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

Anonymous Author(s)

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 at individual bid request level in real time. 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 in 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.

1 INTRODUCTION

Online advertising is a major social and economic driver of the so-called *Information Society*. First, online advertising sponsors free 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, $72.5 B in 2016 with an inter-annual growth rate of 22 % [37]. Third, online advertising represents an important source of jobs. For instance, recent studies have estimated that 0.9M (0.4 %) direct and 5.4M (2.5 %) indirect jobs were associated to online advertising in the EU-28 workforce in 2014 [48]. Fourth, online advertising represents the fundamental source of income of the companies at the forefront of technological innovations such as Google or Facebook [4, 19]. 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 [24, 57].

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 effectively fight invalid traffic.

Specialized companies referred to as verification vendors (e.g. IAS [26], DoubleVerify [17], Whiteops [56]) have emerged offering opaque proprietary solutions for the identification of invalid traffic. These vendors argue that opacity is needed to avoid providing valuable information to potential fraudsters, but previous research has shown that even simple attack vectors can defeat these opaque defenses [14, 34]. In addition, opacity prevents the possibility of independent auditing of these detection techniques. 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 [18, 51, 54] and the lack of transparency in the ecosystem [8, 15].

To meet the demands of advertisers, in this paper, we present the first open source and 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 negatively impact the business model of DSPs. 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 bid requests. Moreover, DSPs handle up to tens of billions of bid requests 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 auditable invalid ad traffic detection system that operates in real-time at the level of individual request. 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

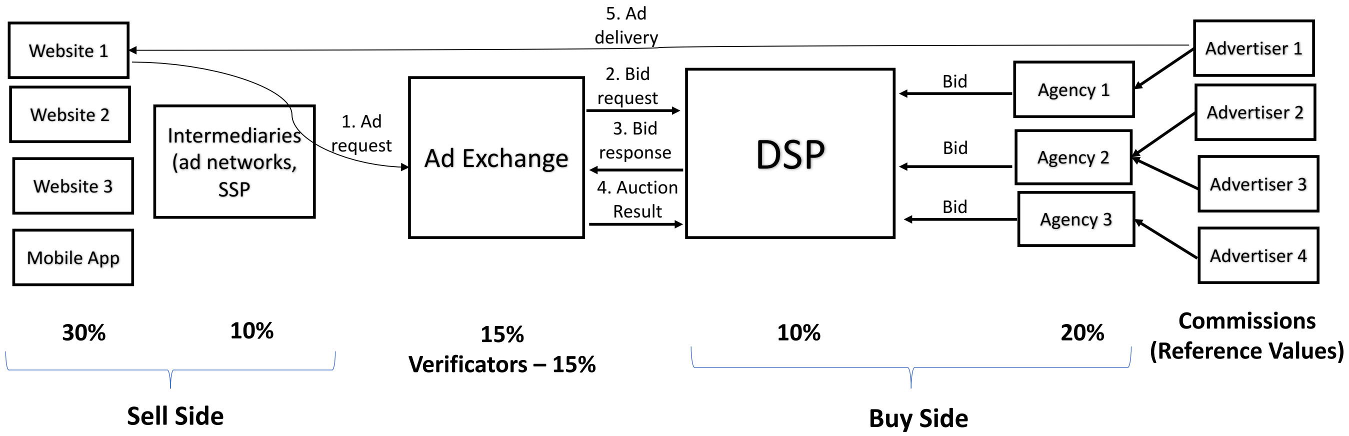

Figure 1: Overview of the programmatic advertising ecosystem operation.

been built in accord with the latest version of openRTB specification [31] and is able to handle up to 500 k bid requests per second, adding a total delay of 3 ms or less to each bid request. 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 for a period of 2 months observing that (in average) 20 % of the daily ad traffic can be safely considered invalid.

Nameles has been publicly released under an open source license. Its code is public and then auditable by anyone. In addition, while the current opaque approach taken by the Ad Tech industry has been shown flawed [14, 34], open-source software has been proven a key success factor in other related areas, e.g. Snort [43] for Network Intrusion Detection. These facts 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.

A user connects to a website[1] offering one (or more) ad space(s). Each ad space is typically leased by the website's publisher to an ad network, 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 referred to 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 referred to as the *bidding process*, which represents the *buy-side* of the programmatic process. The bidding processes is standardized by the openRTB protocol [31]. 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 [31]. 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 advertisers and if so, it creates a bid response including, among other information, the price the advertiser is willing to pay to show its ad in the website and to the user indicated by the bid request. Note that the bid responses to a given bid request have to be received by the Ad Exchange within 100 ms [21]. 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 obtaining the ad, which is immediately downloaded by the browser and shown to the user. This is represented by step 5 in Figure 1. An ad successfully delivered is referred to as an *ad impression.*

In the current programmatic ecosystem Ad Exchanges aggregate ad inventory (i.e., ad spaces) from up to tens of thousands of publishers, each DSP connects up to a 100 Ad Exchanges, receives up to tens of billions of bid requests per day and handles in the order of hundreds to thousands of advertisers.

