Click Fraud Botnet Detection by Calculating Mix Adjusted Traffic Value

A Method for De-Cloaking Click Fraud Attacks that is Resistant to Spoofing

Brendan Kitts
Applied AI Systems
Seattle USA
bkitts@appliedaisystems.com

Jing Ying Zhang, Gang Wu, Raj Mahato
Microsoft Corporation
Seattle USA
jingzha@microsoft.com

***Abstract*— Click Fraud remains one of the most durable fraudulent schemes online. With 50 billion dollars being generated per year by Google alone, a fraudulent publisher is able to capture a significant amount of revenue with a small investment. The most well heeled click fraud attacks employ large distributed botnets, deceptive publisher pages, malware infection, and fake conversion "chaff" in an attempt to cloak fraudulent activity. We describe an algorithm that we call Mix Adjustment which corrects for traffic bias differences. The method is scalable and we show a simple implementation that can be applied to current weblog processing systems. We show two case studies of this algorithm on real fraud detection problems: (a) WOW Bot net detection, (b) Advertiser fraud detection.**

Keywords—click fraud; bot; fraud (*key words****)***

I. INTRODUCTION

Click Fraud schemes aim to produce clicks for financial gain. One of the more insidious methods is to create bogus conversions to give the appearance that traffic is valuable. This can defeat click fraud detection schemes that are analyzing conversion rates. The current paper describes an algorithm that we call "Mix Adjustment" that is designed to analyze traffic quality and is resistant to spoofing. The method is applied to two historical fraud schemes.

II. MIX ADJUSTMENT ALGORITHM

Mix Adjustment is a method that assumes that a particular variable is "unreliable" or is "under suspicion" and takes a second opinion on the variable. For example, an advertiser may have switched off conversion tracking and so their conversions always show as 0 – this may give a false impression that a publisher is producing fraudulent traffic. Likewise fraudsters may be producing "conversion chaff" in an effort to avoid detection. In these situations we want to come up with a more robust estimate of the value of the traffic.

The strategy is to essentially ignore the suspect variable and look at the value of the underlying entities that are associated with the traffic. For instance, if a publisher is under suspicion of producing fraudulent traffic, let's ignore the publisher and look at the Internet Protocol addresses (IPs) that are trafficking on their site and how they performed elsewhere. Each IP is like a little bag of gold. It is expected to have a particular value based on what we have observed it generating elsewhere. We now add up those bags of gold based on how much they've trafficked on the publisher's site. If the publisher's actual conversion performance is higher than that expected based on the sum of IP value, then it is likely that the publisher may be engaging in conversion fraud.

III. FORMAL DESCRIPTION

Let z be a metric that we wish to estimate, y be a variable and Y be an entity which is "under suspicion". For example, z may be Conversions, and y=Y might be Publisher=abc.com. We would normally estimate z as follows:

$$C(z|y = Y) = I(y = Y) \cdot \mathrm{E}[z|y = Y] \quad (1)$$

Where E[] is the expected value, and $I(y=Y)$ refers to the number of impressions or events observed on the suspicious entity. A Mix Adjusted estimate can be defined as below:

$$MAC(z|y = Y) = I(y = Y) \cdot$$
$$\sum_{x=X_i} \Pr(x = X_i|y = Y) \cdot \mathrm{E}\,[z|x = X_i \wedge y \neq Y] \quad (2)$$

The Mix Adjustment calculation introduces a *proxy variable x* to calculate the value of the traffic in locations other than the one that is under suspicion. For fraud applications we recommend selecting a proxy variable that is *prolific*, in that it allows for sampling of traffic widely across the internet, high cardinality, and economically expensive to obtain, so that it is resistant to a fraudster acquiring control of the proxy. IP address is one variable that fits these criteria. x=X_i refers to the proxy variable x taking on value X_i (eg. IP = 1.1.1.1). In the examples that follow we have used this proxy in all cases, although it is not the only proxy that can be used.

