Auto-Detection of Click-Frauds Using Machine Learning

Anshuman Dash
MBA in Business Analytics
REVA Academy for Corporate Excellence
REVA University
Bengaluru, India
anshumand.ba03@reva.edu.in

Satyajit Pal
MBA in Business Analytics
REVA Academy for Corporate Excellence
REVA University
Bengaluru, India
satyajitp.ba03@reva.edu.in

***Abstract*— Fraud is increasing the risk of online marketing, ads and e-business in existing web advertising activities. Online advertising has become one of the leading funding models for websites. Since large amounts of money are involved in online commercials, unfortunately, malicious parties try to gain unfair advantage.**

Even if the online advertisers make permanent efforts to improve the traffic filtering techniques, they are still looking for the best protection methods to detect click-frauds. Click-Fraud occurs by intentional clicking of online advertisements with no actual interest in the advertised product or service. Click Fraud is an important threat to advertisement world that affects the revenue and trust of the advertisers also. Click-fraud attacks are one instance of such malicious behavior, where software imitates a human clicking on an advertisement link. Hence, an effective fraud detection algorithm is essential for online advertising businesses.

The purpose of our paper is to identify the precision of one of the modern machine learning algorithms in order to detect the click fraud in online environment. In this paper, we have studied click patterns over a dataset that handles millions of clicks over few days. The main goal was to assess the journey of a user's click across their portfolio and flag IP addresses who produce lots of clicks, but never end up in installing apps. We have focused on the issue while using various single and ensemble-typed classification algorithms for the fraud detection task. As single classifiers, we employed the Support Vector Machine, kNN algorithms. We have also employed decision tree-based ensemble classifiers, which have been used in data mining. These algorithms are Random Forest and Gradient Tree Boosting.

Keywords—***Click-Fraud Detection, Advertisements, Internet Spammers, Machine learning, Ensemble Models***

I. INTRODUCTION

Online marketing has exposed the world to everyone. Where small companies were struggling to impact in the local areas once, now-a-days the world has become very small while using the concepts of pay per click and digital marketing tools [1].

More than "4 billion people use internet on daily basis and more than 2 billion people" use internet for shopping online. A targeted pay per click campaign is the difference between sinking and swimming as more than 5 billion clicks happen in Google every day.

But there are always more than a few rats in any busy marketplace. Click fraud is one of the most harmful and successful practices in the online marketplace[2]. This technique works by manipulating your PPC campaigns, causing you to lose money, miss valuable sales opportunities, and possibly even destroy your business [3].

There is an entire industry that has been set up to defraud web marketers and consumers. Some mischievous ones, such as hackers; some created for the profit of another group fraudulently, some deliberately vindictive and with the intention of stealing ads from certain networks.

By default, click fraud does not produce an advertiser's profits, but losses "hundreds of millions of dollars" a year to "tens of thousands" of online advertisers [4]. Normally, malicious applications (apps) and malware produce click fraud and account for about "30% of click traffic in ad networks".

The number of click frauds has increased significantly with mobile malware. Fraudsters obviously create legitimate apps or buy respectable men [5]. Such applications perform a legitimate operation, like torch control, but also function as a tool to undermine the clicking behaviour of the user of the computer. In addition, attackers laundered clicks again via their installed user base [6]. As click fraud is based on valid traces, ad-network filters may pass through the clicks. Exclude the use of a small pool of IP addresses to execute the attack. The attack violates a threshold, for example. This ultimately leads to the need for automated techniques for detecting click scams, thus guaranteeing the credibility of the digital advertising ecosystem [7].

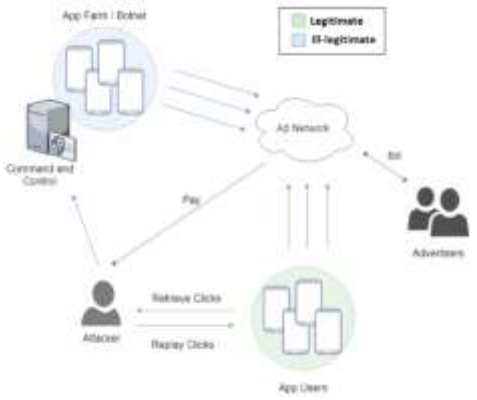


Fig-1: Legitimate & Ill-legitimate Click Fraud

A. History of Advertisements Click Business

In an online advertising market, advertisers pay ad networks for each click on their advertisements, and ad networks pay publishers a share of the revenue [8]. When online advertising has grown into a multi-billion dollar business, click fraud has become a serious and widespread problem. For example, the "Chameleon" botnet infected more than 120,000 host machines in the U.S. and siphoned $6 million a month [9].

Click fraud occurs when miscreants make HTTP requests for destination URLs found in the ads being deployed. Such HTTP requests with malicious intent are called fraudulent clicks. The motive for fraudsters is to increase their own income to the detriment of other parties [10]. A fraudster is typically a publisher or an advertiser. Publishers may place excessive advertising banners on their sites and then fake clicks on the ads to get more money. Unscrupulous advertisers are clicking heavily on a competitor's advertisements in order to deplete the victim's advertising budget. Click fraud is mainly done by using click bots, recruiting human clickers, or tricking users into clicking ads [11].