From a business perspective, each bid represents an opportunity for the DSP to match demand on the buy-side with supply on the sell-side. In effect, 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

[1] Note that the described process is the same when considering a mobile app instead of a website.

transacted and an ad is displayed as a result. However, the advertiser benefits only when the traffic associated with the transaction is valid. This business model is open to fraudulent activity [24, 55, 57] (e.g., a publisher monetizing visits to a website coming from bots) whereas it seems to not offer the right incentives to intermediaries to identify and filter invalid ad traffic. Note that, according to various industry guidelines [20, 54], *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.* Various industry bodies and committees of established bodies have been created to define general guidelines to address the invalid traffic problem: JICWEBS, TAG, Botlab, and MRC's Invalid Traffic Committee [6, 11, 29, 52]. Moreover, verification companies (e.g., IAS, Double Verify or WhiteOps) have emerged recently offering proprietary opaque solutions for filtering invalid traffic. However, the lack of transparency on the used techniques makes tough to assess their actual capabilities. Indeed, recent works have demonstrated inefficiencies in existing solutions for the identification of invalid traffic [14, 34].

Despite these efforts, different studies attribute billions of dollars wasted in invalid traffic every year [24, 55, 57]. This economic damage, the uncertainty in the quality of media transactions and the lack of transparency in the programmatic ecosystem (including verification companies) have pushed advertisers to become increasingly vocal about the invalid ad traffic problem [8, 15, 18, 51, 54].

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 [35]. 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.

3.1 Qualitative Analysis

We assert that DSP companies are rarely profitable [5]. We investigated seven publicly listed DSP companies through their annual income statements and found that only one company had a positive net income [5]. Moreover, depending on the DSP company, variable costs ranged from 30 % to 50 % of the revenue [5]. Variable costs represent a proportion of total costs that vary as a function of revenue. These findings indicate that the current operation of major DSPs creates losses that have a strong correlation with variable costs.

The DSP company win-rate [15] 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 contributes exclusively to increase DSPs costs.

Moreover, 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 % [59] and an ad exchange *fill-rate* (i.e, the fraction of successfully completed auctions) in the range 10 % to 40 % [42]. Consequently, we assert that there is an oversupply of programmatic media impressions, which supports the economic viability of invalid traffic filtering. In other words, even after removing the invalid traffic there will be still enough ad inventory available for DSPs to consume.

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. In addition, filtering invalid traffic would reduce strategic risk associated with undisclosed exposure to ad fraud. In the case of two DSP companies [30, 49], each lost significant fraction of their market capitalization as a direct result of their exposure to invalid traffic becoming evident to investors.

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.

3.2 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 predicted return [7]. Finance theory endorses an investment if NPV is positive and higher than NPV of an alternative investment [7]. In addition to the NPV, we evaluated Enterprise Value (EV) [27], 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 [7]. NPV and EV are widely used as decision-making tools for planning purchases, mergers or acquisitions [7].

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 [22] and are the same for both scenarios.
2) Rate of return r. A typical r for investments made into new systems or products is 20 % [7, 28]. We use this value for both scenarios.

	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
Industry avg. ACME	3.163	5.468	-2.036	1.101	2.079	-1.267

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

3) Invalid traffic filtering rate F. We consider $F = 0$ for Scenario A and F ranging 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 low penalty is imposed for filtering rates up to the average reported fraction of invalid traffic from different studies [16, 58] as well as insights from the industry.
5) Long-term cash flow growth rate G, which is set-up to 2 % in both scenarios [33].

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 (in average) 1,72 and 1,89 times in comparison with Scenario A, respectively.

Finally, we would like to highlight that the results of NPV and EV 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 between December 01, 2016 and Jan 31, 2017. The data is from one of the largest DSPs with 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 (in 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 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 Applications as *Domains* along the paper.

5 SYSTEM REQUIREMENTS, DESIGN AND IMPLEMENTATION

In this section we describe *Nameles*, a 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. Moreover, Nameles is (to the best of the authors' knowledge) the first open source and auditable solution for the detection of invalid ad traffic, thus meeting the requirements of advertisers. We will first describe the fundamental operational requirements of the system and then provide details on its design and implementation.

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 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 100 ms [21]. Hence, the delay introduced should be limited to few ms 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 Confidence Score associated to a bid request indicating the likelihood that such bid request is invalid. Therefore, our system must incorporate an accurate scoring algorithm.