The calculation assumes that the value of traffic for the proxy variable on other sites $\mathrm{E}[z|x = X_i \wedge y \neq Y]$ is formally independent of the potentially biased variable $\mathrm{E}[z|x = X_i \wedge y = Y]$; meaning that estimation of traffic value using the proxy will also estimate the traffic for the suspect entity y.

Mix Adjustment can also be understood as creating a simple Conditional Imputation model [7] with the objective of estimating z. The model that we build is based on the proxy information that is present in the traffic record; *but purposely not including any traffic generated by - or observed with - the suspect variable, as that traffic may have been compromised;* $\mathrm{E}\,[z|x = X_i \wedge y \neq Y]$.

IV. IMPLEMENTATION

Mix Adjustment calculations can be implemented to run on web log data in O(I) time and O($\#X$) space, where I are the number of events in a weblog and $\#X$ the number of values the proxy can take. A sequential scan of the weblog is used to build a hash table (X_i, z) keyed by the proxy variable-value X_i and storing the traffic quality statistics z using O($\#X$) space. A second scan is then used to look up the proxy estimate z and calculate (2) for every weblog event. We have also enclosed an example relational database implementation in pseudo-SQL (Fig. 1). The query reads table Clicks(Keyword, ClientIP, AdvertiserID, IsConverted), where each record represents a click of a user with IP address equal to ClientIP on an AdvertiserID's Keyword, and IsConverted has value 1 if a conversion occurred and 0 otherwise.

```
select advertiserid, sum(1) advclicks, sum(isconverted) advconvs,
sum(convrate_all) conv_mixadjusted,1.0*sum(isconverted)/sum(1)
convrate,1.0*sum(convrate_all)/sum(1) maconvrate
from
(select * from Clicks where cast(biddedkeyword as nvarchar(15)) like
'%xxxx%' ) a
left outer join
(select x.clientip,count(*) clicks,sum(isconverted) conv,
case count(*) when 0 then 0 else 1.0*sum(isconverted)/count(*) end
convrate_all from Clicks x
where not cast(x.biddedkeyword as nvarchar(15)) like '%xxxx%'
group by x.clientip ) b
on a.clientip=b.clientip
group by a.advertiserid
order by a.advertiserid
```

Fig. 1. Example SQL code that implements the Mix Adjustment Algorithm. The code above calculates the Mix Adjusted Conversion Rate (maconvrate) for advertisers bidding on keyword 'xxxx' in which we are trying to find victims of a botnet style attack.

V. FRAUD DETECTION APPLICATIONS

We next discuss several applications of the fraud detection method, and show examples of performance.

A. Which Advertisers are Victims of a Depletion attack?

In 2008 Microsoft adCenter was attacked by the WOW Botnet. According to court documents, WOW attacked adCenter from February 2008 – August 2008. It used hundreds of thousands of IPs in its attack, and clicked on competitors with massive scale [1-3].

According to court documents, the botnet severely impacted several keyword auctions taking down companies who were formerly bidding $15 per click on terms such as "car insurance" by using up their budgets. After the companies were depleted, the botnet then monetized the traffic by selling the leads back to premium advertisers. Microsoft sought $750,000 in damages against the alleged botnet developers Eric Lam [3].

Depletion attacks can be difficult to detect because many advertisers aren't using conversion tracking and may show 0 conversion rate and lots of clicks – the exact pattern that we would normally think is fraudulent. Furthermore conversions are unreliable in any case. Some advertisers may have conversion tracking switched on but may have conversions that are exceedingly rare. In other cases, advertisers may have mis-instrumented their script, resulting in a conversion for every click.

In order to look for suspected victims, we will search for advertisers with a large number of clicks from different IPs, but a low MAC as calculated below (the SQL code implementation is in Fig. 1). A binomial test can provide evidence for how unusual is the low MAC [4]:

$$MAC(ConvRate|Advertiser = A \wedge Keyword = 'xxxx') \sim 0$$

Table I shows the results for hypothetical advertisers during a WOW Botnet style attack [3]. Advertisers A through F all have low traffic quality as measured by MAC, and the divergence for A through D is statistically significant. Advertiser A, D, E, F received a small amount of good traffic (MA Conversion Rate = 0.05%, 0.09%, 0.04%, 0.02%) but the bulk of their traffic is bad. Advertisers E and F were also attacked, but because the bot is designed to attack anyone in position 1, and they tended to be in positions 1.67 and 2.66, they were not as severely impacted by the stream of clicks from the bot.

