		24.8		24.0
SimilarWeb Direct traffic (%)	median	43.1	(68%)	34.2	(34%)	38.1	(49%)	25.6
	IQR	39.5		37.0		34.6		27.8
SimilarWeb Search traffic (%)	median	21.2	(-31%)	29.3	(-5%)	19.5	(-37%)	30.9
	IQR	39.3		46.5		39.5		39.8

- Traffic from popular publishers: This metric represents the percentage of upstream traffic coming to the domain from popular publishers. In particular, the two publishers contributing a larger fraction of traffic to domains are Google and Facebook. Then, for our validation, we will compute the fraction of upstream traffic coming from Google and Facebook to a domain. A very low fraction of traffic coming from them may reveal the presence of low-quality traffic.

- Search Traffic: This metric measures the percentage of traffic coming to the domain from search engines. A very low search traffic percentage is often an indication of low-quality traffic.

- Direct Traffic: This metric measures the percentage of traffic that reaches the domain directly without being redirected from other websites. In this case, a large fraction of direct traffic is usually linked to low-quality traffic.

- Number of sites linking to a domain: An interesting domain attracting high-quality traffic would typically be linked from a large number of other sites. Contrary to this, domains associated with ad fraud or other malicious practices would typically be linked from a lower number of sites.

To obtain these metrics, we have queried two well-known services, Alexa [1] and SimilarWeb [34]. Table 2 presents the median and IQR values for the distribution of each one of these metrics for each Confidence Class. In addition, the table shows the relative difference of the median values of these metrics for the "No", "Low" and "Moderate" Confidence Classes in comparison to the "High" Confidence Class. We observed substantial differences (up to 75% in some cases) between the "High Confidence" Class and the rest.

(2) Low-quality domains and thus their ad inventory are likely to have a short-term lifetime (i.e., being more volatile) in the online advertising ecosystem for multiple reasons: they might be committing fraud and being detected and blacklisted or removed by fraudsters after short periods of times to avoid detection; they provide poor KPIs and thus the bidding prices for their inventory dramatically decrease making them no longer profitable, etc. We have measured the volatility associated with the domains of each of the confidence classes defined by Nameles. In particular, we have computed the fraction of domains of each class in our oldest dataset that are still present one year later in our most recent dataset. The result shows

that (4.5; 1.2; 6.4; 35.0)% of domains within the ("No", "Low", "Moderate" and "High") confidence classes in our 16-17 dataset are still present in our 17-18 dataset. We observe that the volatility of the "High" Confidence Class is one order of magnitude smaller than in other classes[4].

The results from our two-fold objective validation suggest that our scoring mechanism accurately identifies legitimate domains and thus it is suitable for adoption by DSPs. Finally, it is also worth noting that we have worked together with experts from the Ad Tech industry over a period of 18 months to let them subjectively evaluate the results provided by Nameles in extensive trials. The satisfactory results obtained during these tests have led the World Federation of Advertisers to endorse Nameles.

7 RESULTS OBTAINED FROM NAMELES' EXECUTION

In this section, we present the results obtained from applying Nameles to our large-scale dataset. First, we analyze the distribution of domains and traffic across the defined Confidence Classes. Then, we quantify the positive impact that Nameles may have in the profitability of DSPs using as reference the economic model described in Section 3.

-Longitudinal Analysis of domains' confidence level: Figure 5 shows the fraction of domains and ad traffic (i.e., bid requests) belonging to each of the defined Confidence Classes in our 16-17 and 17-18 datasets, respectively. The main bar shows the average, and the error bar shows the 95% confidence interval across the days in the dataset.