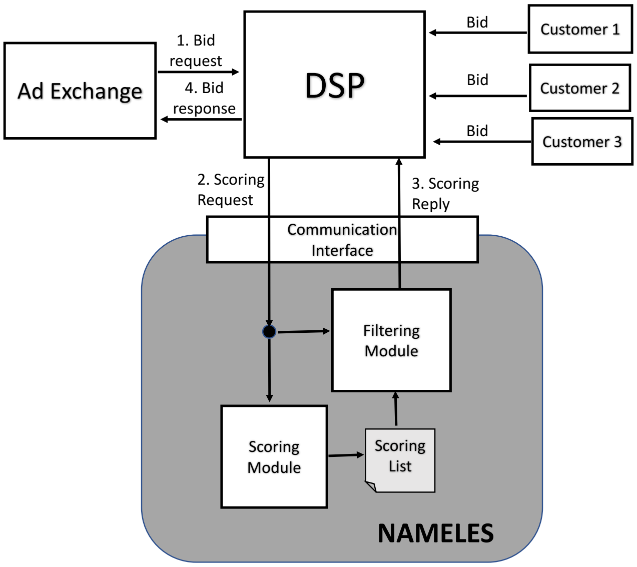

Figure 2: Programmatic Ecosystem Scheme with Nameles.

5.2 System Design

In this subsection, we first present a brief overview of the system functionality and how it is integrated within the programmatic advertising ecosystem and more specifically with the DSPs. Then, we describe in detail each of the functional blocks forming Nameles: the Communication Interface, the Scoring Module, and the Filtering Module.

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 [31]. 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. The traffic profile associated with a given domain may change significantly over time, resulting in a higher (or lower) confidence. To address this issue, the Scoring Module recomputes the Confidence Score of each domain every day. As a result of the described process, the Scoring Module produces every day a *Scoring List* that includes both the Confidence Score and the Confidence Class for each individual domain.

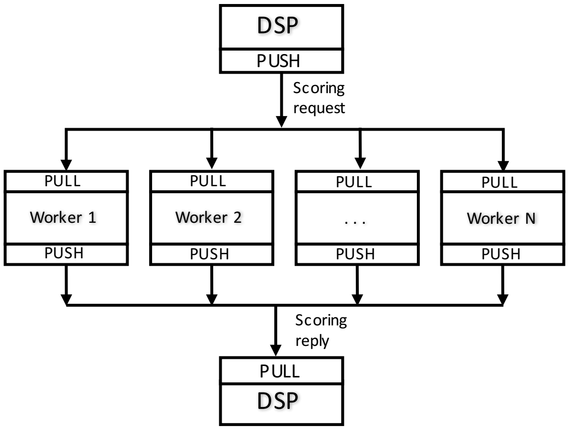

Figure 3: Parallel Pipeline communication architecture.

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. If the domain is not present in the Scoring List, the scoring reply includes NULL values for the Confidence Score and the Confidence Class.

Finally, the communication between the DSP and Nameles is handled by the *Communication Interface Module*.

5.2.2 Communication Interface Module. This module is responsible for handling the communication between the DSP and Nameles. Specifically, it 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 as depicted in Figure 3. 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 result of the filtering process is pushed by the workers to the receiving queue of the DSP.

The parallel pipeline communication structure offers a number of characteristics that make it a suitable solution in our case. First, it is easy to implement, thus requiring a low deployment effort for the DSPs using Nameles. Second, it offers outstanding scalability performance, being able to handle streams of hundreds of thousands of requests per second with processing delays below 3 ms (see Section 6.2). Third, it can be implemented using existing message handling solutions and middleware [2, 25, 47].

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 $d-1$, $d-2$, $d-3$, ...

To produce the *Scoring List*, the Scoring Module implements 3 different algorithms: one to compute the Confidence Score of each individual 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 [40] of the distribution of bid requests across IP addresses for each domain in the considered dataset. 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). We use the following expression to compute the Entropy ($H(X)$) for a domain X:

$$H(X) = log_2(C(X)) - \frac{\sum_{i=1}^{n} C(x_i)\, log_2(C(x_i))}{C(X)} \qquad (1)$$

where, $C(x_i)$ represents the number of bid requests received by the domain from IP$_i$, and $C(X)$ represents the total number of bid requests associated with the domain.

Shannon entropy has been successfully used in a wide range of applications [40] and specifically in the field of anomaly detection [32, 53].

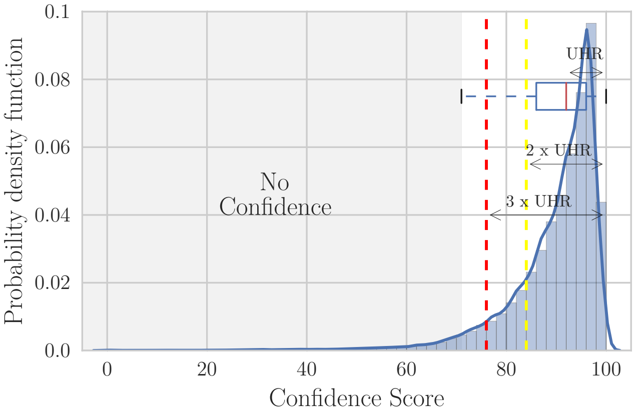

Figure 4: Distribution of Confidence Score (CS) values for domains with more than 500 bid requests at December 1, 2016.

