Click Fraud Detection: Adversarial Pattern Recognition over 5 Years at Microsoft

Brendan Kitts[1], Jing Ying Zhang[1], Gang Wu[1], Wesley Brandi[1], Julien Beasley[1], Kieran Morrill[1], John Ettedgui[1], Sid Siddhartha[1], Hong Yuan[1], Feng Gao[1], Peter Azo[1], Raj Mahato[1]

[1]Microsoft Corporation, One Microsoft Way, Redmond, WA. USA
bkitts *at] g ml dt cm

Abstract. Microsoft adCenter is the third largest Search advertising platform in the United States behind Google and Yahoo, and services about 10% of US traffic. At this scale of traffic approximately 1 billion events per hour, amounting to $2.3 billion ad dollars annually, need to be scored to determine if it is fraudulent or bot-generated [46], [51], [58]. In order to accomplish this, adCenter has developed arguably one of the largest data mining systems in the world to score traffic quality, and has employed them successfully over five years. The current paper describes the unique challenges posed by data mining at massive scale, the design choices and rationale behind the technologies to address the problem, and shows some examples and some quantitative results on the effectiveness of the system in combating click fraud.

Keywords: click fraud; data mining; invalid clicks; botnet; bot.

1 What is Click Fraud?

Pay Per Click (PPC) auctions are a significant engine for the online advertising economy. They have taken Google from a revenue-less start-up company to a giant making $37 billion per year [18]. They show remarkable properties including high relevance [26], [27] and high conversion rates as keywords are typed by a user actively searching for the advertised product [30].

Unfortunately Pay Per Click has an Achilles Heel. Click fraud is the term used to describe artificial clicks generated on advertisements to either create direct or indirect financial gain from the pay per click payouts [36]. Click Fraud strikes at the heart of Pay Per Click's economic model. Advertisers pay for clicks that don't convert, leading them to need to lower bids. Ad networks generate reduced ROI, resulting in fewer advertisers, and innocent publishers receive lower payouts because of revenue being diverted to cheaters [31], [47], [48], [49], [50]. Click fraud is an area which requires significant investments in detection technology and a constant arms race with attackers in order to ensure that the economics of Pay Per Click work to provide value for advertisers, users and publishers [17], [19], [20], [21], [22], [41].

2 Examples of Click Fraud Attacks

A wide range of Click Fraud attacks have been documented in the literature [2], [7], [9], [11], [16], [25], [38], [39], [40], [42], [53], [57].

One of the earliest was a human clicking operation that was uncovered and sued by Google in 2004. Auctions Experts Limited had used 12 employees to click on ads [57].

Leyden, J. [39] reported on a 115 computer click botnet that was designed to execute a low frequency click fraud attack. The controller of the botnet used a Graphical User Interface to manage its slave computers. Each slave computer was configured to click no more than 15 times per day and target specific paid listings.

A much larger scale botnet was uncovered by Panda Labs [9]. ClickBotA was a 100,000 computer botnet designed to execute another sophisticated, low frequency click fraud attack. Machines were infected by downloading a popular screensaver. Infected machines randomly queried a lexicon from a MySQL database, and fired these against the search engine. The infected machine then selected a listing from the ad-results and asked its Central BotNet Controller whether it “canClick”? If the Central BotNet Controller counted less than 20 clicks against that ad-link in the day, then it responded that it could click.

At Microsoft we filed a law suit against Eric Lam and Supercontinental LLC for their alleged activities running the WOW Botnet [42]. The WOW Botnet executed a click fraud attack across hundreds of thousands of IPs. However in this case the intent wasn’t to directly generate revenue from the clicks, but to actually target advertisers. The objective was to deplete the budget of advertisers, eliminating them from the auction, and allowing the attacker – who was actually an advertiser on the same keywords – to monetize high quality human traffic.

3 Overview Of Paper

In this paper we will discuss the unique data mining systems needed for combating this major problem at one of the largest companies in the world [33], [34], [41]. Because click fraud is an area with real financial implications and adversaries trying to attack the system – some of whom may be reading this paper - we will not be able to discuss the specific algorithms and signatures being used to successively combat fraud. We will instead focus on the unique technology needed to operate at massive scale, and what we have learned over five years in this challenging data mining problem. The lessons that we learned developing this system should be helpful at other very large-scale data mining initiatives, particularly those that are searching for rare events and facing adversarial attackers.

4 Why Click Fraud Detection Is Hard

Click Fraud is an adversarial detection problem [12]. Attackers exploit sophisticated methods to cloak their activities including mimicking human behavior and sometimes hijacking legitimate human traffic. They also evolve – after deploying countermeasures, we've watched as different strains start appearing and attempt to break through. The challenges of the click fraud detection problem can be summarized as (a) Throughput requirements, (b) Rapidity of model updates needed to combat attackers, (c) Low frequency nature of attacks (d) User anonymity, (e) Programmability of attacks, (f) Accuracy requirements, and (g) the need to detect and eliminate the effects of fraud within milliseconds.

At the current scale of traffic serviced by Microsoft, approximately 1 billion events per hour need to be scored to determine if it is fraudulent or bot-generated. To provide a sense for scale, this is 300 times more events than US credit card transactions [8]. Similarly the variables in the online space are massive. There are 4 billion possible Internet Protocol (IP) v4 addresses and, as IP v6 is adopted, there will be $3x10^{38}$ IP v6s [14]. adCenter currently detects over half a million IPs per hour. Looking for combinations of IP behavior against thousands of publishers and millions of keywords creates major computational challenges.

This enormous scale is in stark contrast to the number of fraudulent events. Often these events are spread across large numbers of IPs. For example ClickBotA was configured to click fewer than 20 times per IP [9] and PandaLabs which was configured to click less than 15 times per IP [39]. Low frequency attacks are designed to blend in with statistical noise to avoid detection.

Ad Networks also need to intercept and contain attacks in real-time. This is necessary to prevent disruption to the economics of the auction including depletion attacks [7], [50], [32] and ad clickthrough rate prediction spoofing [48]. This creates enormous challenges for computational infrastructure and informs the architecture that needs to be fielded.

5 Filtration System Principles

5.1 Lossless Processing

There are many hard lessons learned in the development of our traffic quality systems and we believe that these could help inform how other attacker detection systems might be effectively designed.

In 2006 adCenter's filtration system was set up to filter traffic by physically discarding records as soon as they underwent certain quality tests. There were a series of these "stages" since different information was available at different points in processing. The intuition behind this was that "if the traffic is bad why bother spending CPU cycles to process it?"