Fig – 2: Advertisements Click Business
(https://www.digitalvidya.com/blog/what-is-ppc/)

[Click fraud is not trivial. Click fraud systems have been growing continuously in recent years [12–15]. Existing detection approaches aim to classify click fraud behaviours from different perspectives, but each has its own limitations. The solutions suggested in [16–19] conduct a traffic analysis on ad network traffic logs to detect publisher inflation fraud. Nonetheless, an advanced click bot can perform a low-noise attack, which makes these unusual behavioural detection mechanisms less successful.]

B. Examples Of Click-Fraud Attacks

Major search engines such as Google and Bing are aware of how serious click fraud detection is. Back in 2005, Lane's Gifts & Collectibles sued Google along with Yahoo! and Time Warner in a collective action case resulting in $90 million settlement with an agreement to improve their tracking and identification of fraudulent clicks [20]. While things have definitely improved in the past 10 years, every PPC advertiser–or ad network–probably thinks that the problem is gone. Detecting search engine click fraud such as Google and Bing means that the big money makers in the industry secure their advertisers and the entire network.

C. Purpose

The goal of this project is to build an adaptive and scalable feature for Rich in fraud detection. This component is able to deal with the large quantity of data that is downgraded via the system and to provide output to improve the accuracy of the reports produced.

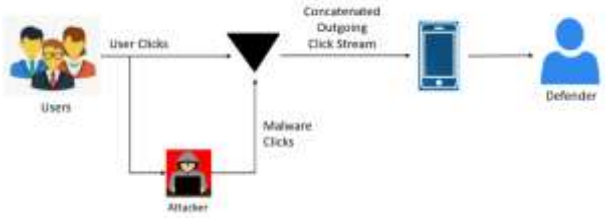


Fig-3: Click-Fraud Detection Problem

D. Overview

The paper provides major application of "machine learning" and "data mining" to solve real-world issues of fraud detection as valuable resources for industry and researchers. So far, the "data mining / machine learning" approach to fraud detection in ads has not been thoroughly studied.

This research includes university-based data, which are collected over 1 month and present many data mining and machine learning algorithms with a difficult problem [21]. The solutions presented in this report answer some important questions in data mining and machine-learning science, including a highly imbalanced output variable distribution, heterogeneous data (mixing number and class variables) and noisy patterns of missing / unknown values.

The analysis and feature engineering of exploratory data were shown to be crucial milestones for the detection of fraud. In general, there has been a systematic study of spatial and temporal factors at various granularity rates leading to the creation of nice, predictive characteristics to detect specific fraud [22].

A wide range of algorithms for single and ensemble learning have been tested in the detection of fraud, with a significant improvement over the single algorithms [23]. Coupling ensemble learning with evaluation of the feature rating often shows the key features to differentiate fraudulent from ordinary.

In this paper, the overview of the captured dataset, challenges, and evaluation procedures have been presented.

II. THEORY

A. Terms & Concepts

- *Click-through And Click-Through Rate:* CTR stands for the click-through level of Internet marketing: a measure calculating the number of click-throughs that advertisers earn per experience. Achieving a high click rate is crucial for the success of Pay-Per-Click, as it affects both value and compensation at any time anyone clicks on an ad request [24]. The rate at which your PayPal-Click advertisements are clicked is the click-through rate. This number represents the proportion of people who watch announcements (impressions) and then click on the ad. The click rate can usually be viewed on the PPC account dashboard. This is the formula for CTR:

(Total Clicks on Ad) / (Total Impressions) = Click Through Rate

Fig-4: Click Through Rate
(https://www.wordstream.com/click-through-rate)

- *Pay-Per-Click:* This marketing is an advertisement network in which advertisers do not pay for printing or ad positioning alone [25]. The bid can impact positioning, but only when an advertiser clicks on an online client. The advertiser charges. On search results pages of search engines such as Google and Bing the most popular PPC ad format appeared. Advertisers may position their brand, product or service in the form of an ad to a specific keyword or behavior [26].

- *Google AdWords:* It is Google's advertising service for companies wishing to show ads on the Google network. The AdWords program allows businesses to set an advertising budget and charge only by clicking on the ads [27]. The ad network concentrates mainly on keywords. Corporate users of AdWords may build advertisements with keywords that will be used by people searching the Internet through the Google search engine. The keyword will show your ad when it is checked. AdWords in the top marketing headings that appear on the right or above Google search results under the heading "Sponsored Links." Google search users are then forwarded to your website if your AdWords ad is clicked upon.

