Traffic Anomaly Detection Based on the IP Size Distribution

Fabio Soldo
Google Inc.
fsoldo@google.com

Ahmed Metwally
Google Inc.
metwally@google.com

***Abstract*—In this paper we present a data-driven framework for detecting machine-generated traffic based on the IP size, *i.e.*, the number of users sharing the same source IP. Our main observation is that diverse machine-generated traffic attacks share a common characteristic: they induce an anomalous deviation from the expected IP size distribution. We develop a principled framework that automatically detects and classifies these deviations using statistical tests and ensemble learning. We evaluate our approach on a massive dataset collected at Google for 90 consecutive days. We argue that our approach combines desirable characteristics: it can accurately detect fraudulent machine-generated traffic; it is based on a fundamental characteristic of these attacks and is thus robust (*e.g.*, to DHCP re-assignment) and hard to evade; it has low complexity and is easy to parallelize, making it suitable for large-scale detection; and finally, it does not entail profiling users, but leverages only aggregate statistics of network traffic.**

I. Introduction

Today, a large number of Internet services such as web search, web mail, maps, and other web-based applications are provided to the public free of charge. At the same time, designing, deploying, and maintaining these services is expensive. They must have high availability, be able to serve any user, anonymous or logged in, and from anywhere in the world. This is often possible due to the revenue generated by Internet advertising, an industry that in 2009 generated over $22 billion [1], [2] in the U.S. alone.

For the above-mentioned reasons, fraud detection is a critical component for the well-being of many Internet services. Hit inflation attacks refer to the fraudulent activities of generating charges for online advertisers without a real interest in the products advertised. They can be classified into publishers' and advertisers' attacks. Publishers' hit inflation attacks use fraudulent traffic in an attempt to increase publishers' revenues from online advertising. Advertisers' hit inflation attacks aim at increasing the overall amount of activities, such as impressions or clicks associated with the advertisements of their competitors. The main objective of advertisers' hit inflation attacks is depleting their competitors advertising budgets. In this paper, we focus on publishers' attacks, but the same discussion applies to advertisers' attacks.

Hit inflation attacks can be performed in many ways, using different network infrastructures and levels of sophistication. Fig. 1 depicts a simple scenario with three publishers, where each publisher represents a different type of traffic. Advertisements on the publisher sites *thispagemakesmoney.com* and *thispagetoo.com* receive legitimate traffic, *i.e.*, users interested in the advertisements clicked on them. Advertisements on *thispagetoo.com* also receive fraudulent traffic. For instance, the publisher might ask her friends to repeatedly click on advertisements displayed on her site. Finally, in a more sophisticated hit inflation attack, publisher *iwontmakemoney.com* uses a botnet to automatically generate a large amount of fraudulent traffic. This simple example illustrates the complexity of the problem.

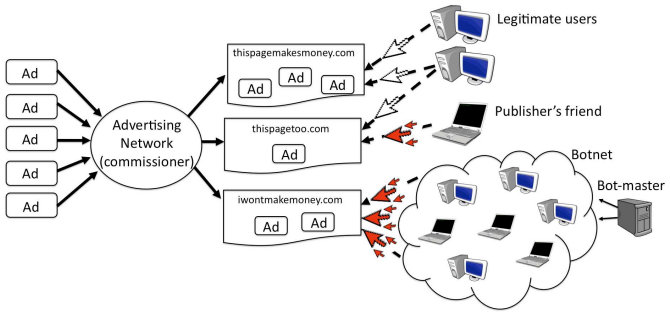


Fig. 1. Three publishers contract with an advertising network to host advertisements for a commission, for each click on these advertisements. The three publishers illustrate three types of traffic: (1) advertisements on the publisher site *thispagemakesmoney.com* are clicked only by legitimate users (white pointers); (2) advertisements on *thispagetoo.com* are clicked by both legitimate and fraudulent users (red pointers); and (3) advertisements on *iwontmakemoney.com* are not clicked by legitimate users–instead, *iwontmakemoney.com* uses a large botnet to generate fraudulent traffic.

Hit inflation attacks represent the biggest threat to the Internet advertising industry [3], [4], [5]. In this paper, we share our experience in building a fraud detection system at Google. Hit inflation attacks represent a specific application of the techniques and methodologies presented here. However, these can be applied, generally, to detect machine-generated traffic. The main contributions of this work are as follows:

- For the first time, the *IP size*, defined as the number of users sharing the source IP, is used as a discriminative feature for detecting machine-generated traffic. Our key observation is that several types of attacks induce an anomalous deviation from the expected publishers' IP size distribution.

Fabio Soldo was partly at an internship with the Traffic Quality Team at Google Inc. and partly at UC Irvine while this work was conducted. The work was partially supported by the NSF CyberTrust grant 0831530.

- We design a principled framework that estimates the expected IP size distribution, based on historical data, and domain-specific insights, and detects anomalous traffic using statistical learning techniques.
- We evaluate our approach on a massive data set of click logs collected at Google for over 90 consecutive days. This allows us to validate our implementation on a rich data set comprising diverse types of click traffic and machine-generated attacks.

Our approach combines several desirable characteristics: it successfully detects fraudulent traffic; it has low complexity and is easy to parallelize, making it suitable for large-scale detection; it is based on a fundamental characteristic of machine-generated traffic, and is thus robust (*e.g.*, to DHCP re-assignment) and hard to evade; and finally, it does not entail profiling users individually, but leverages only aggregate statistics.