However because records were being dropped it meant that the number of records coming out from processing was significantly less than the number of records going in. This led to repeated executive escalations due to the concern that adCenter may be dropping traffic. Each of these escalations required a detailed, manual investigation to resolve.

When we re-platformed the system in 2007, we ensured that the new design would be non-lossy – every impression, click and conversion would be processed by our systems so that we could see filtered and unfiltered data [41]. A useful analogy is to think of this like a “Conservation Law for Clicks”. Clicks would be neither created nor destroyed, however could be “transformed” from one classification to another [37]. There are rare circumstances in which traffic may need to be dropped due to a denial of service attack, however we will discuss later that a lighter-weight component is designed for looking for this and we maintain a minimal rate of sampling in order to continue to evaluate the traffic when drastic action is necessary.

5.2 Rapid Update Capabilities

Drops in early stage processes led to another undesirable phenomenon. Because logic was scattered across different systems, it became difficult to update the filtration logic. Simulating the system also required changes to multiple systems, making testing difficult.

adCenter’s filtration logic is now isolated in a single, centralized filtration decision point called the Minerva Module. Filtration decisions are then fed to all downstream systems. This facilitates maintenance, troubleshooting, and rapid updates to the filtration logic.

Rigorous test and deployment systems have been created around this one module so that the model can be rapidly updated. Because of the ability to completely test this component, hotfix model updates can be deployed to production when needed. The centralized architecture has dramatically increased our speed in responding to attacks.

5.3 Rules Representation

It is well-known that a variety of machine learning algorithms could be used to solve a particular classification problem. However what is not as widely known, is the impact of different algorithms on daily operations. For example, one of adCenter’s early systems utilized Naïve Bayes to predict clickthrough rate on keywords. This seemed like a good idea, but led to a global model with hundreds of thousands of weights. When inevitable issues with bad ad selections emerged, there were many possible causes and it was difficult to isolate the problems and fix them. In designing the filtration system, we intentionally chose a rules representation. Rules have a number of advantages for large-scale fraud detection:

Every filtration decision has a reason. Every rule can be identified with a ruleID. When the rule fires, the model outcome (eg. filter / don’t filter) as well as the ruleID that fired, can then both be output for reporting.

Bad rules can be discretely identified. It is possible to report on each ruleID that fired along with that rule’s accuracy in identifying bot traffic. As a result, every rule can be clearly measured. Rules that are performing poorly can be easily removed without upsetting the rest of the model.

Machine-induced and Human Expert rules both supported seamlessly. Rules can of course be induced using a variety of induction techniques including decision trees.

Decisions are auditable. By recording the reason for a traffic classification, it also becomes possible to easily audit the system. In the IAB / MRC audit it is possible to verify that the system is filtering traffic for the reasons that we expect [21], [22], [41].

Ease of troubleshooting. In addition, if a publisher's traffic is being filtered – perhaps inappropriately – the exact rule which fired is recorded. This makes it possible to rapidly debug any issues with the filtration system.

Ease of interpretation. When we do find cases where publishers are pereptarating fraud, being able to see the rules that are detecting them helps greatly in understanding the kind of exploit that they are using.

Updatability. Because the rules are understandable and discrete, it is very easy and fast to update rules. There are no global interactions that need to be considered.

Ease of integrating findings from other teams. One of the really nice features of rules is that they allow researchers, investigators, and others to develop rules or discover methods for detecting attacks which can then be simply plugged in. We solicited rules from a variety of fraud teams including our investigation team, and promised to name the rules they developed after their inventor. Each person who came up with rules could track their rule's revenue impact and fraud detection performance, and they could try to lead in detecting attackers.

5.4 Rule Bitmaps ("Multiple Rules")

As is true of other rules based systems, multiple rules may trigger for a given input. For example, both an IP blacklist and bot detection rule may trigger. When this occurs it is typical in Expert Systems literature for a conflict resolution algorithm to determine the winning rule which should fire [3].

Because of the selection algorithm, some rules may be "shadowed" by other rules which tend to trigger with a higher priority – for example the IP blacklist may have higher priority than the bot detection rule. When this happens it can cause problems for measuring rule quality, since the bot detection rule may "appear" to be performing poorly, however this is because of other rules which are "taking credit" for the traffic. Without the ability to accurately measure the performance of each rule, it is possible to inappropriately remove "poor performing" rules without awareness of the value of the rule or its interactions.

adCenter's approach to this problem is to both create a single "winning" rule for basic reporting purposes, as well as to record every rule which was triggered for detailed analysis with full awareness of rule interactions.

adCenter achieves this by creating a bitmap output for each record. Each bit position on the bitmap corresponds to a rule. For example the following bit representation 01000101 means that rules 1, 3, and 7 all triggered. The unique number for this configuration of rules is 69. An example of a bitmap translated into human-readable format is below, showing that 5 rules triggered for a particular input is below.

```
;;;;;;;;;;;;;;;;;RULEA1;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;RULEC2;;;;;;;;;;;;;;;;;;;;;
;;RULED1;;;RULEA2;DUP;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;
```

The bit position doubles as the rule priority number, with lower bits having higher priority. In the example above, the bit priority means that rule 1 actually is the "winning" rule. In the example above "DUP" is recorded as the "winning" rule for basic OLAP Cube reporting, and this is what is presented to Business users.

Bitmaps allow rule performance to be calculated without overlap from other rules. They also allow the construction of any new model using existing rules. Let's say that we have tested a model, and found that two of the rules appear to be "malfunctioning". We would like to promote the model to production, but without the problematic rules. Unfortunately, there could be unanticipated overlap effects – when the two rules are removed, other existing rules may suddenly soak up the same traffic. Using bitmap it is possible to calculate a new model using any configuration of existing rules – just using the rule output that we have generated from the production system and without having to re-process any data through the new model. The desired rule configuration simply needs to be chosen, and then the rules that are being de-activated simply "zeroed out" from the bitmap. Next the records can be aggregated based on any bit having triggered, resulting in the new filtration performance. Thus we have a "soft simulation" that we can produce for business users that has absolutely perfect fidelity with the production system had we altered the rule mix in code.

5.5 Model Flighting ("Multiple Models")

adCenter Filtration Models (the logic governing filtration) are "flighted". This is done by executing multiple parallel "Candidate" models all at the same time as the Deployment algorithm is running. Each of the models then outputs its filtration results including rule reason for its decision, and these parallel assessments are passed downstream for reporting.