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 avoids making a direct comparison of domains with different volumes of bid requests. 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 request associated to a domain. In essence, we compute the ratio of the entropy ($H(X)$) 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:

$$CS(X) = 100\left(1 - \frac{\sum_{i=1}^{n} C(x_i)\, log_2(C(x_i))}{C(X)\, log_2(C(X))}\right) \qquad (2)$$

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 address 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 4 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 indicates that most domains present homogeneous traffic patterns (represented by high CS) whereas as we move towards low values, fewer domains are found presenting increasingly deterministic patterns. In

[2] For instance, this can be the result of a domain receiving most of its visits from scrapers or from other types of bots associated with fraudulent ad traffic.

other words, 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 [38], 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 $max\,(CS) - 3\,\mathrm{UHR} > CS \geq 25\,\mathrm{percentile} - 1.5\,\mathrm{IQR}$.
- *Moderate Confidence Class*: formed by domains whose CS falls in the range $max\,(CS) - 2\,\mathrm{UHR} > CS \geq max\,(CS) - 3\,\mathrm{UHR}$.
- *High Confidence Class*: formed by domains whose CS falls in the range $CS \geq max\,(CS) - 2\,\mathrm{UHR}$.

Figure 4 shows the four defined Confidence Classes for the Confidence Score distribution of the December 1, 2016, dataset.

- Predicting the Scoring List: The Scoring list used at day d has to be inferred from a prediction algorithm applied on 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 our dataset. This analysis revealed that CS values present a 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 [39]. 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 optimal 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 received scoring request 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), the domain's CS and the domain's Confidence Class. The scoring reply is sent to the DSP through the Communication Interface module. The DSP can leverage this information to define its own invalid traffic filtering policy. 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. For doing this, we used resources with negligible cost in comparison to typical resources available for DSPs and relying on open-source technology.

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 Figure 3 on top of ZeroMQ [2] (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 id, 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 temporal 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 temporal 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

[3] The UHR is measured as the distance between the median and the maximum value of the CS distribution.

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. Finally, the table computed with the data at day d serves as scoring list for day $d + 1$.

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. Specifically, the scalability and delay requirements, and accuracy pertaining to the scoring of domains.

6.1 Experimental Set-up

To conduct the performance evaluation, we have deployed an experimental set-up that replicates a production set-up in actual business use 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, 48 GB of RAM and 6 TB of hard drive capacity with a non-recurring-cost (NRC) of ∼$6k. The servers implementing the "Communication and Filtering" and the "Scoring" modules are similar, a Dell PowerEdge R730xd with 24-cores, 64 GB of RAM and 46 TB hard drive capacity with a NRC of ∼$13 k. Each server is connected to a common 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 the 62 daily datasets, including between 1.7-1.9 B bid requests, and confirmed that the computation time of the Scoring List 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.

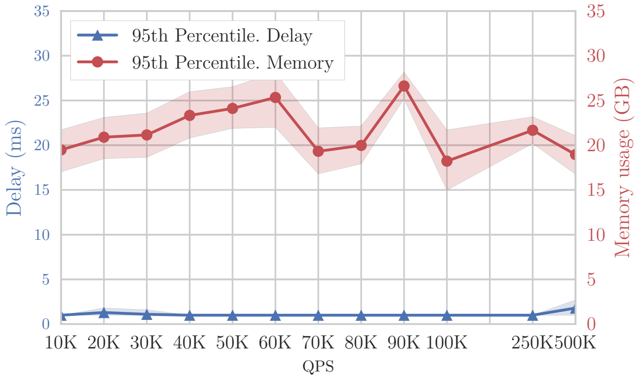

Figure 5: 95 percentile of delay and memory consumption for the filtering process at different input request rates.

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 5 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 5 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 are 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.

$CS_d(X) \setminus CS^*_d(X)$	No C.	Low C.	Mod. C.	High C.
No Confidence		0.65 %	0.28 %	0.07 %
Low Confidence	0.57 %		1.13 %	0.10 %
Moderate Confidence	0.30 %	1.06 %		3.02 %
High Confidence	0.06 %	0.10 %	2.93 %	

Table 2: Average miss-classification rates among the Confidence Classes for the 62 daily samples in the dataset.

6.3 Scoring Accuracy

In order to measure the accuracy of our scoring method, we first assess the accuracy of our prediction algorithm and then the accuracy of the Confidence Scores assigned to the domains.