The remainder of this paper is organized as follows. In Section II, we define the IP size and describe how attacks using machine-generated traffic affect the IP size distribution. In Section III, we describe the data set used in this study. In Section IV, we summarize the notation used throughout this paper. In Section V, we show how to distinguish a publisher's legitimate traffic from fraudulent traffic. In Section VI, we show how to detect fraud at the publisher's level. In Section VIII, we discuss the strengths and limitations of this work. In Section IX, we conclude the paper.

II. IP Size and Machine-Generated Attacks

A. IP Size

We define the *IP size* as the number of users sharing the same IP address. Estimating the IP size is a challenging problem in its own. Several users might share the same host machine, or might connect through the same Network Address Translation (NAT) device or even a cascade of NATs, as illustrated in Fig. 2. Moreover, the IP size changes over time as new users join the local network and share the same public IP and others leave, or as the IP address gets reassigned to a different host.

In this paper, we use the IP size estimation provided by the Google IP Size system [6]. In [6], application-level logs of trusted users, including search queries and advertisement clicks, are aggregated at the IP level. This data is used to build a probabilistic model of users activities. Then, the IP size is estimated as a function of both the rate of activities observed and the diversity of the observed traffic.

B. Observed IP Size Distributions

For each publisher, and a given time period T, we measure its IP size distribution. This is defined as the empirical distribution of the sizes associated with advertisements on her website during time period T.

Different publishers naturally exhibit different IP size distributions. Fig. 3 shows two examples of IP size distributions that are typically seen on (1) a website that receives average desktop traffic, and (2) a website that receives average mobile traffic. First, where a website receives mainly desktop traffic, most of the clicks have small sizes because, typically, only a handful of users share the same IP address. As such, the IP size distribution is highly skewed toward the left. Second, where a website receives mainly mobile traffic, the IP size distribution exhibits two distinct modes. This is because mobile users typically access the Internet either with public IP addresses, which have relatively small sizes, or through large proxies, which are shared by numerous users. Generally, different publishers have different IP size distributions depending on both the type of their services, and the type of traffic driven to their websites.

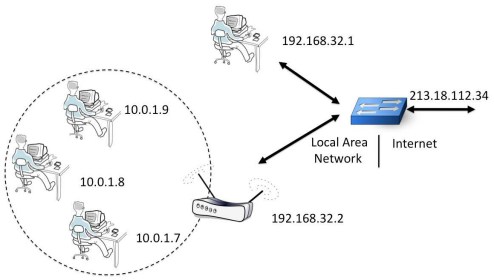


Fig. 2. Public IP address 213.18.112.34 is shared by 4 users. Thus, the IP size of 213.18.112.34 is 4. Intuitively, this means that we expect this specific IP address to roughly generate 4 times the number of activities generated by a single user.

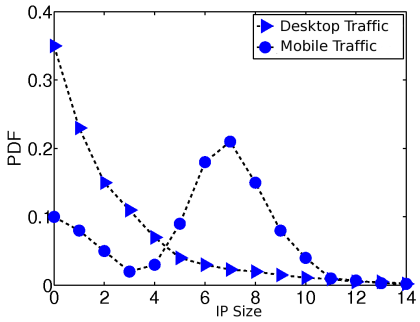


Fig. 3. Two example of publishers with two different IP size distributions.

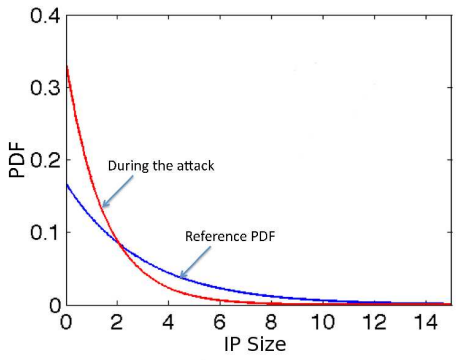
C. IP size Distributions

machine-generated attacks are performed in various ways, depending on the resources available, motivations and skills of the attackers. For instance, if an attacker controls a large number of hosts through a botnet, the attack can be highly distributed across the available hosts to maximize the overall amount of traffic generated while maintaining a low activity profile for each host individually. We refer to this type of attacks as *botnet-based* attacks. Conversely, if an attacker controls a few hosts but still wants to generate a large amount of traffic, she can use anonymizing proxies, such as TOR nodes, to hide the actual source IPs involved. We refer to this type of attacks as *proxy-based* attacks. Botnet- and proxy-based attacks are two diverse examples in the wide spectrum

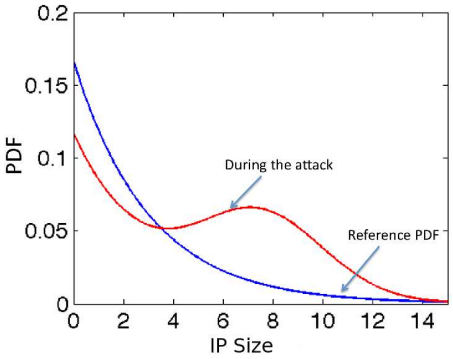
of possible attacks using machine-generated traffic, in terms of both the resources required and level of sophistication.

Fig. 4 illustrates these two attacks and how they affect the IP size distribution associated with a publisher. Assume that we have an a-priori knowledge of the expected IP size distribution based on historical data. Let the blue curve be the expected distribution of IP sizes. Fig. 4(a) depicts an example of botnet-based attack. Bots are typically end-user machines and so have a relative small IP size. Intuitively, this is because end-user machines are easier to compromise than large well-maintained proxies. As a result, a botnet-based attack generates a higher than expected number of clicks with small size. Analogously, a proxy-based attack skews the IP size distribution towards large IP sizes because a higher than expected number of clicks comes from large proxies, as in Fig. 4(b).