Flighting algorithms in this way makes it possible to explicitly score one filtration model as better than another, and also makes it possible to examine the impact of filtration as opposed to no filtration. For instance, it is possible to compare the true positive and false positive rates of two different algorithms.

Creating the parallel candidate models has also proven one of the key methods for being able to rapidly deploy model updates in a safe manner and observe them before "promoting" them to production. A final benefit is the potential of using the simultaneously evaluating models to actively execute genetic optimization of rules using each flighted model as an instance of the population.

5.6 Redundant Keys ("Multiple Keys")

We have sometimes been asked "what is the definition of a user in the system?" In general the problem with creating a single user key is that it is possible to defeat any such definition.

Combinations of keys or information also can fall victim to fraudulent attackers. For example, the fraudster may generate random strings for their useragent, and approach similar to "cache busting" but designed to bust statistical models. In such a simple approach, any key built off useragent string and IP address would fail [14]. As a result effective identification of users can only be possible by deploying multiple redundant definitions of the user, and ensuring that the detection system is capable of looking for suspicious behavior at many levels and across many variables.

6 Architecture

6.1 Overview

Microsoft's filtration system has been developed to meet the incredible challenges outlined in the previous section. A schematic of the system is shown in Fig. 7. The system entails both real-time components as well as offline systems operating near real-time.

The process starts when a user visits a publisher site (1), and their browser executes a HTTP request that calls to adCenter for ads (2). This sends a request to the adCenter delivery engine (3). Within 2 milliseconds that request is sent to the ARTEMIS (adCenter Real-Time Extendible Model Impression Scoring) real-time scoring system (4) which determines billability in real-time (Filtration) as well as calculating any price adjustments that are needed (Smartpricing [52]) (5). ARTEMIS is also capable of supporting advertiser-controlled real-time bidding based on traffic quality as proposed in [33], and can also send instructions back to reduce the Delivery Engine's level of service specifically to that traffic if the system is under significant attack.

Assuming typical traffic quality, the adCenter Delivery Engine proceeds to hold its auction and return a set of ads to the user (7).

One might think that if traffic is known to be fraudulent, then ads should stop being served back. In fact this is almost never done. The reason is because if the Ad Server changes its behavior and stops serving ads, then attackers can use this behavior as a kind of training signal for rapid training of their attacking programs. By continuing to serve back traffic, it both allows more data to be collected about the attacker, and also impairs the ability of attackers to probe the filtration system. In Email Spam attackers also set up accounts in order to test spam attacks, and the same techniques are used in click fraud [12].

After the ads are served back, the ads can be improperly copied, pasted, and so on since fundamentally they are HTML. Because of this the adCenter click link is encrypted and contains information about where the ad was served, and the originating request [41].

When a user clicks on those ads (8) they are redirected through the adCenter redirection server (9). This server unencrypts the link payload and records the click (10) and sends the user to their ultimate advertiser landing page. If the user then purchases a product they may execute a conversion script (11) which calls back to an adCenter server which logs the conversion event.

The click, impression, and conversion events are recorded in weblogs. These are loaded in batch fashion within the hour for processing by the Minerva offline filtration system (12). Minerva (MINing and Retroactive Analysis) is a very large, 1,000 machine grid, that is designed to be non-lossy and develop the most complete picture possible of the impression, click or conversion event, the events leading up to it, and whether it is billable. In order to preserve the dynamics of the auction, Minerva respects all non-billable decisions made by ARTEMIS, and will itself only re-classify from billable to non-billable [33], but is able to bring significantly more resources to bear on traffic quality and is the final decision-maker for the system. Minerva renders the final decision on billability and flows to all downstream reporting systems so that advertiser reports (13), publisher reports (14), and internal reports (15) all show filtered data. As a result of this architecture, within milliseconds attacks are detectable using ARTEMIS, and after just a little over an hour after an impression and click was recorded, the advertiser is able to see “final” billing results.

A variety of other systems are also important for detection purposes and are used near real-time.

The Fraud workbench (16) allows human Fraud Ops team to review customers and disable their accounts. It also includes an automated machine learning module which runs every hour and creates a probability of fraud which is then provided to the human Fraud Ops team. If the customer is new and the probability is high enough, the account will be paused for a specified number of hours to allow the human Fraud team time to review the customer account. The Fraud workbench is designed in a similar fashion to the very large-scale click filtration system in that it is rules based and each decision is made visible to the Fraud Ops team. The Fraud Ops team can in turn decide to initiate action against a suspected fraudster.

adCenter’s Blacklist Capture System (17) (described in [41]) was developed to pull in 3rd party IP data to help assess whether these IPs are legitimate and billable. The system is currently operational and pulling lists every 15 minutes.

adCenter deploys crawlers (18) to publisher sites to analyze their content and operation and compare against data collected in the course of serving ads. These crawlers analyze everything from site keywords, to looking for deceptive practices and links to other known fraudsters.

Bot instrumentation (20) is technology described in the adCenter Description of Method [41] and provides telemetry for the detailed investigation of traffic sources.

Packet sniffers (21) (internally known as “MH logs”) are also described in the adCenter Description of Method [41] are special weblogs that sample 100% of the HTTP protocol headers in the request received by adCenter. This allows for extremely deep analysis into the origin of the traffic as well as the likelihood of it being produced by automated processes. Packet sniffers operate automatically to collect detailed information on a sample of traffic, and can also be configured to collect all records from particular data sources.

7 Metrics

There are two major ways to define traffic quality (a) true-positive, false-positive, detection rates of confirmed fraudsters, and (b) overall traffic quality metrics that encompass a lot of information about the traffic. adCenter utilizes both approaches.

7.1 Case Base

adCenter is fortunate to be able to collect confirmed fraudulent cases because of the its highly expert and dedicated investigation team. Approximately 40 traffic quality investigation tickets per month are collected. These cases are saved into a “case base” which is literally a copy of logs, but where they are known to be generated by a particular bot. Historical logs with these known tags can then be replayed against the filtration system to see if it detects the known attack.

7.2 Traffic Quality Metrics

We also define Q1 and Q2 metrics which each measure “value per click” compared to typical traffic where 1.0 indicates standard traffic, and values greater than 1.0 indicate less valuable traffic. These metrics do not attempt to measure fraud per se, but instead measure the traffic quality or marketplace health for advertisers on the system.

8 Detection Techniques

The rules used for detection fall into 7 major categories described below:

8.1 Bot Signatures

Every week the engineering team reviews with the support team to look at new types of attacks. The behavior of these attacks is analyzed, and if necessary new rules are developed to combat them. The key is to store literally the electronic format data that the system would have processed, but to separately have a case label for these records. It is then possible to replay the traffic through the system and determine its effectiveness in detecting the historical attack.