On average, in our 16-17 dataset (11.2; 8.8; 33.3; 46.7)% of the traffic is associated with ("No", "Low", "Moderate" and "High") Confidence Classes. Whereas in our 17-18 dataset (7.0; 13.9; 36.3; 42.8)% of the traffic belong to the ("No", "Low", "Moderate" and "High") classes. We find an overall stable behavior across time. In particular, there exists ~20% low-quality traffic (belonging to the "No" and "Low" confidence classes) in both datasets. These results imply that, for instance, a DSP handling 50 B bid request per day using a policy that filters traffic belonging to "No" and "Low" classes would

---

[4]There is an apparent systemic volatility since even in the high confidence class only 35.0% of the domains remain in the 17-18 dataset. While there exists such systemic volatility, the reported one is likely to be inflated because our datasets represent a subsample of the overall ad inventory available at the measurement time.
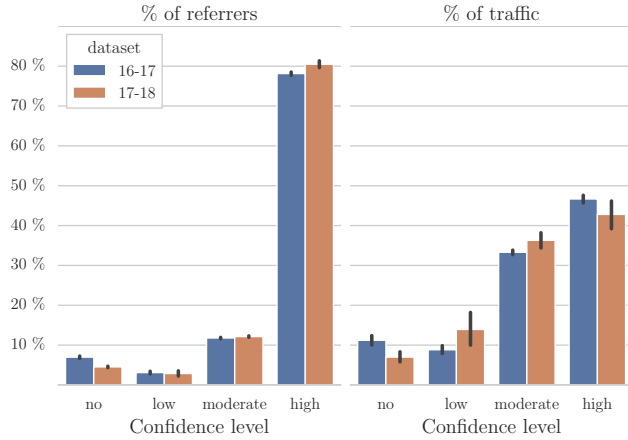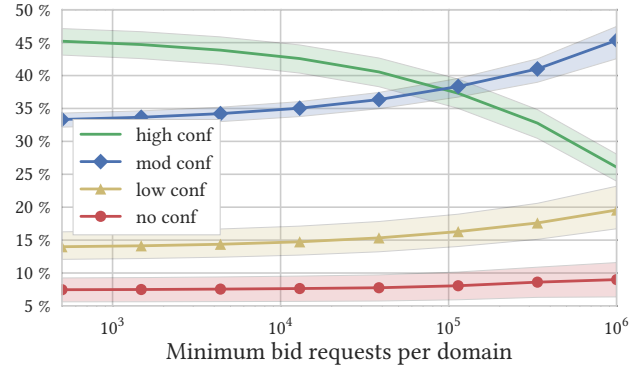
Figure 5: Percentage of domains and ad traffic in each of the Confidence Classes across the 62 days of the dataset. The main bar presents the average value whereas the error bars show the confidence interval at the 95%.

Figure 6: Percentage of ad traffic in each Confidence Class as function of the popularity (i.e., bid requests) of domains.

eliminate (on average) around 10 B (20%) bid requests every day. However, it is worth mentioning that there seems to be a transfer of ~5% traffic from the "No" to the "Low" confidence class between the 16-17 and 17-18 datasets, this may be interpreted as a slight improvement in the behavior of the ecosystem for DSPs filtering traffic at the "No" confidence level. Contrary, a transfer of ~4% traffic is observed from the "High" to the "Moderate" confidence class, which may likewise be interpreted as a slight worsening of the ecosystem behavior for DSPs accepting traffic exclusively from the "High" confidence class.

In addition, we analyzed how popularity relates to confidence. To this end, we computed the average (and standard deviation) fraction of traffic within each Confidence Class for domains with at least 500, 1 k, 10 k, 50 k, 100 k, and 1 M bid requests per day for our 16-17 dataset. Figure 6 shows the results. One may expect that as more popular domains are considered, the fraction of domains within the "High" Confidence Class would increase and the fraction in other groups would decrease. However, we observe the opposite trend between "Moderate" (which increases) and "High" (which decreases) classes. In the case of "Low" and "No" Classes we observe just a light increase after 100 k daily bid requests.