Despite their differences, most attacks share a common characteristic: they induce an unexpected deviation of the IP size distribution. The attacks in Fig. 4 represent two opposite scenarios. However, in both cases the attack is revealed as a deviation from the expected IP size distribution. In general, different deviations represent different signatures of attacks.



(a) botnet-based attack



(b) proxy-based attack

Fig. 4. Types of attacks and their effect on the IP size distribution: The blue curve represents the expected IP size distribution. The red curve represents the IP size distribution during the attack. Fig. (a) illustrates a botnet-based attack: clicks are generated by a large number of bots. These are typically end-user machines and thus skew the distribution toward small IP sizes. Fig. (b) illustrates a proxy-based attack: the IP addresses generating the clicks are rerouted through anonymizing proxies (*e.g.*, TOR nodes). Since many users share these proxies, this attack skews the IP size distribution toward large IP sizes.

III. The Data Set

A. Key Features

In this paper, we use advertisement click logs collected at Google from a sample of hundreds of thousands of different publisher websites. We use these logs to gain insights into modern machine-generated traffic attacks, as well as to test and evaluate the performance of our system on real data. In this section, we briefly describe the data set and the specific features used in this study.

We analyze a sample of click logs collected for a period of 90 consecutive days. Our analysis and development rely on the following fields in each entry:

- Source IP: the source IP address that generated the click.
- Publisher ID: the unique identifier associated with each publisher.
- Time: the timestamp associated with the click.
- Fraudulent click flag: a binary flag which indicates whether or not the click was labeled as fraudulent by any of the fraud detection systems already in place at Google.

In addition to the click logs, we also used two Google internal databases:

- *IP Size* database, which keeps tracks of the IP size [6]. We use the sizes estimated from the click traffic to filter fraudulent clicks. These were called click sizes of the IPs in [6].
- *Geographical IP* database, which provides up-to-date geographical information on source IP addresses.

B. Assessing the Quality of Traffic

In this paper, we leverage an internal classifier that takes as input click logs of network traffic and determines the likelihood that the network traffic is fraudulent machine-generated traffic. We call the score obtained through this system the *quality score*. This classification system takes as input a variety of features that accounts for different types of user inputs, and different types of anomalies. This classifier provides us with an estimate on the aggregate quality of a large set of clicks. Similar classifiers exist for other kinds of attacks depending on the application. For instance, in the case of email spam a classifier can be built on several features of the email. One of the features could be the ratio of users that labeled this email as spam. Another feature could be the number of valid and invalid recipient addresses, and so on.

We also define the *fraud score* as a function of the ratio between the number of fraudulent clicks and the total number of clicks, with different weights assigned to the fraudulent clicks depending on the reason for tagging them as fraudulent.

Finally, we also use two sets of blacklists, the Gmail Blacklist [7] and the Spamhaus Exploit Blacklist (XBL) [8], to determine whether or not the IP addresses that generate fraudulent ad events are also known to generate other types of abusive traffic. Gmail blacklist is a list of source IPs that are likely to send email spam. Spamhaus XBL is a realtime database of hosts infected by some exploits.

IV. Notation

Each click, c, is associated with a source IP address, IP_c, that generated the click, and with a publisher site, P_k, that hosted the advertisement clicked. Let S_c be the IP size associated with IP_c, and let n be the number of clicks on advertisements hosted by P_k in a certain time period, T.

Let us first consider a single publisher, P_k. We model the IP sizes, $S_1, ..., S_n$, as the realizations of a sample of n i.i.d. random variables, $\mathcal{S}_1, ..., \mathcal{S}_n$, that can take a finite number of positive values $B_1 < B_2 < ... < B_m$, where $B_1 = 1$ is the minimum number of legitimate users sharing an IP, and B_m is the largest IP size observed in the training period. The probability distribution associated with $\{\mathcal{S}_c\}$, is defined by some (unknown) values $p_1, ..., p_m$, where $p_i = \mathbb{P}[\mathcal{S}_c = B_i]\ \forall c$. In general, when dealing with multiple publishers these values depend on the publisher itself, *i.e.*, $p_i = p_i(P_k)$.

Let $\tilde{f}_i$ be the observed frequency of IP sizes associated with B_i, *i.e.*, the count of clicks that have size B_i: $\tilde{f}_i = \#\{S_c : S_c = B_i\}$, and f_i be the relative number of clicks of size B_i, *i.e.*, $f_i = \tilde{f}_i/n$. As the number of observed clicks increases, f_i approaches p_i as quantified by the Central Limit Theorem, $\frac{f_i - p_i}{\frac{p_i}{\sqrt{n}}} \rightarrow_{n\rightarrow\infty} \mathcal{N}(0, 1-p_i)$. This allows us to approximate the unknown value p_i with measurable quantities, f_i, and derive formal confidence bounds. Finally, assume that we have an estimate of the true (unknown) probability distribution: $p_i = r_i,\ \forall i$. We will describe how to estimate the f_i and r_i variables and use these values to detect frauds.

V. Click Filtering

In this section, we focus on the general scenario where the click traffic received by a publisher is a mixture of both legitimate and fraudulent clicks. Our goal is to automatically detect and filter out the fraudulent clicks.

A. IP Size Histogram Filter: Overview

As shown in Fig. 4, machine-generated traffic attacks naturally induce an anomalous IP size distribution. Keeping this in mind, we implement a detection system based on the IP size histogram that automatically filters fraudulent clicks associated with any publisher. Our system proceeds through the following main steps.

- First, we group publishers with similar legitimate IP size distributions.
- Second, for each group, we build a statistical model of the click traffic based on historical data. Since the IP size distribution might change over time, a fresh estimation is periodically computed.
- Third, we partition live click traffic of each publisher into separate buckets depending on the IP size value, and filter out sets of clicks of any publishers that violate the computed model with some statistical confidence[1].