8.2 Distribution Tests

Daswani et.al. [10] describe detection in which an expected distribution is compared against an actual distribution. To the degree to which the distributions diverge, the traffic may be artificially generated.

8.3 Scale families and reference curves

We often get questions about how we handle proxy IPs. These are IPs such as AOL or Mobile carrier IPs which service a large amount of traffic. Might not we filter an awful lot of good traffic if we are applying our basic frequency caps and other rules?

The basic approach for handling these IPs is to create a family of versions of a rule but at different scale. For example, say that we have a rule BOT1 which is working very well to identify traffic with less than 20 clicks. We can create another version of this rule that works at 200 clicks, 2000 clicks, and 20,000 clicks. At these levels of scale the anomaly or signature that is being detected is usually much more subtle and so it doesn't need to register comparatively very high, however with the large number of clicks it is statistically significant. The end result is a family of rules eg. BOT1-A..BOT1-E that are designed to operate at different levels of scale, and which naturally handle proxy IPs as well as other large sources of traffic.

The "scale family" is a discretized version of a significance test which takes into account the number of observations when determining whether a variable is statistically significant / significantly different from the norm.

8.4 Traps

Traps are special tests designed to identify bot activity. These generally utilize "active" methods outlined by Daswani et. al. [10].

8.5 Extremata

Extramata are rules designed to identify and remove extreme behavior [4], [5]. In general trying to identify the unusual extremes can help to remove activity that is not typical of focused human browsing behavior as well as pre-emptively "protecting" the system from robotic behavior that has yet to be encountered. In general many extremata can be pre-loaded into the system to catch highly anomalous conditions.

8.6 Key families

In order to be effective, fraudsters need to camouflage their attack perfectly in multiple dimensions. This leads to an important strategy for success. Detection should not just use one or two criteria. It should use as many as possible.

In practice often an effective rule can be developed that looks for activity from an attribute of interest such as an IP. Similar rules can often be developed which look for the same kind of unusual behavior coming a different levels of aggregation from other attributes – such as publisher, advertiser and so on. We call these key families. Therefore even if an attacker cloaks some of their activity they are unlikely to remain hidden in every dimension and the redundant key family rules will tend to detect them.

8.7 Machine-induced decision trees

The final kind of rule is of course the machine learning induced rule. adCenter currently uses a C4.5-like induction method described in [29]. Although these rules are effective, they are less interpretable and are used more in cases where traffic quality is being assessed rather than specific bot signatures.

9 Results

Microsoft adCenter's filtration system has been in operation for over 5 years. In this time the systems have been steadily improved and enhanced.

9.1 Automated Detection Performance

We can summarize the overall performance of the system by looking at filtration rates. These provide some information on how much traffic is being automatically flagged. We would expect that fraudsters should show higher filtration rates, although it is also true that high filtration rate does not necessarily imply fraudulent activity. There are some cases in which advertisements may have been placed improperly attracting lots of accidental clicks for example.

Fig. 1 shows the filtration rate ratio for fraudulent publishers versus non-fraudulent over the same time period. This shows clearly that fraudulent publishers are being filtered more aggressively.

Fig. 2 shows that fraudsters are shifted in terms of their filtration rate with a mode that is 1.8x normal. Table 2 shows if the publisher is a fraudster, then they are likely to be filtered at a rate that is 1.49x higher than normals, with the 10th and 90th percentiles ranging between 0.9x and 1.94x. The difference in distribution of filtration rates between fraud and non-fraud is statistically significant with $p<0.01$ under Wilcoxon Rank-Sum Test.

What do fraudulent publishers look like? Although we cannot go into details, we can show what happened during some of those releases. Fig. 3 shows the days leading up to, and just after, one of our model updates. Four fraudulent publishers suddenly had their filtration rates go to 100%. The investigation team was alerted to these cases because of the high filtration rate, and proactively investigated them to determine the cause of the filtration. They confirmed that this was indeed fraudulent activity, and the rules that were triggered were some specifically geared to particular bots. The support team tried to reach out to these publishers but found that their accounts had been abandoned, and the publishers stopped requesting traffic less than a week after their filtration went to 100%.

9.2 Click Fraud Investigation Team

Table 3 shows reasons for fraudulent account take-downs provided by the human support team, as well as the true positive rate for those reasons. True positives are finalized only after sometimes lengthy investigations in which the investigation team is able to determine whether a customer is engaging in fraud or is not.

Most of the fraudulent revenue identified and remitted by the investigation team was found after seeing a high invalid click rate (61%). The true positive rate for investigations triggered by high invalid click rate is also around 75%, which makes it the top performing detection category given the volume of fraudulent activity being removed. In addition, the high quality of automated detection has not only improved the fraud team's accuracy and speed of detection, but has also freed it up to spend more time conducting deep investigations.

9.3 Traffic Quality

Although the effectiveness in detecting known fraudsters is promising, it remains to be seen what is the overall effect on the advertiser in their day-to-day advertising? We can measure traffic quality using our Q1 and Q2 metrics that we introduced earlier.

Table 1 shows traffic quality for major classes of rules used in adCenter. Known bots clearly have very bad traffic (eg. Q2=1,773). A range of other rules are also shown, some of which may not necessarily be fraudulent, but for which the business has decided not to bill such as Defective traffic (which is actually a little better than the norm at Q2=0.9; so is likely human traffic, but for which we can't bill due to errors in the click request or out-of-date account information). Interestingly, the IAB Robots and Spiders List [63] – which represents a commendable industry effort to track and catalog bots for companies to implement industry standard filtration - produces only 0.02% and 0.04% additional filtration in our system (Q2=3.3 and 2.4).

Fig. 4 shows traffic quality for filtered traffic as well as kept traffic. The filtered traffic undergoes massive spikes in the Q2 measure, from being consistently about 5x worse than billable traffic to sometimes as high as 50x. In addition, the irregularity of the traffic shows that this traffic would be extremely disruptive for advertisers that expect a consistent standard of traffic and value. The kept traffic line (Fig. 5), in contrast, is extremely stable. Indeed, the metric has shifted by only a few percent over the period. This shows that adCenter is delivering good value to advertisers, protecting them from bad traffic, and is maintaining consistent value over a long period of time.