-**Nameles' impact on DSPs' profitability:** The results in the previous subsection provide specific figures on the filtering rates of Nameles at different confidence levels. For instance, in our 16-17 dataset, a filtering rate of 11.21% filters out traffic from domains with very rare traffic patterns that offer no confidence. A filtering rate of 20.04% eliminates traffic offering low or no confidence, and a filtering rate of 53.34% filters any domain that does not provide high confidence.

Using these filtering rates as input to the economic model presented in Section 3 gives us an estimation of the impact that Nameles is expected to have in the profitability of a DSP. The obtained results indicate that filtering at the "No Confidence", "Low Confidence" and "Moderate Confidence" level offer NPV (and EV) improvements in

comparison to the scenario without filtering of 41, 54 and -204% (14, 19 and -71%). We observe that filtering at the "Moderate Confidence" level would not be recommended. On the other hand, filtering at the "No Confidence" or "Low Confidence" class leads to a strong positive economic impact. Note that results considering the 17-18 datasets are consistent.

8 NAMELES' APPLICABILITY AND EXTENSIBILITY

- **Applicability by other players:** In addition to DSPs, Nameles can be easily integrated with other players of the programmatic advertising supply chain such as Ad Exchanges or SSPs, which handle a representative fraction of the ad traffic of a given domain. Also, Publishers can integrate Nameles to self-generate the CS of their websites and mobile apps.

-**Extensibility to other signals:** The concept of entropy allows us to compute the CS considering different signals. In particular, in addition to the default signal used in this paper (CS of domains), we have computed the CS of individual IP addresses. To this end, we consider the traffic pattern generated by each IP address as the distribution of ad requests it sends across different domains. Having the CS for the IP and the domain within a bid request enriches the decision capacity of the DSP since such bid request can be dropped due to a low CS associated with the IP address as well. We have computed the CS for all IP addresses (with more than 500 entries) for every daily sample in our dataset and based on it, we have re-calculated the fraction of traffic belonging to the Confidence Classes more likely to be filtered by DSPs' policies (i.e., *None* and *Low* Confidence). Adding the information about the CS of IP addresses increases the fraction of traffic in these two categories in less than 0.5%. This result indicates that the application of Nameles at the level of domains suffices to identify more than 99% of low-quality ad traffic.

9 NAMELES' LIMITATIONS

In this section, we discuss the main limitations of Nameles and argue why despite them, Nameles is still an important contribution.

- **False Negatives:** They are represented by the invalid traffic not identified by Nameles. In the presence of Nameles, any attacker owning a domain d would be undetected if it is able to generate a normal (i.e., similar to the mass) traffic pattern for d. This is obviously doable, but it would require the attacker to increase the complexity of the attack. First, the attacker would need to infer the CS threshold over which it would not be detected. Second, an attacker performing n daily visits to its domain from m IPs leading to a low CS would need to either reduce n or increase m to increase its CS over the quality threshold defined by different DSPs. Both approaches lead to a revenue reduction.

To obtain a ballpark estimation of such reduction, we have computed the number of daily visits that all 8 K domains in the "No" and "Low" Confidence Classes in a given day of our dataset would need to remove in order to pass the threshold of the "Moderate" Class. We found that on average these domains would need to eliminate 38,5% visits leading to a roughly similar reduction in their revenue. Therefore, even when failing in the detection, Nameles contributes to reducing the profitability of possible attacks significantly.

-**False Positives:** They are represented by domains wrongly assigned a low CS. While false positives may have serious implications in other businesses, it is well-established in the programmatic advertising industry that false positives are not an issue for the buy-side, i.e., advertisers and DSPs, which are the target of our solution. The existing oversupply of ad spaces discussed in Section 3 guarantees that wrongly filtering a legitimate domain would not result in a lost opportunity of placing an ad that would be provided by other legitimate non-filtered domain. Therefore, we assert that false positives are not an important consideration in adopting Nameles.