[1]We remove from this analysis publishers that do not receive a statistically significant number of clicks in the period of time considered. In these cases, we do not have enough information to provide a statistically sound estimation.

B. Grouping Publishers

Identifying a proper grouping of publishers is the first fundamental step in combating machine-generated traffic. A good grouping of publishers should ensure that publishers in the same group naturally share a similar IP size distribution, while publishers in different groups might not.

As observed in Sec. II-B, the type of services provided by the publisher's website and the type of traffic driven to her website affect the IP size distribution of a publisher. Furthermore, this is also influenced by the geo-location of the source IP addresses visiting her website. The rationale behind this is that different countries have different IP size distributions due to various reasons, such as heavy use of proxy, population density vs. number of IP addresses available, and government policies.

For these reasons, we group together publishers that provide the same type of service (*e.g.*, web search, services for mobile users, content sites, and parked domain websites), and receive clicks from the same type of connecting device (*e.g.*, desktops, smart-phones, and tablets), and from IP addresses assigned to the same country. For instance, if a publisher receives clicks from more than one type of device, its traffic is split depending on the type of devices, and accordingly assigned to different groups. This provides a fine grained grouping of publishers which takes into account the various factors that affect the IP size.

C. Threshold Model for Legitimate Click Traffic

After grouping publishers, we compute a statistical threshold model of the click traffic associated with each group.

First, we aggregate the click traffic received by any publisher within the same group, over a time period τ. To account for the long tail of IP size distributions [6], we bin the click traffic of each publisher using a function of the actual IP size.

Next, we set a minimum quality score, q_{min}, that a set of legitimate clicks should satisfy. Different websites have different quality scores depending on various factors, such as the services provided and the advertisements displayed. Thus, we compute q_{min} as a fixed fraction of the average quality score associated with each group of publishers.

For each group and each bucket we compute a percentile threshold, t. In real time, if any publisher receives more than $t\%$ of her traffic on this bucket, its traffic from this bucket gets filtered. To set t, we carry out a fine-grain scan of all the possible percentiles of this bucket. For each percentile, p, we aggregate the traffic from all the publishers that received more than $p\%$ of their traffic from that bucket, with some binomial confidence threshold. If the quality score of this aggregated traffic is lower than q_{min}, we set p as a candidate threshold. At the end, we pick the threshold, t, to be the candidate threshold that has the highest impact, i.e., the largest number of discarded traffic. This technique takes into account the observed empirical distributions, the number of available samples (IP sizes), and the desired confidence level.

Intuitively, the filtered clicks represent regions of high probability for specific publishers, *i.e.*, spikes in their IP size

distributions, that also have a significantly lower quality than what we would expect for the same group of publishers and set of advertisements.

D. Performance Results

In this section, we assess the effectiveness of the IP size histogram filter in identifying attacks. We implement our system using Sawzall [9], a Google-built language specifically designed to handle massive data sets using a distributed MapReduce-based infrastructure. Each phase of the above filter is distributed across a few hundred machines using the MapReduce framework [10]. For the results described in this section we used a training period of $\tau = 90$ days to build the threshold model, and a testing period of $\tau_{live} = 30$ day.

Figures with sensitive values, including the quality score, the fraud score, and the number of clicks have been anonymized as follows: the original values have been transformed by arbitrary constants so as to preserve trends and relative differences while obscuring the absolute numbers.

IP size Distributions. Fig. 8(a) through Fig. 8(d) depict two groups of publishers, named here A and B for anonymity purpose. These groups consist of publishers whose websites provide similar services and whose click traffic comes from the same country and the same type of device.

Each figure is a four-dimensional plot. The x-axis represents the bucket of the IP size, while the y-axis represents the probability value. Each point is associated with a single publisher and represents the probability that the publisher receives a click of a certain size. In Fig. 8(a) and 8(c) the size of data points represents the number of clicks and the color represents the scaled fraud score. Fig. 8(b) and 8(d) display the same points as in Fig. 8(a) and 8(c) with the difference that the size represents the number of clicks fed to the quality classifier system, and the color represents the scaled quality score. We chose to plot circles with different sizes to represent different levels of statistical confidence.

These figures confirm on real data the motivating intuition discussed in Fig. 4. Fig. 8(a) and Fig. 8(b) show the results on one of the largest groups, comprising hundreds of publishers. Despite the complexity of the problem and the variety of possible attacks, Fig. 8(a) shows that spikes in the IP size distribution of a publisher are reliable indicators of high fraud score. In fact, most points associated with an anomalous high probability are red, thus indicating that they are known to be fraudulent clicks. As an additional validation, in Fig. 8(b) we analyze the corresponding quality score. The spikes corresponding to high fraud score also have very low, or zero, quality score. This confirms that the clicks identified by our systems are indeed fraudulent clicks.

Fig. 8(c) and Fig. 8(d) illustrate a sample group where the IP size distribution detects machine-generated traffic that would have been undetected otherwise. For instance, Fig. 8(c) shows the case of a publisher that has about 70% of its clicks in bucket 6. This spike in distribution is particularly suspicious since all other publishers in the same group have 15% or less click of this size. The quality score associated with this point confirms this intuition. In fact, despite the large number of clicks (size in Fig. 8(d)) we observe a very low quality score. Similarly, a small group of publishers have most of clicks in buckets 11 or 12. Also in this case, the known fraud score is low, but the so is the quality score. This hints towards a previously undetected attacks, possibly orchestrated by a group of colluding publishers.