Fig. 6 (bottom right subplot) shows post-filtration, billable clicks, which is the same data that are reported throughout Microsoft, sent to Advertisers on their billing reports, reported to Publishers, and so on. The constituent bot traces, artificial level changes and spikes are gone, leaving a human-looking profile. By removing the bot activity and leaving in place a clean, human timeseries, Microsoft is better able to assess the performance of its online services, as well as ensuring that machine learning systems throughout the platform are learning on human data, allowing adCenter to work to maximize user search relevance, advertiser performance and publisher value [59].

10 Conclusion

Click Fraud is a major challenge for online advertising. Effective execution requires constant investment and development. We have discussed the design of Microsoft's click filtration systems, and the choices that were needed to operate at massive scale, and to detect sophisticated adversarial attackers. We believe that the design that we have developed – including real-time and near-real-time components, rapid update capabilities, and so on – should translate well to other large-scale fraud detection domains.

Acknowledgments. We would like to thank Raj Mahato, Albert Roux, Ron Mills, Brandon Sobottka, Matthew Rice, Sasha Berger, Jigar Mody, Dennis Minium, Kamran Kanany, Tudor Trufinescu, Dinesh Chahlia, Ken Pierce, Hank Hoek, Tao Ma, Karl Reese, Narayanan Madhu, Dimitry Berger, Rageesh Maniyembath, Meena, Joseph Morrison, Kiran Vemulapalli, Anthony Crispo, Matthew Bisson, Igor Chepil, Matthew Ford, Sachin Ghani, Amjad Hussain, Steve Marlar, Bill Morency, Gerry Moses, Steve Sullivan and many others.

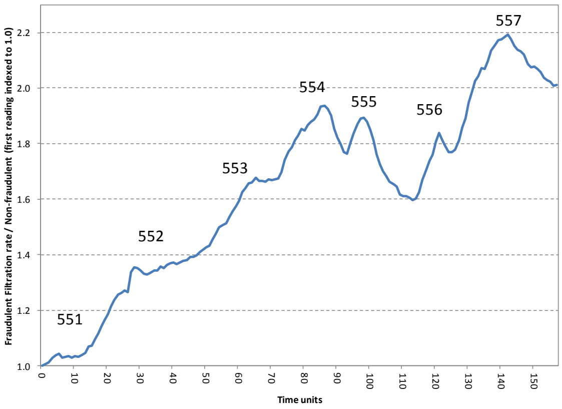

Fig. 1. Filtration rate of fraudulent publishers versus non-fraudulent publishers, expressed as a ratio of the fraudulent rate to non-fraudulent. x-axis is time, y-axis is ratio, and indexed to the ratio on the first date in the timeseries above. The numbers indicate major product releases.

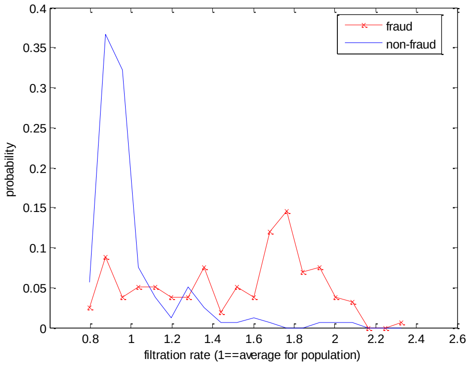

Fig. 2. Filtration rate ratio for fraudulent versus non-fraudulent publishers

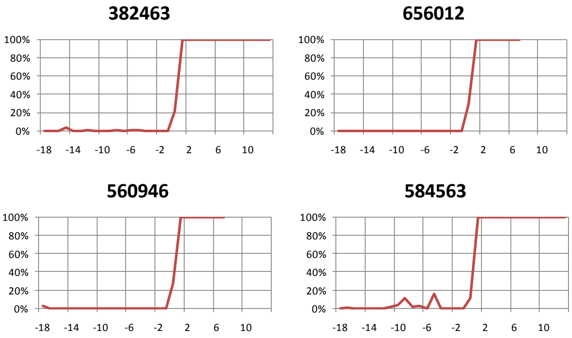

Fig. 3. Filtration rates for four fraudulent publishers. After a rule update their filtration rates went to 100%. The time-axis shows days leading up to a model update and following the model update.

Table 1. adCenter Rule Categories and Q1 and Q2 traffic quality metrics.

Category	% clicks	Q1	Q2
Billable	82.63%	1.0	1.0
Double click	6.48%	1.9	1.8
Known bot	2.47%	937.8	1,773.6
Staleness	2.13%	2.4	3.5
User freq cap	1.92%	7.2	12.7
Suspicious	1.92%	2.5	3.3
Bad proxy	1.08%	84.3	71.2
Business non-billable	0.70%	1.6	1.8
Refractory	0.35%	3.1	1.9
Outlier	0.15%	3.2	4.2
Defective	0.12%	1.3	0.9
IAB Browser list	0.04%	2.8	3.3

IAB Robots and Spiders	0.02%	1.7	2.4

Table 2. Filtration Rate for Fraudulent vs Non-Fraudulent Publisher (1.0 = Average for Population)

Filtration Rate (ratio vs Population)	Fraudulent Publisher	Normal Publisher
Mean	1.4942	1.0000
Variance	0.1456	0.0479
90th pctl	1.9404	1.2794
10th pctl	0.9038	0.8508

Table 3. Click Quality Investigations Team Reason for Investigation and True Positive Rate

% of true fraud revenue detected	TP Rate % (accounts)	Reason
61.09%	75%	High Invalid Click Rate
18.85%	50%	R1
9.88%	63%	R2
8.79%	75%	R3
1.09%	50%	R4
0.29%	33%	R5

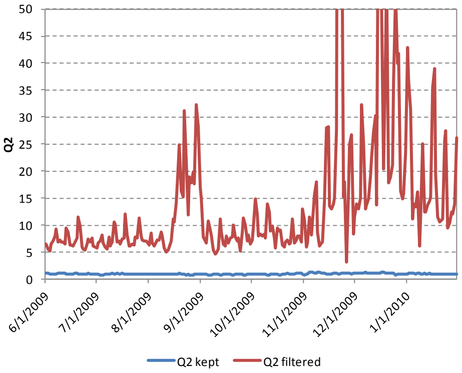

Fig. 4. Q2 statistic for filtered (upper irregular line) versus kept traffic (lower line). The upper line has Q2 rates ranging from 5 to over 50. This activity is variable because of ongoing fraud and bot activity. The lower line which is billed traffic is extremely stable. adCenter is filtering out the activity in the upper line and trying to maintain good performance for advertisers.