6.3.1 Accuracy of prediction algorithm. For each of the daily datasets, 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 miss-classification rate of domains among Confidence Classes. Table 2 presents a summary of the average miss-classification rate between each pair of Confidence Classes across the 62 days in our dataset. First of all, we observe that miss-classification rates are below 3.02 % between any pair of classes. A careful analysis of the miss-classified 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 mis-classifications 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 with the invalid traffic problem; attack vectors, domain traffic profiles, and others. As a result, there are no reliable ground truth datasets available for evaluating invalid traffic filtering solutions. However, contrary to propriety verification solutions that suffer from this same issue, Nameles source code can be independently audited. To validate the accuracy of the Confidence Scoring, we performed an assessment using a twofold approach. First, we conducted an analysis that relies on the following metrics, which are extensively used in the Adtech industry to infer the quality of traffic of a domain:

- *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.

- *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 reach the domain directly without being redirected from other website. 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.

We have queried two well-known services, Alexa [3] and SimilarWeb [41], to obtain these metrics for those domains in our dataset with more than 500 associated bid requests. Note that not all the metrics are offered by both services. Table 3 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, suggesting that our scoring mechanism is able to accurately identify legitimate domains.

Secondly, we have worked closely together with experts from the Ad Tech industry over a period of 18 months to improve, as well as to subjectively evaluate, the results provided by Nameles in extensive trials. The satisfactory results obtained during these tests have led the World Federation of Advertisers as well as renowned Ad Fraud research consultants, unnamed to preserve author's anonymity, to endorse Nameles.

We conclude that both the objective analysis based on proxy metrics pertaining the confidence level of a domain, as well as the evaluation conducted by individual experts, suggest that the accuracy of Nameles' scoring system is suitable for adoption by DSPs.

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 use the corresponding fractions of traffic associated with each Confidence Class as filtering rate input to the economic model described in Section 3.2 in order to quantify the positive impact that Nameles may have in the profitability of DSPs.

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

Table 3: Value of external quality metrics associated with domains in each of the defined Confidence Classes in our dataset.

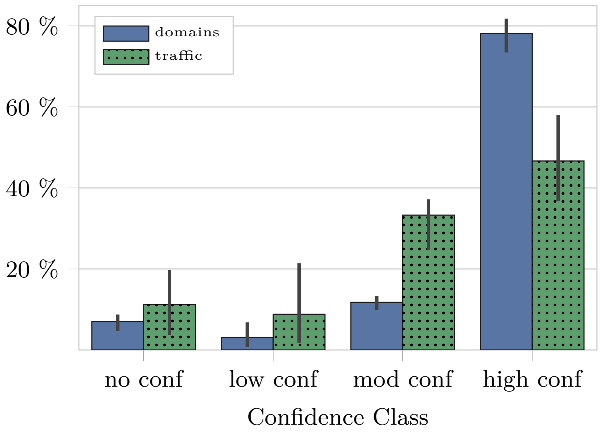

Figure 6: Percentage of domains and ad traffic in each of the Confidence Classes across the 62 days of the dataset. The main bar presents the average value whereas the error bars show the minimum and maximum values.

7.1 Longitudinal Analysis of domains' confidence level

Figure 6 shows the fraction of domains and ad traffic (i.e., bid requests) belonging to each of the defined Confidence Classes for the 62 days in our dataset. The main bar shows the average fraction and the error bar shows the maximum and minimum values across the days in the sample. Note that these results are obtained for domains with at least 500 bid requests in a day in order to guarantee that we have statistically meaningful information about the traffic pattern of the domain. In average (11.21; 8.83; 33.30; 46.66) % of the traffic is associated with ("No", "Low", "Moderate" and "High") Confidence Classes. For instance, a DSP handling 50 B bid request per day using a policy that filters traffic belonging to "No" and "Low" classes would eliminate (in average) around 10 B (20 %) bid requests every day.

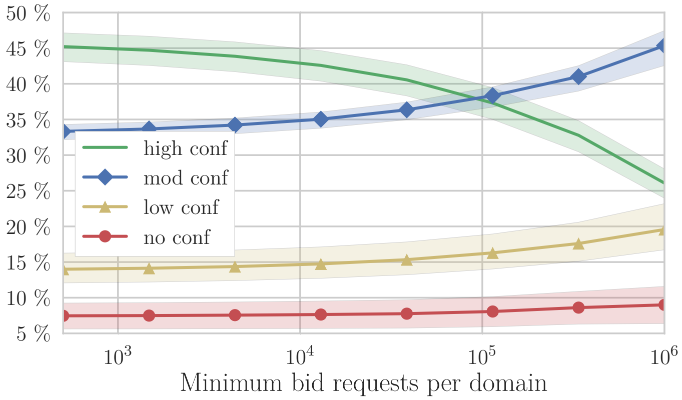

Figure 7: Percentage of ad traffic in each Confidence Class as function of the popularity (i.e., bid requests) of domains.

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. Figure 7 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 case of "Low" and "No" Confidence Classes we observe just a light increase after 100 k daily bid requests.

7.2 Nameles' impact on DSPs' profitability

The results in the previous subsection provide specific figures on the filtering rates that Nameles provides at different confidence level. For instance, 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 a high confidence.

Using these filtering rates as input to the economic model presented in Section 3.2 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 strong positive economic impact.