TABLE I. ADVERTISERS BIDDING ON KEYWORD 'XXXX' DURING A HYPOTHETICAL WOW BOTNET STYLE ATTACK

Ad	*Clicks*	*Conv Rate*	*MA Conv Rate*	*Spend*	*CPC*	*Pos*	*p-value*
A	4645	0.00%	0.05%	$66,506.94	$14.32	1.01	0.000
B	2845	0.00%	0.00%	$49,139.69	$17.27	1.01	0.000
C	2822	0.00%	0.00%	$28,878.03	$10.23	1.01	0.000
D	1062	0.47%	0.09%	$10,554.94	$9.94	1.01	0.000
E	156	0.00%	0.04%	$1,177.93	$7.55	1.67	0.069
F	83	0.00%	0.02%	$432.66	$5.21	2.66	0.241
G	24	0.00%	4.20%	$342.89	$14.29	1.71	0.275
H	28	0.00%	5.65%	$902.17	$32.22	2.29	0.299

B. Which Advertisers are Perpetrating a Depletion attack?

It is also possible to try to find advertisers who may be the source of a depletion attack. The method would involve finding an advertiser for which their Mix Adjusted Conversion Rate is high, where-as the Mix Adjusted Conversion Rate of other advertisers in their same auction is low.

In all cases save one in Table I, the conversion rate from advertisers is zero because advertisers aren't using conversion tracking (D is the exception). This would present an insurmountable challenge were conversion rate relied upon for detection. Instead, Mix Adjusted Conversion Rate reveals that all 6 top click advertisers (A..F) are being attacked with large quantities of value-less traffic.

The bottom two advertisers, G and H, are actually accounts owned by the WOW botnet attacker. They have only a small number of clicks and also aren't using conversion tracking so traffic quality isn't visible directly. However Mix Adjusted Conversion Rate reveals a very healthy latent conversion rate (4.20% and 5.65% respectively).

C. Conversion Fraud Detection

Pay Per Acquisition is generally lauded as one of the best methods for minimizing fraud for advertisers [6]. Some time ago we were working an affiliate who was using their own

conversion tracking system with a Pay Per Acquisition billing model.

A simple analysis of this advertiser's conversion data led to a counter-intuitive result. Normally as a visitor takes more time on a web page, their conversion rate increases (Fig. 3). For example, traffic that spends x seconds or greater on a website might have a conversion rate of 0.3%. Traffic that spends 2x seconds or greater might have a conversion rate of 0.5%, and so on. At this particular affiliate we were seeing the opposite result. For very brief dwell times, conversion rate was highest, and then decreased as the traffic spent more time on the page (Fig. 2). Somehow the more time visitors took reading content on the website, the less they were likely to move to the next stage and convert.

This result gave us pause. Not only was the affiliate using a completely different conversion tracking system, but it was behaving differently to what we expected.

In order to get to the bottom of the mystery, we performed Mix Adjustment Analysis to measure the expected conversion rate of the IPs that were transacting on their web site at different durations on the web page:

$$MAC(ConvRate|Affiliate = A \wedge Duration = d)$$

We discovered that on Mix Adjusted quality, the affiliate's traffic actually had exactly the relationship that we expected to see – given longer dwell times on their website, Mix Adjusted Conversion Rate increased (Fig. 3). More importantly, the traffic that the affiliate was reporting to be high converting using their conversion signal was revealed to have almost no Mix Adjusted Conversion value at all (Fig. 2 and Fig. 3).

We later found that the affiliate was the victim of a massive conversion fraud operation, which was being run in order to generate billing events. Mix Adjustment penetrated the "conversion chaff" and revealed the underlying traffic quality.