- **Presence of NATs:** The widespread use of NATs may pollute the computation of a domain's CS since visits coming from different users behind a NATed IP address would be all assigned to that IP address, and thus Nameles could consider the traffic pattern of that domain more deterministic than it actually is. To analyze this potential limitation, we have re-computed the CS of all domains in our dataset using as input the visits coming from <IP, User Agent> pairs. The obtained results are similar to those presented in the paper. Given that the impact of NATs when considering <IP, User Agent> pairs as input is expected to be limited, we conclude that Nameles is minimally affected by the presence of NATs.

10 RELATED WORK

In addition to the abovementioned commercial proprietary solutions [11, 20, 42], the research community has also addressed the identification of invalid ad traffic. The proposed solutions focus on detecting invalid traffic at the sell side of the online advertising chain, i.e., publishers web pages [7] or delivered ads [5, 17]. These solutions analyze the interaction of the user with the web page or the served ad in order to identify commonly known attacks such as visits generated by bots [5] or redirection attacks [36]. None of these solutions are valid for DSPs. To the best of the author's knowledge, Stitelmant et al. [37] proposed the only alternative solution to Nameles able to operate at the DSP level. By analyzing the degree of overlapping in the IPs visiting two (or more) domains,

their solution identifies potential invalid traffic. This is a complementary technique to our normalized entropy score, and thus both solutions can be used in parallel.

From a methodological perspective, there is a previous work that has used entropy to identify invalid video visits to a Chinese video portal [6]. The authors of this paper propose to use entropy as the final metric to assess the traffic quality and a semi-supervised classification that rely on manually labeled samples to differentiate between valid and invalid video traffic. However, as discussed in Section 5, the native Shannon entropy has an important drawback since its interpretation depends on the volume of associated events. To overcome this limitation, we use a Confidence Score based on a normalized version of entropy. Moreover, instead of using manual labeling of suspicious traffic, we define unsupervised statistically supported outlier detection method. Hence, Nameles clearly advance the state-of-the-art from a methodological perspective as well.

11 CONCLUSION

This paper introduces Nameles, a system for the detection of invalid ad traffic, which is one of the main problems faced by the online advertising industry. Nameles has been designed to meet the requirements of both advertisers and DSPs that together form the so-called buy-side of the programmatic advertising industry. On the one hand, Nameles is the first available open source solution for the identification of invalid traffic, responding to the advertisers' demand for transparency. On the other hand, the paper provides economic supported evidence that, contrary to conventional wisdom, show how DSPs may increase their profitability with invalid traffic filtering. For this, the applied solution needs to be highly scalable and operate in real time and at the level of individual bid requests. Nameles meets these requirements.

A Nameles prototype has been thoroughly tested in a realistic deployment. We demonstrate that even with modest resources, Nameles is able to process tens of billions of bid requests per day, with processing delays below 3ms per request and good detection accuracy. Moreover, applying Nameles on two datasets including each almost 2 B bid requests per day for a period of two months, we observe the presence of 20% invalid traffic. The evidenced performance of Nameles along with our open-source vision has led the World Federation of Advertisers to endorse Nameles as a solution to counter invalid traffic by the Adtech industry.

REFERENCES

[1] Alexa: actionable analytics for the web 2018. Retrieved Oct 31, 2018 from http://www.alexa.com
[2] Inc. Alphabet. 2018. 2017 Annual report. Retrieved Oct 31, 2018 from https://abc.xyz/investor/static/pdf/20171231_alphabet_10K.pdf
[3] Richard A Brealey, Stewart C Myers, Franklin Allen, and Pitabas Mohanty. 2012. *Principles of corporate finance*. Tata McGraw-Hill Education.
[4] Internet Advertising Bureau. 2017. *Transparency is the key to programmatic success*. Technical Report. Retrieved Oct 31, 2018 from https://www.iab.com/wp-content/

uploads/2015/08/IABDigitalSimplifiedProgrammaticAdvertisingTransparency. pdf

[5] Patricia Callejo, Ruben Cuevas, Angel Cuevas, and Mikko Kotila. 2016. Independent Auditing of Online Display Advertising Campaigns. In *Proceedings of the 15th ACM Workshop on Hot Topics in Networks*. ACM.