Analysis of a single bucket. In Fig. 5, we focus on bucket 0 of Fig. 8(a), as this is the bucket with the largest number of data points. We study how the number of filtered clicks, the fraud score, and the quality score vary with the percentile threshold set by the histogram filter for this bucket. We also analyze the number of incremental fraudulent clicks, *i.e.*, the number of fraudulent clicks detected solely by the IP size histogram filter and not by other systems, as well as the incremental quality score, *i.e.*, the quality score associated with the incremental fraudulent clicks. As we can see from Fig. 5, there is a sweet spot around 0.7 that identifies a small fraction of clicks, about 1% of the total number of clicks in this buckets, that have both high fraud score and low quality score.

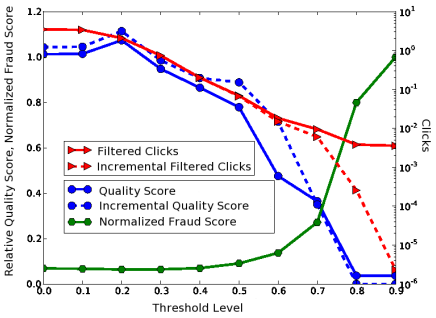


Fig. 5. Analysis of a single bucket.

Performance over time. Fig. 6 shows how the proposed system performs over time. We run the IP size histogram detection every day for a month and we compute the fraud score and quality score of the filtered click traffic. The fraud score is consistently high and stable over time, while the quality score of the filtered traffic remains an order of magnitude lower than the quality score of the unfiltered traffic for the same group of publishers.

Overlap with Other Blacklists. In Fig. 7 we analyze the overlap between IPs filtered by the IP size histogram filter, and IPs listed in Gmail blacklist [7] and in Spamhaus Exploit blacklist (XBL) [8]. For each day, we compile a blacklist of IPs that sent fraudulent clicks during that day. The x-axis represents the time difference between the day we compile our blacklist, and the day the Gmail and Spamhaus blacklists were compiled. A zero value indicates that we compare blacklists associated with the same day. Negative values indicate that our blacklist is some days older than the blacklist compiled by Gmail or Spamhaus XBL. Positive values indicate the opposite scenario. The y-axis represents the percentage of IPs detected with our system that are also found in other blacklists.

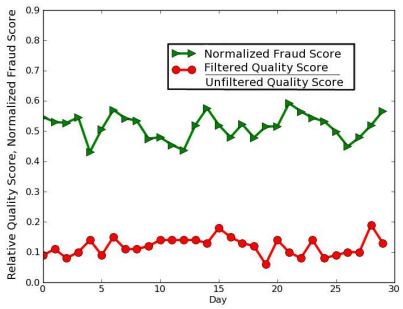


Fig. 6. Fraud score and quality score for different days

Interestingly, we observe that a large percentage of fraudulent clicks are generated by IPs that also generate other kinds of abusive traffic, such as spam emails. In particular, up to 45% of fraudulent clicks are generated by source IPs listed either in Gmail blacklist or in Spamhaus XBL.

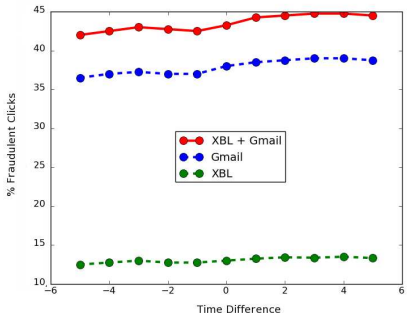


Fig. 7. Percentage of fraudulent clicks generated by IPs listed on the Gmail blacklist or on Spamhaus XBL.

VI. Flagging Entities

The IP size histogram filter described in Section V can distinguish between a set of legitimate and a set of fraudulent clicks by automatically detecting anomalous spikes in a distribution associated with low quality click traffic. To avoid detection a fraudster could attempt to spread its clicks across various buckets so as to achieve the same overall effect while avoiding generating high probability regions in few buckets. Therefore, we need additional methods that look at the entire IP size distribution.

In this section, we consider the IP size distributions associated with entities. An entity can be a user-agent, an e-mail domain, a publisher, a city, a country, and so on. In general, an entity is any dimension that aggregates ad events. For each type of entity, we can build a detection system based on the IP size distribution. This is useful to build several complementary defense mechanisms that protect against different types of attacks.

Assume that we have an estimate of the expected entity's IP size distribution, $r = \{r_i\}_{i=0}^{B_m}$, and that the observed IP size distribution is $f(P) = \{f(P)_i\}_{i=0}^{B_m}$. In this section, we want to detect deviations between the expected and the observed entity distribution, r and $f(P)$, that are induced by machine-generated traffic.

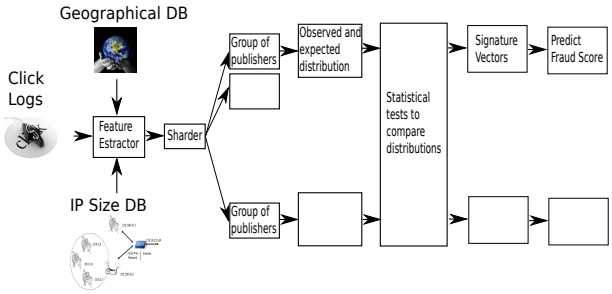


Fig. 9. Flagging entities - system overview: We feed as input the click logs and the information provided by Google IP size and Geographical databases. The feature extractor module extracts only the features we are interested in, as discussed in Section III. Next, the sharder partitions the data into groups based on the type of entity, the type of connecting device, and the geo-location of the source IP. For each of these groups, we compute an expected distribution, r, from the historical data of legitimate clicks. For each entity, we compute an observed distribution of IP sizes, $f = f(P)$. We compare the observed and expected distribution using several statistical methods. Finally, these results are combined in a signature vector specific to each entity and we use this information to predict the entity's fraud score.