As highlighted in Section 3.2, these results apply in the scenario defined by the considered realistic assumptions.

8 NAMELES' APPLICABILITY, EXTENSIBILITY AND RELEASE

In this section, we first elaborate on the applicability of Nameles by other players of programmatic advertising further than DSPs. Second, we discuss how Nameles can be easily extended to assess the traffic patterns of IP address as well as to integrate new detection mechanisms, which will enhance its capacity to filter invalid traffic. Finally, we present our reasons to release Nameles as an open source solution.

8.1 Applicability

In this paper, we have presented a prototype of Nameles ready to be integrated with DSPs. We have focused in DSPs since our economic analysis in Section 3 provides evidences that they are negatively affected by invalid ad traffic. However, 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. Moreover advertisers and legit publishers can indirectly use Nameles. For instance, if DSPs provide information about the CS of domains to their advertisers, they can define their advertising buy plan using such info. Publishers can use their CS as a reference to self-evaluate the quality of ad traffic in their domains and identify potential problems (e.g., the unconscious use of an illegitimate source of traffic).

8.2 Extensibility

8.2.1 Signals' extensibility.

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 and/or the domain. 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 Classes). Adding the information about the CS of IP addresses increases the fraction of traffic in these two categories in less than 0.5 %. Note that to compute this we have also considered invalid all ad traffic coming from IPs located in data-centers[4] as recommended by industry guidelines [12]. This result indicates that the application of Nameles at the level of domains suffices to identify more than 99 % of low quality ad traffic. Then, due to the substantial computational overhead associated with obtaining the CS for millions of IPs every day, DSPs may find more efficient to focus exclusively on monitoring domains.

8.2.2 Integrating complementary detection techniques.

Key advantages of Nameles are its modularity and simplicity, which allow easy extension, modification, and improvement of the platform. For instance, the current implementation of Nameles uses the normalized entropy as the information for identifying invalid traffic, and we acknowledge that this technique is not able to identify all types of invalid traffic (see Section 9). However, the Scoring module can be extended to include other detection techniques (e.g., Co-Visitation network [45]) to improve the efficiency of the platform. Indeed, since its inception, Nameles was designed with the goal to serve as a platform for the community-led industry-wide effort to fight invalid traffic in programmatic advertising for reasons discussed in next subsection.

8.3 Release as Open Source

Nameles is, to the best of the author's knowledge, the first available open source solution for the identification of invalid ad traffic. We summarize next the reasons that lead us to release it as open source: 1) It meets the demands of advertisers that, led by the WFA, are claiming transparent and auditable solutions; 2) Recent studies show that even simple attacks can defeat existent opaque proprietary solutions. Instead, open source products, Snort [43] or Bro [50], have been proven efficient in related areas such as Network Intrusion Detection; 3) The identification of invalid traffic is a very complex problem that needs to be addressed in a community effort. This common effort needs to build upon initial open source solutions such as Nameles.

9 NAMELES' LIMITATIONS

The identification of invalid traffic is a very complex problem, so neither Nameles' nor any other technique is a *one-size-fits-all* solution to end the problem. In the rest of the section we discuss the main limitations of Nameles and argue why despite of them, Nameles is still an important contribution. As in any detection system, Nameles' potential limitations are associated to false negatives and false positives.

- **False Negatives:** They are represented by the invalid traffic not identified by Nameles. In the presence of Nameles, any

[4] IPs have been mapped to datacenters using the list from [9].

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. Note that different DSPs can configure different thresholds, making this inference exercise more difficult. 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 in order to make its CS overpass the quality threshold defined by the different DSPs. Both approaches lead to a reduction in the obtained revenue.

To obtain a ball park estimation of such reduction, we have computed the number of daily visits that all 8K 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 in average these domains would need to eliminate 38.5 % visits leading to a roughly similar reduction in their revenue.

Therefore, despite being subject to invalid traffic attacks, Nameles contributes to significantly reduce the profitability associated to them.

-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 3.1 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.

10 RELATED WORK

In the recent years several studies have unveiled different types of attacks used for generating invalid traffic with the goal of generating monetary gain fraudulently [36, 46, 55], with reported revenues of up to millions of dollars per day [57]. To address the problem of invalid ad traffic, verification vendors such as Integral Ads Science [26], Double Verify [17], and WhiteOps [56] have emerged in recent years. Also major players of the Adtech industry claim to devote significant attention to address this issue [1]. Unfortunately, all existing commercial solutions are based on opaque proprietary technologies, and it is hard to assess their efficiency in identifying invalid traffic. Some recent studies have proven the inefficiencies of such solutions in identifying even simple invalid traffic attacks [14, 34].