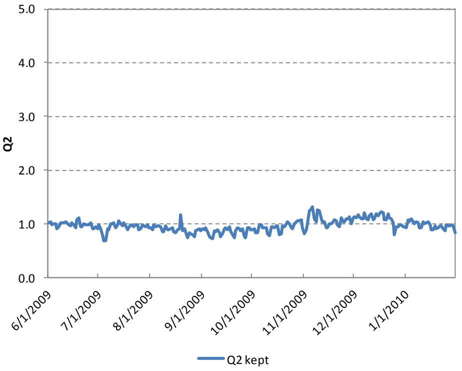

Fig. 5. Close-up of the Q2 timeseries for kept traffic. The filtration system helps to ensure that the metric varies by only a few percent from its average value across six months of campaigns, despite rampant click fraud attacks that are underway.

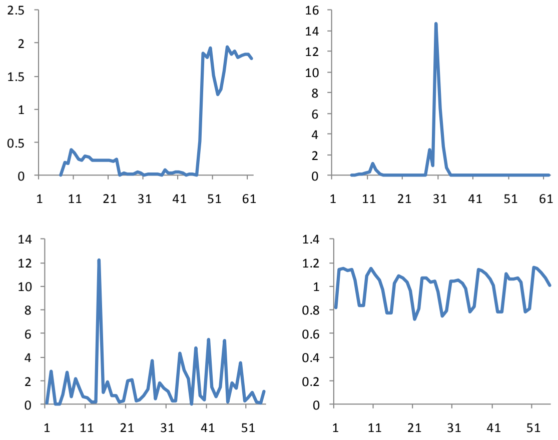

Fig. 6. 60 day timeseries for 4 rules. The first three are all discrete rules designed to look for bot activity. The top left is a clearly robotic process that runs at different levels of aggressiveness. The top right is a burst attack that was observed at day 31 and then disappeared. The bottom left shows continuous attack activity. The final graph on the bottom right shows the post-filtration timeseries. By removing the bot activity and leaving in place a clean, human timeseries, Microsoft is better able to optimize its ad engine for users, advertisers and publishers.

References

1. Article 29 Working Group: http://ec.europa.eu/justice_home/fsj/privacy/workinggroup/index_en.htm (2010)

2. Boyd, C.: IE Used to Launch Instant Messaging and Questionable Clicks, blog.SpywareGuide, http://blog.spywareguide.com/2006/10/ie_used_to_launch_instant_mess.html (2006)
3. Buchanan, B. and Shortliffe, E.: Rule-Based Expert Systems, Addison Wesley, Reading, MA. http://www.u.arizona.edu/~shortlif/Buchanan-Shortliffe-1984/Chapter-02.pdf (1984)
4. Buehrer, G., Stokes, J. and Chellapilla, K.: A Large-Scale Study of Automated Web Search Traffic, Proceedings of the 4th international workshop on Adversarial information retrieval on the web (AIRWEB) 2008, April 22, 2008. http://research.microsoft.com/apps/pubs/?id=69505 (2008a)
5. Buehrer, G., Stokes, J. Chellapilla, K. and Platt, J.: Classification of Automated Web Traffic, in Weaving Services and People on the World Wide Web, Springer Berlin Heidelberg (2008b)
6. Chickering, M.: WinMine Toolkit, Microsoft Research Technical Report MSR-TR-2002-103, http://research.microsoft.com/en-us/um/people/dmax/WinMine/WinMine.pdf (2002)
7. Claburn, T.: Microsoft Sues Three For Click Fraud, http://www.informationweek.com/news/internet/ebusiness/showArticle.jhtml?articleID=217900193 June 16 2009 (2009)
8. Credit Cards.com: http://www.creditcards.com/credit-card-news/credit-card-industry-facts-personal-debt-statistics-1276.php (2010)
9. Daswani, N. and Stoppelman, M.: The Anatomy of ClickBot A, Usenix, HotBots 2007, http://static.usenix.org/event/hotbots07/tech/full_papers/daswani/daswani.pdf (2007)
10. Daswani, N., Mysen, C., Rao, V., Weis, S., Gharachorloo, K. and Ghosemajumder, S.: Online Advertising Fraud, in Jakobsson, M. and Ramzan, Z. (eds), Crimeware, Symantec Press http://shumans.com/onlineadvertisingfraud.pdf (2008)
11. Edelman, B.: The Spyware - Click-Fraud Connection -- and Yahoo's Role Revisited, April 4, 2006, http://www.benedelman.org/news/040406-1.html#e1 (2006)
12. Goodman, J.: Spam Filtering: Text Classification with an Adversary, Invited Talk at KDD 2003 Workshop on Operational Text Classification Systems (2003)
13. Ipensatori: http://ipensatori.com/
14. Fielding, R. et. al.: Hypertext Transfer Protocol -- HTTP/1.1, Network Working Group, RFC 2616 (1999)
15. Fraser, N.: Neural Network Follies, September 1998 http://neil.fraser.name/writing/tank/ (1998)
16. Gandhi, M. Jakobsson, M., Ratkiewicz, J.: “Badvertisements: Stealthy Click-Fraud with Unwitting Accessories, Anti-Phishing and Online Fraud”, Journal of Digital Forensics Practice, Vol. 1, Special Issue 2, November 2006 (2006)
17. Ghosemajumder, S.: Findings on Invalid clicks, July 21, 2006 http://googleblog.blogspot.com/2006/03/update-lanes-gifts-v-google.html (2006)
18. Google: Form 10-Q, Quarterly Report Pursuant to Section 13 or 15(d) of the Security Exchange Act of 1934, United States Security and Exchange Commission, http://www.sec.gov/cgi-bin/browse-edgar?action=getcompany&CIK=0001288776&owner=include (2010)
19. Google Ad Traffic Quality Resource Center, http://www.google.com/adwords/adtrafficquality/
20. Yahoo: Yahoo Traffic Quality Center, http://searchmarketing.yahoo.com/trafficquality/
21. Google: Google IAB Click Measurement Description of Method http://adwords.google.com/support/aw/bin/answer.py?hl=en&answer=153707
22. Yahoo: Yahoo Search Marketing Click Measurement Guidelines: Description of Method (2009)