A. Flagging Entities: System Overview

Fig. 9 illustrates the workflow of the system we implemented at Google. The first step is the estimation of the expected IP size distribution of each entity. Each group might have a different IP size distribution. However, entities within the same group are expected to share a similar distribution. Since the majority of fraudulent clicks are already filtered out by existing detection systems, we use the aggregate distribution of legitimate IP sizes within each group as an estimation of the true (unknown) IP size distribution for that group. Next, we use a set of statistical methods to accurately characterize the deviation between the observed and expected distribution. As noted in Fig. 4, different attacks result in different deviations in the IP size distribution. Finally, we use an ensemble learning model [11] to combine the outcome of these methods in a signature vector specific to each entity, and we train a regression model that identifies and classifies signatures associated with fraudulent entities.

B. Measuring Anomalous Deviations

To accurately characterize the deviation, if any, between the observed and the expected distribution of each entity we use an ensemble of different statistical methods. These can be grouped in four wide categories:

- vector-based methods: include the L_p distance, the cosine similarity, and the Pearson correlation coefficient. These methods measure either the geometrical or the angular distance between two distributions.
- skewness-based methods: include computing the sample skewness, and the Bowley skewness [12], as well as other domain-specific metrics. These methods measure both the

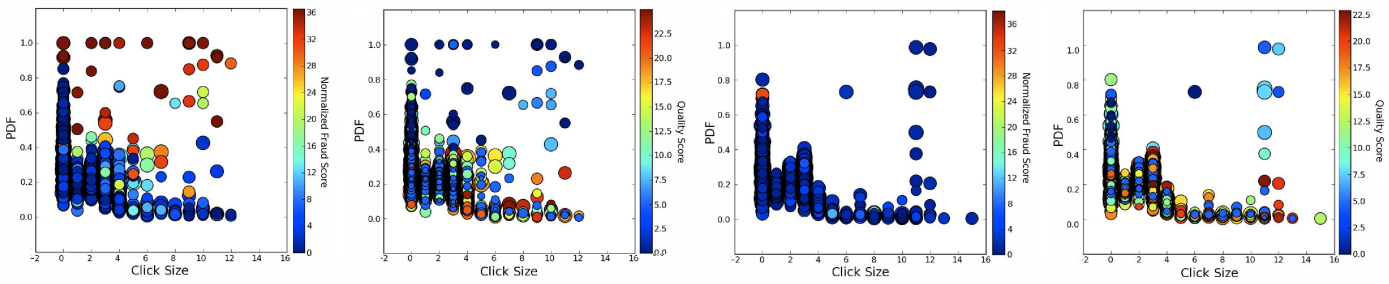


Fig. 8. Figures (a)-(d) show the IP size distribution of two groups of publishers, named A and B for anonymity purpose, which include hundreds of different publishers. Each point represents the percentage of clicks, of a given size, received by a publisher. For each group of publishers, we illustrate two figures. In Figure (a), (c), the color indicates the scaled fraud score. The volume is proportional to the number of clicks associated with the data point. In Figure (b), (d), the color indicates the scaled quality score.

direction – left-skew vs. right-skew, and the magnitude of the asymmetry exhibited by the given distributions.

- entropy based methods: include the Jensen-Shannon and the Kullback-Leibler divergence [13]. These methods measure how concentrated or spread apart the values realized by the given distributions are.
- goodness-of-fit tests: include the Kolmogorov-Smirnov and the Chi-square test statistic. These methods estimate the likelihood that the observed IP sizes are generated by the expected distribution.

C. Combining Statistical Methods

In general, different methods for comparing probability distributions provide different information as they measure different properties. For instance, if we measure the skewness of a distribution, all symmetric distributions will be considered similar to each other as they have null skewness. However, if we measure other properties, such as, the L_2 distance, two symmetric distributions will, in general, be different from each other. Using an ensemble of statistical methods provides a more accurate characterization of the observed deviation than using a single methods. This is particularly important in analyzing massive data sets, comprising a wide range of different patterns.

In order to precisely measure the observed deviation and identify fraudulent entities, we combine the outcome of the statistical methods described in Sec.VI-B in a signature vector, σ_k, specific to each entity, P_k. Intuitively, significant deviations from the expected distribution, measured by several statistical methods, represent strong indicators of fraudulent click traffic. For this reason, we model the fraud score, ϕ_k, as a linear function of the observed deviations,

$$\phi_k = \sum_{j=1}^{p} \theta_j \sigma_{kj} \tag{1}$$

where, σ_{kj} indicates the j-th component of σ_k and θ_j is the weight associated with it. We determine the optimal set of weights, θ, in Eq. (1) that minimize the least-square cost function, $J(\theta) = \sum_{k \in \mathcal{K}} \left(\bar{\phi}_k - \sum_{j=1}^{p} \theta_j \sigma_{kj} \right)^2$ using a stochastic gradient descent method trained on a small subset of entities, $\mathcal{K}$, which includes legitimate distributions and known attacks provided both by other automated systems, and by manual investigation of the logs. The model in Eq. (1) is then applied to a large data set of entities to predict the fraud score as a function of their IP size distribution.