VI. Conclusion

Mix adjustment is a useful method for scoring traffic that may be subject to either active spoofing, fraudulent behavior, or may be heterogeneous in terms of definition or implementation. As long as enough proxies can be utilized, and the proxies are prolific and well mixed, the method can be surprisingly effective at de-cloaking spoofed variables and exposing the underlying traffic quality. The technique also has useful non-fraud applications such as Smartpricing [5] where conversion rate differences also create significant challenges for valuing the traffic and calculating appropriate price adjustments.

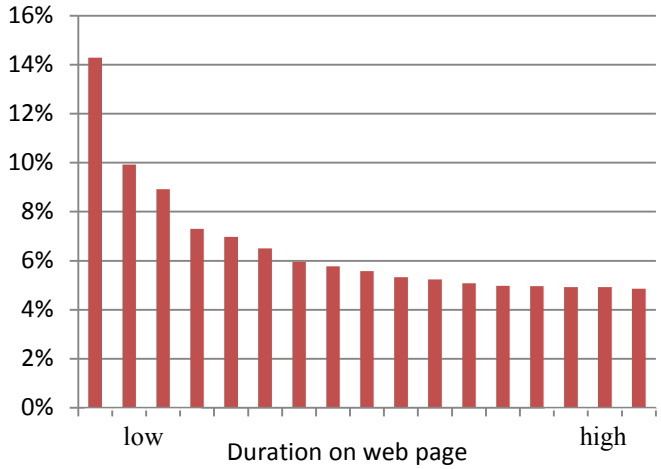

Fig. 2. Conversion Rate (y-axis) reported by affiliate versus duration on website (x-axis). The affiliate's traffic showed highest Conversion Rate for traffic that spent little time on the website. This was unusual.

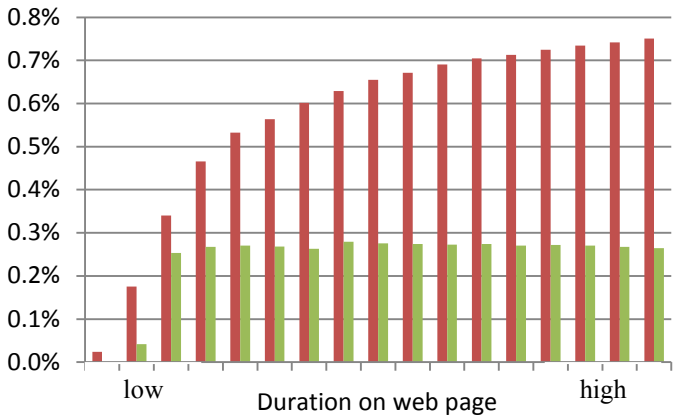

Fig. 3. Mix Adjusted Conversion Rate (y-axis) versus duration on web page (x-axis). High bars are standard quality traffic; low bars are traffic from the affiliate in question. Mix Adjusted Conversion Rate analysis revealed that short duration traffic was being fraudulently generated.

References

[1] S. Clifford, Microsoft Sues Three in Click-Fraud Scheme, NYTimes.com, June 15, 2009, http://www.nytimes.com/2009/06/16/business/media/16adco.html, 2009.

[2] T. Cranton, Using Enforcement to Crack Down on 'Click Fraud', Microsoft on the issues website, http://microsoftontheissues.com/cs/blogs/mscorp/archive/2009/06/15/using-enforcement-to-crack-down-on-click-fraud.aspx, 2009.

[3] Microsoft, Microsoft Corporation vs Eric Lam et. al., Civil Case No. C09-0815, Complaint for Injunctive Relief and Damages, United States District Court Western Ditrict of Washington at Seattle, June 2009.

[4] W. Press, S. Teukolsky, W. Vetterling, B. Flannery, Numerical Recipes: The Art of Scientific Computing, NY: Cambridge University Press, 2007.

[5] Google AdSense, About Smartpricing, http://support.google.com/adsense/bin/answer.py?hl=en&answer=190436, 2013.

[6] B. Jansen and T. Mullen, Sponsored search, International Journal of Electronic Business, Vol. 6, No. 2, 2008.

[7] M. Fichman and J. Cummings, Multiple Imputation for Missing Data, Tepper School of Business. Paper 113, 2003.