23. Internet Advertising Bureau: IAB Impression Measurement Guidelines, http://www.iab.net/media/file/US_meas_guidelines.pdf (2005)
24. Internet Advertising Bureau: IAB Click Measurement Guidelines, http://www.iab.net/media/file/click-measurement-guidelines2009.pdf (2009)
25. Jackson, C., Barth, A., Bortz, A., Shao, W. and Boneh, D.: Protecting Browsers from DNS Rebinding Attacks, Proceedings of the 14th ACM conference on Computer and communications security, October 26, 2007, pp. 421 – 431 (2007)
26. Jansen, B. J.: The Comparative Effectiveness of Sponsored and Non-sponsored Results for Web Ecommerce Queries. ACM Transactions on the Web. 1(1), Article 3, http://ist.psu.edu/faculty_pages/jjansen/academic/pubs/jansen_tweb_sponsored_links.pdf (2007)
27. Jansen, B., Flaherty, T., Baeza-Yates, R., Hunter, L., Kitts, B., Murphy, J.: The Components and Impact of Sponsored Search, Computer, Vol. 42, No. 5, pp. 98-101. May 2009 http://ist.psu.edu/faculty_pages/jjansen/academic/pubs/jansen_sponsored_search_ieee.pdf (2009)
28. Kantarcioglu, M., Xi, B., Clifton, C.: A Game Theoretic Approach to Adversarial Learning, National Science Foundation Symposium on Next Generation of Data Mining and Cyber-Enabled Discovery for Innovation, Baltimore, MD, http://www.cs.umbc.edu/~hillol/NGDM07/abstracts/poster/MKantarcioglu.pdf (2007)
29. Kitts, B.: Regression Trees, Technical Report, http://www.appliedaisystems.com/papers/RegressionTrees.doc (2000)
30. Kitts, B. Laxminarayan, P. and LeBlanc, B.: Cooperative Strategies for Keyword Auctions, First International Conference on Internet Technologies and Applications, Wales. September 2005. (2005)
31. Kitts, B. Laxminarayan, P. and LeBlanc, B., Meech, R.: A Formal Analysis of Search Auctions Including Predictions on Click Fraud and Bidding Tactics, ACM Conference on E-Commerce - Workshop on Sponsored Search, Vancouver, UK. June 2005. http://research.yahoo.com/workshops/ssa2005/papers/kitts-ssa2005.doc (2005)
32. Kitts, B., LeBlanc, B.: Optimal Bidding on Keyword Auctions, Electronic Markets: The International Journal of Electronic Commerce and Business Media, Vol. 14, No. 3. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.97.5261&rep=rep1&type=pdf (2004)
33. Kitts, B.: Click Fraud Protector, US Patent Application 11/559,291, November 13, 2006 http://appft1.uspto.gov/netacgi/nph-Parser?Sect1=PTO1&Sect2=HITOFF&d=PG01&p=1&u=/netahtml/PTO/srchnum.html&r=1&f=G&l=50&s1=20080114624.PGNR. (2006a)
34. Kitts, B.,, Najm, T., Burdick, B.: Identifying Automated Click Fraud Programs, US Patent Application Number 11/745,264, May 7, 2007 http://www.freepatentsonline.com/20080281606.pdf (2006b)
35. Kitts, B., Hetherington, K., Vrieze, M.: Large-scale mining, discovery and visualization of WWW user clickpaths, International Journal of Image and Graphics, Vol. 2, No. 1, pp. 21-48, World Scientific Publishing Company http://rux3zw.blu.livefilestore.com/y1p-Bu06VAB9MxYPqE0EE2VVckMZUnF9vombK4inkzhG4-q2FrpgnEHKvfrjP-EEZzCF4fd_gvF5m6YsTKGBVkfxh0gCRwVpeWL/Kitts_IJExpertSystems2.zip?download (2002)
36. Kitts, B. LeBlanc, B., Laxminarayan, P.: Click Fraud, American Society for Information Science and Technology Bulletin, December / January 2006, pp. 20-23,

http://www3.interscience.wiley.com/cgi-bin/fulltext/112738427/PDFSTART?CRETRY=1&SRETRY=0 (2006)

37. Kitts, B.: Introducing adCenter ClickIDs, adCenter Blog, June 17. http://community.microsoftadvertising.com/blogs/advertiser/archive/2009/06/17/introducing-adcenter-clickids.aspx (2009)
38. Leyden, J.: "Click-fraud menace spreads using IM", The Register, October 6, 2006, http://blog.spywareguide.com/2006/10/ie_used_to_launch_instant_mess.html (2006a)
39. Leyden, J.: "Botnet implicated in click fraud scam", The Register, May 15, 2006, http://www.theregister.co.uk/2006/05/15/google_adword_scam/ (2006b)
40. Cross-site request forgery http://en.wikipedia.org/wiki/Cross-site_request_forgery
41. Microsoft: Microsoft adCenter Click Measurement Description of Method, posted October 2009 https://adcenterhelp.microsoft.com/Help.aspx?market=en-US&project=adCenter_live_Std&querytype=topic&query=MOONSHOT_CONC_ClickMethod.htm (2009)
42. United States District Court: Microsoft vs Eric Lam et. al., Civil Case Number CO 9-0815, United States District Court, Western Division of Washington at Seattle, June 2009 http://graphics8.nytimes.com/packages/pdf/business/LamComplaint.pdf (2009)
43. Microsoft: Microsoft Privacy Policy, http://privacy.microsoft.com/en-us/bing.mspx (2010a)
44. Microsoft: Microsoft Privacy Policy, http://www.bing.com/community/blogs/search/archive/2010/01/19/updates-to-bing-privacy.aspx (2010b)
45. Microsoft: Microsoft Privacy Policy http://microsoftontheissues.com/cs/blogs/mscorp/archive/2010/01/19/microsoft-advances-search-privacy-with-bing.aspx (2010c)
46. Microsoft: Microsoft Form 10-Q, Quarterly Report Pursuant to Section 13 or 15(d) of the Security Exchange Act of 1934, United States Security and Exchange Commission, http://www.microsoft.com/msft/SEC/default.mspx (2010)
47. Mungamuru, B. and Garcia-Molina, H.: Managing the Quality of CPC traffic, (2008)
48. Mungamuru, B., Weis, S., Garcia-Molina, H.: Should Ad Networks Bother Fighting ClickFraud (Yes, They Should), (2008)
49. Mungamuru, B. and Garcia-Molinja, H.: Predictive Pricing and Revenue Sharing, (2008)
50. Mungamuru, B. and Weis, S.: Competition and Fraud in Online Advertising Networks, in Tsudik, G. (ed), Foundations of Computing Lecture Notes in Computer Science, Vol. 5143, pp. 187-191, Springer-Verlag, 2008. (2008)
51. Nielsen: Nielsen Reports December U.S. Search Rankings, January 13, 2010 http://blog.nielsen.com/nielsenwire/online_mobile/nielsen-reports-december-u-s-search-rankings/ (2010)
52. Rey, B. and Kannan, A.: Conversion Rate Based Bid Adjustment for Sponsored Search Auctions, WWW 2010, April 24-30, Raleigh, NC. (2010)
53. Schonfeld, E.: The Evolution Of Click Fraud: Massive Chinese Operation DormRing1 Uncovered, Oct 8, 2009, http://techcrunch.com/2009/10/08/the-evolution-of-click-fraud-massive-chinese-operation-dormring1-uncovered (2009)
54. Sterling, G.: Microsoft Earnings Beat Estimates Online Services Post Loss, More On Bing And The iPhone, Search Engine Land, January 29 2010 http://searchengineland.com/microsoft-earnings-beat-estimates-online-services-post-loss-more-on-bing-and-the-iphone-34696 (2010)
55. Sherman, C.: Yahoo Settles Clickfraud Lawsuit, Search Engine Watch, June 28, 2006 http://blog.searchenginewatch.com/060628-202403 (2006)