D. Performance Results

Fig. 10 shows the accuracy of the model in Eq. (1) in predicting the fraud score as a function of the number of statistical methods used to compare distributions. First, we analyze the accuracy of our system when all methods are used. Next, we iteratively remove the feature that causes the least amount of variation in the prediction accuracy until we are left with a single feature [14]. We train on 10% of the entities, and test it on the remaining entities. Fig. 10 shows that using multiple comparison methods that measure different type of deviations allows us to reduce the prediction errors, down to a 3% error. This is about 3 times lower than when using a single method. Moreover, we observe that additional methods improve the accuracy of the model but with decreasing gain.

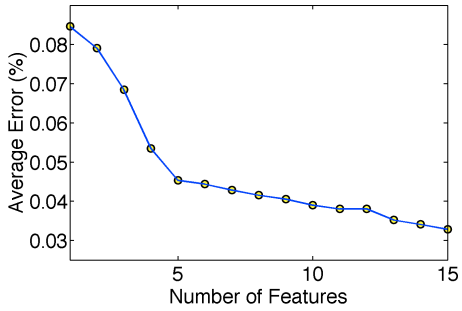


Fig. 10. Prediction accuracy: number of comparison methods vs. average error in predicting the fraud score.

To validate the goodness-of-fit of the model in Eq. (1) we also compute the adjusted coefficient of determination, $\bar{R}^2$:

$$\bar{R}^2 = 1 - \frac{n-1}{n-p} \frac{SS_{err}}{SS_{tot}} \tag{2}$$

where, $SS_{err} = \sum_k (\tilde{\phi}_k - \phi)^2$ is the sum of squares of residuals. Eq. (2) can be interpreted as the amount of variance captured by proposed model. Moreover, in contrast with the R^2 statistic, which does not decrease with more regressors,

$\bar{R}^2$ penalizes the use of a large number of regressors unless it significantly improves the explanatory power of the model. Fig. 11 shows that as we use more statistical tests, the adjusted coefficient of determination increases. This demonstrates that additional features increase the explained variance of the model. When all features are used, the model in Eq. (1) captures over 40% of the total variation in the data. This result is particularly significant in a large data set that includes a wide range of patterns of click traffic.

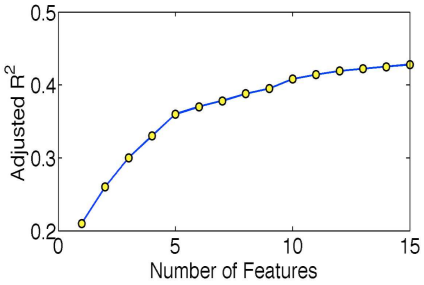


Fig. 11. Prediction accuracy: number of comparison methods vs. R^2. As the number of features increases, the adjusted coefficient of determination, R^2, increases as well, and so does the explained variance.

VII. Related Work

Prior to this work, few research papers presented methods to systematically combat click fraud. [15], [16] propose alternatives to the current pay-per-click (PPC) model in an attempt to remove the incentives for click fraud. [15] proposes to charge advertisers based on the percentage of time an ad is displayed (pay-per-percentage of impressions) rather than on the number of clicks it generated. [16] uses cryptographic credentials to authenticate clients. However, impressions are not a measure of a customer's interest and thus, advertisers cannot easily quantify their return on investment. Moreover, modifying the current PPC model will require changes at a global scale that are not likely to occur in the forecastable future.

A different line of research has proposed a data analysis approach to discriminate legitimate from fraudulent clicks. [17] focuses on the problem of finding colluding publishers. The proposed system analyzes the IP addresses generating the click traffic for each publisher and identifies groups of publishers that receive their clicks from roughly the same IPs. [18] addresses the scenario of a single publisher generating fraudulent traffic from several IPs. The authors propose a system to automatically detect pairs of publisher and IP address that are highly correlated. [19] presents a detailed investigation on how a large botnet was used to launch click fraud attacks.

In the wide area of anomaly detection, [20] represents a recent survey on various categories of anomaly detection systems. Our work in this paper falls in the category of statistical anomaly detection, *i.e.*, we define as an anomaly an observation that is extremely unlikely to have been generated by the probabilistic model assumed. [21] discusses various fraudulent schemes in telecommunications and possible techniques to mitigate them. [22] presents a histogram filter similar in spirit to the IP size histogram filter. However, our work differs in both the problem scope and the approach used to measure deviations and compare distributions.

VIII. Discussion

In Sections V and VI, we used the IP size distribution for detecting machine-generated traffic and we evaluated the effectiveness of our detection with respect to various metrics. In this section, we discuss strengths and limitations of our approach beyond those metrics.

Strengths. First, our approach does not require any identification or authentication of the users generating the clicks. It only uses aggregate statistical information about the IP size. Second, the proposed system is fully automated, has low complexity (it scales linearly in the amount of data to be processed), and is easy to parallelize. This makes it suitable for large-scale detection. Third, the IP size is robust to DCHP reassignment. Clicks generated from a specific host have the same size regardless the specific IP address assigned. This is particularly useful in practice, since a large fraction of IPs are dynamically reassigned every 1-3 days [23]. Fourth, the IP size-based detection is hard to evade. In fact, even if the attacker knows the legitimate distribution of IP sizes for all publishers in her group, and the exact mechanisms used to estimate the IP size, she still would need to generate clicks according to the legitimate IP size distribution. However, the attacker has access only to a limited number of bots. Further, even for those bots, she cannot control the activities of legitimate users sharing the compromised machines. This in turn affects the IP size and limits her ability to arbitrarily shape the IP size distribution.