[6] Liang Chen, Yipeng Zhou, and Dah Ming Chiu. 2014. Fake view analytics in online video services. In *Proceedings of Network and Operating System Support on Digital Audio and Video Workshop*. ACM, 1.

[7] Vacha Dave, Saikat Guha, and Yin Zhang. 2013. Viceroi: catching click-spam in search ad networks. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*. ACM, 765–776.

[8] Shailin Dhar. 2016. *Mystery Shopping Inside the Ad Fraud Verification Bubble.* Technical Report. Retrieved Oct 31, 2018 from http://www.slideshare.net/ShailinDhar/mystery-shopping-inside-the-adverification-bubble

[9] Digiday. 2016. WTF is programmatic? Retrieved Oct 31, 2018 from http://digiday.com/wp-content/uploads/2016/07/WTF_programmatic-2016.pdf

[10] Distil. 2015. Distil Networks Releases New Data on The State of Digital Advertising Fraud. Retrieved Oct 31, 2018 from https://resources.distilnetworks.com/press-releases/distil-networks-releases-new-data-on-the-state-of-digital-advertising-fraud

[11] Double Verify 2018. Retrieved Oct 31, 2018 from http://www.doubleverify.com

[12] Emarketer. 2016. The Ad Industry's Focus on Fraud Has Intensified. Retrieved Oct 31, 2018 from https://www.emarketer.com/Article/Ad-Industrys-Focus-on-Fraud-Has-Intensified/1014430

[13] Inc. Facebook. 2018. 2017 Annual report. Retrieved Oct 31, 2018 from https://s21.q4cdn.com/399680738/files/doc_financials/annual_reports/FB_AR_2017_FINAL.pdf

[14] Joint Industry Committee for Web Standards. 2016. *Traffic Fraud: Good Practice Principles for Reducing Risk to Exposure to Ad Fraud.* Technical Report. Retrieved Oct 31, 2018 from https://jicwebs.org/wp-content/uploads/2018/02/JICWEBS_Good_Practice_Principles_Ad_Fraud_May_2016.pdf

[15] Google. 2018. DoubleClick RTB Protocol. Latency Restrictions and Peering. Retrieved Oct 31, 2018 from https://developers.google.com/ad-exchange/rtb/peer-guide

[16] Gurufocus 2018. Retrieved Oct 31, 2018 from http://www.gurufocus.com

[17] Hamed Haddadi. 2010. Fighting online click-fraud using bluff ads. *ACM SIGCOMM Computer Communication Review* 40, 2 (2010), 21–25.

[18] Lucy Handley. 2017. Businesses could lose $16.4 billion to online advertising fraud in 2017: Report. Retrieved Oct 31, 2018 from http://www.cnbc.com/2017/03/15/businesses-could-lose-164-billion-to-online-advert-fraud-in-2017.html

[19] Pieter Hintjens. 2013. *ZeroMQ: messaging for many applications.* "O'Reilly Media, Inc.".

[20] Integral Ad Science (IAS) 2018. Retrieved Oct 31, 2018 from https://integralads.com

[21] Investopedia. 2017. DCF Analysis: Coming Up With A Fair Value. Retrieved Oct 31, 2018 from http://www.investopedia.com/university/dcf/dcf4.asp

[22] Investopedia. 2018. Rate of Return. Retrieved Oct 31, 2018 from www.investopedia.com/terms/r/rateofreturn.asp

[23] IAB Technology Laboratory. [n. d.]. *Real Time Bidding (RTB) Project. OpenRTB API Specification Version 2.5.*

[24] Wenke Lee and Dong Xiang. 2001. Information-theoretic measures for anomaly detection. In *Security and Privacy, 2001. S&P 2001. Proceedings. 2001 IEEE Symposium on*. IEEE, 130–143.