56. Tuzhilin, A.: The Lane’s Gifts v. Google Report, Google Blog, http://googleblog.blogspot.com/pdf/Tuzhilin_Report.pdf (2006)
57. Weinberg, N.: Google Wins Click-Fraud Case vs Auction Experts, Web Pro News, July 5, 2005 http://www.webpronews.com/topnews/2005/07/05/google-wins-clickfraud-case-vs-auction-experts (2005)
58. Whitney, L.: Bing grabs 10 percent of search market, CNET News, September 2009, http://news.cnet.com/8301-10805_3-10354394-75.html (2009)
59. Wu, G. and Kitts, B.: Experimental comparison of scalable online ad serving, Fourteenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD2008), pp. 1008-1015. http://rux3zw.blu.livefilestore.com/y1pVToJ3_G2tk3693OleN7sKfA8z1_-JiuSzVMGiKqRJBfUFZJ1156HAUoF4aAETTAvfkJX434VcOi1M7eWYaAh4PDrSmMQXQ-P/AdServing_kdd2007_v20.doc?download (2008)
60. Yahoo: Yahoo IAB Click Measurement Description of Method http://rds.yahoo.com/_ylt=A0oGk1DU0MpLWU8AAupXNyoA;_ylu=X3oDMTEya2E4ZjNuBHNlYwNzcgRwb3MDMQRjb2xvA3NrMQR2dGlkA0RGUjVfODQ-/SIG=12q1jopde/EXP=1271669332/**http%3a//help.yahoo.com/l/us/yahoo/ysm/sps/screendocs/click_meas_dom.pdf (2009)
61. Zarokian, P.: “Click Fraud – Is it Happening to you?”, Who’s Clicking Who Report, October 1, 2003, http://www.clickfraudreport.com/archives/2003/10/index.html (2003)
62. Zhang, L. and Guan, Y.: Detecting Click Fraud in Pay-Per-Click Streams of Online Advertising Networks, The 28th International Conference on Distributed Computing Systems, http://csis.pace.edu/~ctappert/dps/d861-09/team5-2.pdf (2008)
63. Internet Advertising Bureau: IAB/ABCe International Spiders & Bots List, Internet Advertising Bureau, http://www.iab.net/iab_products_and_industry_services/1418/spiders (2010)

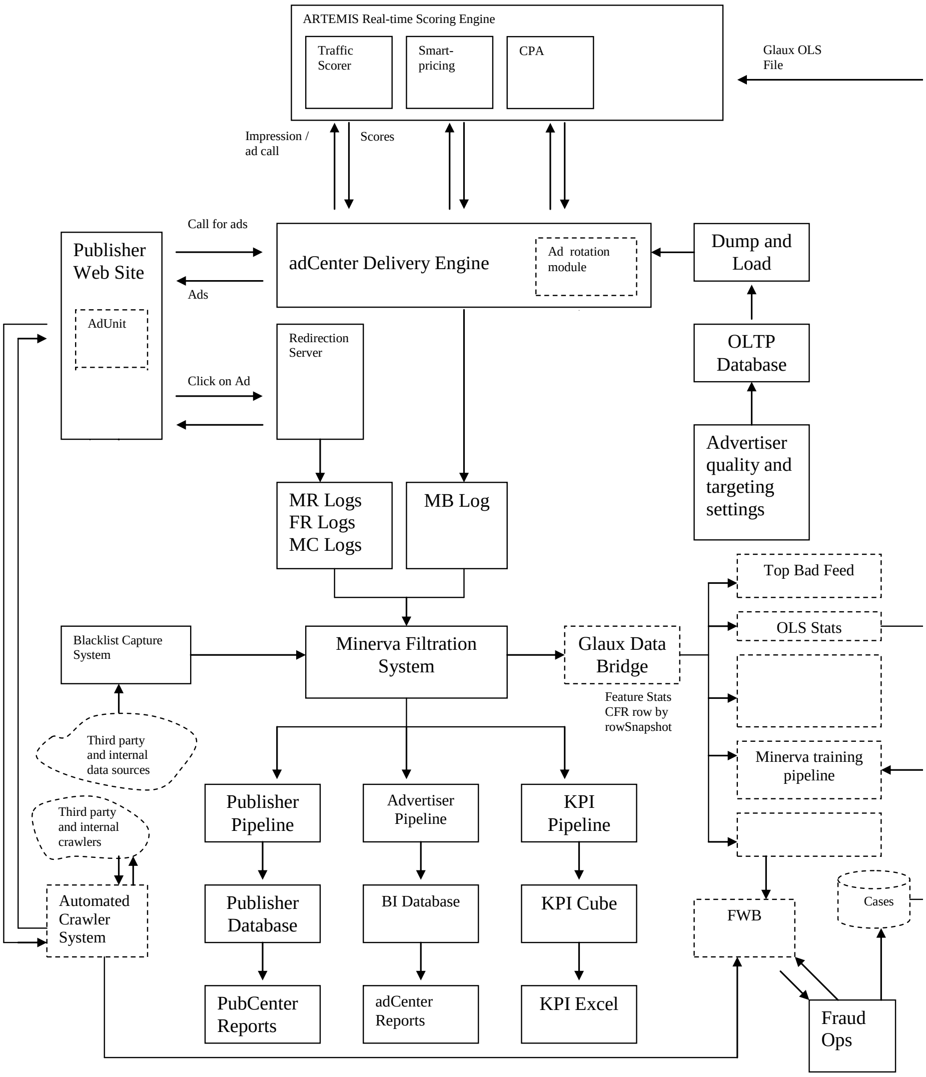

Fig. 7. adCenter Traffic Quality Architecture