Limitations and the bigger picture. The methods developed in this paper are currently used as part of a larger detection system deployed at Google in conjunction with complementary techniques. In fact, it is part of Google's strategy to have several defenses in place so that they complement each other in covering the attack space and providing defense in depth. A limitation of our approach is that it requires a statistically significant number of clicks for each publisher. A single publisher that receives a few clicks can evade the proposed system, but at the expense of throttling its own attacks. A set of publishers with a few clicks each can potentially collude to generate an aggregate large number of fraudulent clicks. In this case, approaches that identify colluding publishers, such as [17], would catch them. Moreover, applying the techniques proposed in this paper entails having an automated way of assessing the quality of large bodies of traffic.

Finally, the focus of this paper is on click traffic. However, we believe that the key features exploited here, namely, the IP size generating the malicious activity, and the techniques we developed, are potentially applicable to a wide range of fraud detection problems. Instead of looking at the "size" of IP sources generating clicks, we can analyze the size of

IPs generating other malicious activities, and apply a similar statistical framework for detecting anomalous distributions.

IX. Conclusion

In this paper, we present a data-driven approach to combat machine-generated traffic based on the IP size information– defined as the number of users sharing the same source IP address. Our main observation is that diverse machine-generated traffic attacks share a common characteristic: they induce an anomalous deviation from the expected IP size distribution. Motivated by this observation, we implemented a fraud detection system that detects hit inflation attacks at different levels of granularity using statistical learning models. We show that the proposed model can accurately estimate fraud scores with a 3% average prediction error.

Acknowledgment

The authors would like to thank Prof. Athina Markopoulou and Prof. Christos Faloutsos for their advises and useful discussions, and the Traffic Quality Team at Google, especially Razvan Surdulescu and Michael McNally, for the valuable comments and for helping us deploy the fraud detection system.

References

[1] Interactive Advertising Bureau, Annual Report, http://www.iab.net/media/file/IAB-Ad-Revenue-Full-Year-2009.pdf.

[2] The Kelsey Group, Annual Forecast, http://www.kelseygroup.com/press/pr080225.asp.

[3] N. Kshetri, "The economics of click fraud," *IEEE Security and Privacy*, 2010.

[4] C. Mann, "How click fraud could swallow the Internet," *Wired Magazine*, 2006.

[5] B. Grow, B. Elgin, and M. Herbst, "Click Fraud–The dark side of online advertising," *BusinessWeek online*, vol. 10, no. 02, 2006.

[6] A. Metwally and M. Paduano, "Estimating the Number of Users behind IP Addresses for Combating Abusive Traffic," *ACM SIGKDD 17th International Conference on Knowledge Discovery and Data Mining*, 2011.

[7] B. Taylor, "Sender reputation in a large webmail service," in *Third Conference on Email and Anti-Spam (CEAS)*, 2006.

[8] Spamhaus XBL, http://www.spamhaus.org/xbl/.

[9] R. Pike, S. Dorward, R. Griesemer, and S. Quinlan, "Interpreting the data: Parallel analysis with Sawzall," *Scientific Programming*, vol. 13, no. 4, pp. 277–298, 2005.

[10] J. Dean and S. Ghemawat, "MapReduce: Simplified data processing on large clusters," *Communications of the ACM*, vol. 51, no. 1, pp. 107–113, 2008.

[11] P. Sollich and A. Krogh, "Learning with ensembles: How overfitting can be useful," *Advances in neural information processing systems*, pp. 190–196, 1996.

[12] K. Najim, E. Ikonen, and A. Daoud, *Stochastic processes: estimation, optimization, & analysis*. Butterworth-Heinemann, 2004.

[13] J. Lin, "Divergence measures based on the shannon entropy," *IEEE Transactions on Information Theory*, vol. 37, no. 1, pp. 145–151, 1991.

[14] K. Kira and L. Rendell, "The feature selection problem: Traditional methods and a new algorithm," in *Proceedings of the National Conference on Artificial Intelligence*. John Wiley & Sons Ltd, 1992, pp. 129–129.

[15] J. Goodman, "Pay-per-percentage of impressions: an advertising method that is highly robust to fraud," in *Workshop on Sponsored Search Auctions*, 2005.

[16] A. Juels, S. Stamm, and M. Jakobsson, "Combating click fraud via premium clicks," in *Proceedings of 16th USENIX Security Symposium*. USENIX Association, 2007, pp. 1–10.

[17] A. Metwally, D. Agrawal, and A. El Abbadi, "Detectives: detecting coalition hit inflation attacks in advertising networks streams," in *Proceedings of the 16th international conference on World Wide Web*. ACM, 2007, pp. 241–250.

[18] A. Metwally, F. Emekçi, D. Agrawal, and A. El Abbadi, "SLEUTH: Single-pubLisher attack dEtection Using correlaTion Hunting," *Proceedings of the VLDB Endowment*, vol. 1, no. 2, pp. 1217–1228, 2008.

[19] N. Daswani and M. Stoppelman, "The anatomy of Clickbot. A," in *Proceedings of the first conference on First Workshop on Hot Topics in Understanding Botnets*. USENIX Association, 2007, p. 11.

[20] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Computing Surveys (CSUR)*, vol. 41, no. 3, pp. 1–58, 2009.

[21] R. Becker, C. Volinsky, and A. Wilks, "Fraud Detection in Telecommunications: History and Lessons Learned," *Technometrics*, vol. 52, no. 1, pp. 20–33, 2010.

[22] A. Kind, M. Stoecklin, and X. Dimitropoulos, "Histogram-based traffic anomaly detection," *IEEE Transactions on Network and Service Management*, vol. 6, no. 2, pp. 110–121, 2010.

[23] Y. Xie, F. Yu, K. Achan, E. Gillum, M. Goldszmidt, and T. Wobber, "How dynamic are IP addresses?" *ACM SIGCOMM Computer Communication Review*, vol. 37, no. 4, pp. 301–312, 2007.