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 [45] or delivered ads [9, 23]. 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 [9] or redirection attacks [44]. None of these solutions are valid for DSPs. To the best of the author's knowledge, Stitelmant et. al [13] proposed the only alternative solution to Nameles able to operate at DSP level. By analyzing the degree of overlapping in the IPs visiting two (or more) domains, their solution identifies potential invalid traffic. Note that this is a complementary traffic-based detection technique to our normalized entropy score and thus it can be incorporated to Nameles to improve its detection capability.

From a methodological perspective, there is a previous work that has used entropy to identify invalid video visits to a Chinese video portal [10]. 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, Namless clearly advance the state-of-the-art from a methodological perspective as well.

11 CONCLUSION

This paper introduces Nameless, 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 evidences that, contrary to the 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 a good detection accuracy. Moreover, applying Nameles on a 64 days dataset including almost 2 B bid requests per day, we observe the presence of 20 % invalid traffic.

The evidenced performance of the current version 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] 2017. AdSense Help. How Google prevents invalid activity. (2017). Retrieved May 19, 2017 from https://support.google.com/adsense/answer/1348752?hl=en-GB&ref_topic=1348566

[2] Faruk Akgul. 2013. *ZeroMQ*. Packt Publishing.

[3] Alexa: actionable analytics for the web 2017. (2017). Retrieved May 19, 2017 from http://www.alexa.com

[4] Inc. Alphabet. 2017. 2016 Annual report. (02 Feb. 2017). Retrieved May 19, 2017 from https://abc.xyz/investor/pdf/2016_google_annual_report.pdf

[5] An adtech autopsy 2017. (2017). Retrieved 2017-01-27 from http://autopsy.pw

[6] Botlab 2017. (2017). Retrieved May 19, 2017 from http://botlab.io

[7] Richard A Brealey, Stewart C Myers, Franklin Allen, and Pitabas Mohanty. 2012. *Principles of corporate finance.* Tata McGraw-Hill Education.

[8] Internet Advertising Bureau. 2017. *Transparency is the key to programmatic success.* Technical Report. Retrieved May 19, 2017 from https://www.iab.com/wp-content/uploads/2015/08/IABDigitalSimplifiedProgrammaticAdvertisingTransparency.pdf

[9] 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, 120–126.

[10] 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.

[11] Media Rating Council. 2014. (2014). Retrieved May 19, 2017 from http://mediaratingcouncil.org

[12] Media Rating Council. 2015. Invalid Traffic Detection and Filtration Guidelines Addendum. (15 Oct. 2015). Retrieved May 19, 2017 from http://mediaratingcouncil.org/101515_IVT%20Addendum%20FINAL%20(Version%201.0).pdf

[13] 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.

[14] Shailin Dhar. 2016. *Mystery Shopping Inside the Ad Fraud Verification Bubble.* Technical Report. Retrieved May 19, 2017 from http://www.slideshare.net/ShailinDhar/mystery-shopping-inside-the-adverification-bubble

[15] Digiday. 2016. WTF is programmatic? (July 2016). Retrieved May 19, 2017 from http://digiday.com/wp-content/uploads/2016/07/WTF_programmatic-2016.pdf

[16] Distil. 2015. Distil Networks Releases New Data on The State of Digital Advertising Fraud. (22 Oct. 2015). Retrieved May 19, 2017 from https://resources.distilnetworks.com/press-releases/distil-networks-releases-new-data-on-the-state-of-digital-advertising-fraud

[17] Double Verify 2017. (2017). Retrieved May 19, 2017 from http://www.doubleverify.com

[18] Emarketer. 2016. The Ad Industry's Focus on Fraud Has Intensified. (09 Feb. 2016). Retrieved May 19, 2017 from https://www.emarketer.com/Article/Ad-Industrys-Focus-on-Fraud-Has-Intensified/1014430

[19] Inc. Facebook. 2017. 2017 Q1 Quarterly report. (04 May 2017). Retrieved May 19, 2017 from http://d18rn0p25nwr6d.cloudfront.net/CIK-0001326801/2309fab3-1f67-46bc-8d66-7039bc4a0e68.pdf

[20] Joint Industry Committee for Web Standards. 2015. *Traffic Fraud: Best Practices for Reducing Risk to Exposure.* Technical Report. Retrieved May 19, 2017 from http://www.jicwebs.org/images/JICWEBS_Anti-Fraud_Best_Practices_June_2015.pdf

[21] Google. 2017. DoubleClick RTB Protocol. Latency Restrictions and Peering. (11 Jan. 2017). Retrieved May 19, 2017 from https://developers.google.com/ad-exchange/rtb/peer-guide

[22] Gurufocus 2017. (2017). Retrieved May 19, 2017 from http://www.gurufocus.com

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

[24] Lucy Handley. 2017. Businesses could lose $16.4 billion to online advertising fraud in 2017: Report. (15 March 2017). Retrieved May 19, 2017 from http://www.cnbc.com/2017/03/15/businesses-could-lose-164-billion-to-online-advert-fraud-in-2017.html

[25] Michi Henning and Mark Spruiell. 2003. Distributed programming with ice. *ZeroC Inc. Revision* 3 (2003), 97.