[25] Investopedia. J.B. Maverick. 2015. What is the long-term average growth rate of the telecommunications sector? Retrieved Oct 31, 2018 from http://www.investopedia.com/ask/answers/071515/what-longterm-average-growth-rate-telecommunications-sector.asp

[26] Marciel Miriam, Cuevas, Ruben, Banchs Albert, Gonzalez Roberto, Traverso Stefano, Ahmed Mohamed, and Azcorra Arturo. 2016. Understanding the Detection of View Fraud in Video Content Portals. (2016).

[27] World Federation of Advertisers. 2016. *Compendium of ad fraud knowledge for media investors.* Technical Report.

[28] ANA & White Ops. 2016. *2015 Bot Baseline: Fraud in Digital Advertising.* Technical Report.

[29] White Ops. 2016. *The Methbot Operation.* Technical Report.

[30] PwC. 2018. *IAB internet advertising revenue report 2017 full year results.* Technical Report. Retrieved Oct 31, 2018 from https://www.iab.com/wp-content/uploads/2018/05/IAB-2017-Full-Year-Internet-Advertising-Revenue-Report.REV_.pdf

[31] P. J. Rousseeuw and A. M. Leroy. 1987. *Robust Regression and Outlier Detection.* John Wiley & Sons, Inc., New York, NY, USA.

[32] Said E Said and David A Dickey. 1984. Testing for unit roots in autoregressive-moving average models of unknown order. *Biometrika* 71, 3 (1984), 599–607.

[33] C. E. Shannon. 1948. A mathematical theory of communication. *The Bell System Technical Journal* 27, 3 (July 1948), 379–423.

[34] SimilarWeb: Website Traffic & Mobile App Analytics 2018. Retrieved Oct 31, 2018 from https://www.similarweb.com

[35] Smaato. 2014. Mobile RTB Insights Report Q3 2014. Retrieved Oct 31, 2018 from http://www.newsroom-publicismedia.fr/wp-content/uploads/2014/11/Smaato_Q3_2014_RTB_Report.pdf

[36] Kevin Springborn and Paul Barford. 2013. Impression Fraud in On-line Advertising via Pay-Per-View Networks. In *22nd USENIX Security Symposium (USENIX Security 13)*. USENIX, 211–226.

[37] Ori Stitelman, Claudia Perlich, Brian Dalessandro, Rod Hook, Troy Raeder, and Foster Provost. 2013. Using co-visitation networks for detecting large scale online display advertising exchange fraud. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 1240–1248.

[38] IHS Technology. 2017. *The Economic Contribution of Digital Advertising in Europe.* Technical Report. https://datadrivenadvertising.eu/wp-content/uploads/2017/09/DigitalAdvertisingEconomicContribution_FINAL-1.pdf

[39] Financial Times. 2017. Google charges for YouTube ads even when viewed by robots. Retrieved Oct 31, 2018 from https://www.ft.com/content/f9da727c-6207-11e5-9846-de406ccb37f2

[40] Arno Wagner and Bernhard Plattner. 2005. Entropy based worm and anomaly detection in fast IP networks. In *Enabling Technologies: Infrastructure for Collaborative Enterprise, 2005. 14th IEEE International Workshops on*. IEEE, 172–177.

[41] WFA. 2014. WFA guide to Programmatic Media. Retrieved Oct 31, 2018 from https://www.wfanet.org/app/uploads/2017/04/programmatic.pdf

[42] White Ops 2018. Retrieved Oct 31, 2018 from https://www.whiteops.com

[43] IMPERVA INCAPSULA. Igal Zeifman. 2017. Bot Traffic Report 2016. Retrieved Oct 31, 2018 from https://www.incapsula.com/blog/bot-traffic-report-2016.html

[44] Weinan Zhang, Shuai Yuan, and Jun Wang. 2014. Optimal real-time bidding for display advertising. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 1077–1086.