[26] Integral Ad Science (IAS) 2017. (2017). Retrieved May 19, 2017 from https://integralads.com

[27] Investopedia. 2017. DCF Analysis: Coming Up With A Fair Value. (2017). Retrieved May 19, 2017 from http://www.investopedia.com/university/dcf/dcf4.asp

[28] Investopedia. 2017. Rate of Return. (2017). www.investopedia.com/terms/r/rateofreturn.asp

[29] Joint Industry Committee for Web Standards (JICWEBS) 2016. (2016). Retrieved May 19, 2017 from https://jicwebs.org

[30] Schubert Jonckheer & Kolbe. 2016. Rocket Fuel Executives Under Investigation. (16 Feb. 2016). Retrieved May 19, 2017 from http://www.classactionlawyers.com/blog/2016/2/16/rocket-fuel-executives-under-investigation

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

[32] 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.

[33] Investopedia. J.B. Maverick. 2015. What is the long-term average growth rate of the telecommunications sector? (15 July 2015). Retrieved May 19, 2017 from http://www.investopedia.com/ask/answers/071515/what-longterm-average-growth-rate-telecommunications-sector.asp

[34] 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).

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

[36] Paul Pearce, Vacha Dave, Chris Grier, Kirill Levchenko, Saikat Guha, Damon McCoy, Vern Paxson, Stefan Savage, and Geoffrey M Voelker. 2014. Characterizing large-scale click fraud in zeroaccess. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 141–152.

[37] PwC. 2017. *IAB internet advertising revenue report 2016 full year results.* Technical Report. Retrieved May 19, 2017 from https://www.iab.com/wp-content/uploads/2016/04/IAB_Internet_Advertising_Revenue_Report_FY_2016.pdf

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

[39] 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.

[40] C. E. Shannon. 1948. A mathematical theory of communication. *The Bell System Technical Journal* 27, 3 (July 1948), 379–423. https://doi.org/10.1002/j.1538-7305.1948.tb01338.x

[41] SimilarWeb: Website Traffic & Mobile App Analytics 2017. (2017). Retrieved May 19, 2017 from https://www.similarweb.com

[42] Smaato. 2017. Mobile RTB Insights Report Q3 2014. (2017). Retrieved May 19, 2017 from https://www.smaato.com/resources/reports/mobile-rtb-insights-q3-2014

[43] Snort - Network Intrusion Detection & Prevention System 2017. (2017). Retrieved May 19, 2017 from https://www.snort.org

[44] 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. https://www.usenix.org/conference/usenixsecurity13/technical-sessions/paper/springborn

[45] 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.

[46] Brett Stone-Gross, Ryan Stevens, Apostolis Zarras, Richard Kemmerer, Chris Kruegel, and Giovanni Vigna. 2011. Understanding Fraudulent Activities in Online Ad Exchanges. In *Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference (IMC '11)*. ACM, 279–294. https://doi.org/10.1145/2068816.2068843

[47] Storm, Apache 2015. (2015). Retrieved May 19, 2017 from http://storm.apache.org

[48] IHS Technology. 2015. *Paving the way: how online advertising enables the digital economy of the future.* Technical Report. https://www.iabeurope.eu/files/9614/4844/3542/IAB_IHS_Euro_Ad_Macro_FINALpdf.pdf

[49] The Telegraph. 2015. Matomy shares battered in digital ad fraud crackdown. (23 Sept. 2015). Retrieved May 19,

2017 from http://www.telegraph.co.uk/finance/newsbysector/mediatechnologyandtelecoms/digital-media/11558623/Matomy-shares-battered-in-digital-ad-fraud-crackdown.html

[50] The Bro Network Security Monitor 2014. (2014). Retrieved May 19, 2017 from https://www.bro.org

[51] Financial Times. 2017. Google charges for YouTube ads even when viewed by robots. (Sept. 2017). Retrieved May 19, 2017 from https://www.ft.com/content/f9da727c-6207-11e5-9846-de406ccb37f2

[52] Trustworhy Accountability Group (TAG) 2016. (2016). Retrieved May 19, 2017 from https://tagtoday.net

[53] 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.

[54] WFA. 2014. WFA guide to Programmatic Media. (2014). Retrieved May 19, 2017 from https://www.wfanet.org/app/uploads/2017/04/programmatic.pdf

[55] WhiteOps. 2016. *The Methbot Operation*. Technical Report.

[56] Whiteops 2017. (2017). Retrieved May 19, 2017 from https://www.whiteops.com

[57] ANA & WhiteOps. 2016. *2015 Bot Baseline: Fraud in Digital Advertising*. Technical Report.

[58] IMPERVA INCAPSULA. Igal Zeifman. 2017. Bot Traffic Report 2016. (24 Jan. 2017). Retrieved May 19, 2017 from https://www.incapsula.com/blog/bot-traffic-report-2016.html

[59] 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.