- *Click Fraud:* It is an unethical practice when individuals click an ad from a page (banner advertising or paid text links) to increase the number of clicks payable to the advertiser. Click fraud is an illegal practice. Illegal clicks can either be achieved by clicking on advertisement hyperlinks by someone manually or by using automated software or programmed on-line bots to click on those banner ads to pay for text ad links per click. Research has shown that clicking fraud is committed by persons using click fraud to maximize personal banner ad profits, and businesses using click fraud to deplete the budget of a competitor's publicity. Pay-per-click ads (PPC) is commonly associated with click fraud [28].

- *Impression Fraud:* It is when an ad cannot be seen in the eye, but it still takes account of experiences. Pixel filling, ad stacking and fraudulent traffic are the most common fraudulent methods. Nevertheless, malware may also happen in mobile and fraud-creating websites.

Fig-5: Impression Fraud
(https://blog.anura.io/blog/what-is-impression-fraud-and-how-does-it-work)

- *Click Bot:* It is a traffic bot breed that aims at spiking the ad count. Whenever a fraud ad is in operation, a click bot normally belongs to the crime scene. This attracts a real user who visits the site and clicks on an ad. From here, it is easy to see how the PPC project could make money bleed from this tiny piece of software [29].

III. MACHINE LEARNING ALGORITHMS

A. *Classification Approach*

First, the overview of the machine learning field is discussed, followed by the description of difference between unsupervised and supervised classification and relevant methods. These methods include outperforms K-Nearest Neighbors classification, Classification Trees, Support Vector Machine, Random Forest and Gradient Tree Boosting, in terms of accuracy rate, recall rate and precision rate. The rapid development of data mining techniques and methods resulted in Machine Learning forming a separate field of Computer Science. The basic idea of any machine learning task is to train the model, based on some algorithm, to perform a certain task: classification, cauterization, regression, etc. Training is done based on the input dataset, and the model that is built is subsequently used to make predictions. The output of such model depends on the initial task and the implementation.

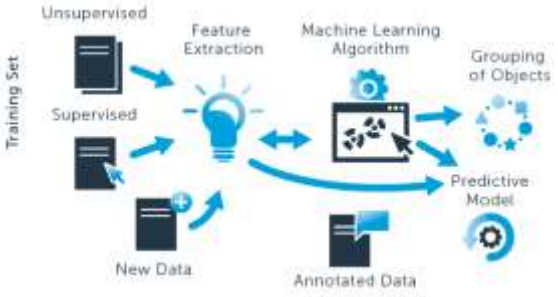


Fig-6: General Workflow of The Machine Learning Process
(https://tibacademy.in/machine-learning-training-in-marathahalli/)

B. *Supervised and Unsupervised Learning*

There are two approaches to machine learning-supervised and unsupervised learning. Learning is based on labelled data in supervised training. There is an initial dataset in this case, in which data samples are mapped to the correct result. On this dataset the model is trained, where "the correct results are known." Unlike Supervised Learning, there is no initial data labelling in Unsupervised Learning. Instead of predicting a

certain value, the aim is to find some pattern in a set of unsorted data.

C. Classification Methods

The question of classification or cauterization can be seen from a machine learning point of view: unidentified click-fraud forms are cautioned into several clusters based on specific algorithmic characteristics. On the other hand, we can reduce this problem to classification after training a model with the large dataset of malicious and benign files. This issue can be reduced for known click-fraud types to classifications with only a limited group of classes, which certainly include the click-fraud model, which can be used more easily to identify the correct class, and result is more accurate than with cauterization algorithms.

D. K-Nearest Neighbors

[K-Nearest Neighbours (KNN) is one of the simplest, though, accurate machine learning algorithms. KNN is a non-parametric algorithm which means that the data structure is not assumed. For classification problems as well as regression problems, KNN can be used. The prediction in both cases is based on the instances of k training which are closest to the input example. The result would be a group, to which the input instance belongs, foreseen by the majority of the votes of k closest neight.]

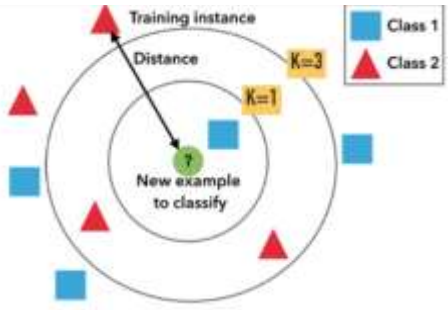


Fig-7: KNN Example
(https://medium.com/@adi.bronshtein/a-quick-introduction-to-k-nearest-neighbors-algorithm-62214cea29c7)

$$Hamming\ Distance:\ d_{ij} = \sum_{k=1}^{p} |x_{ik} - x_{jk}|$$

$$Manhattan\ Distance:\ d_1(p,q) = ||p-q||_1 = \sum_{i=1}^{n} |p_i - q_i|$$

$$Minkowski\ Distance = \left(\sum_{i=1}^{n} |x_i - y_i|^p\right)^{1/p}$$

The most used method for continuous variables is generally the Euclidean Distance, which is defined by the formulae below:

$$EuclidianDistance = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}\ ;\ p\ and\ q\ are\ the\ points\ in\ n-space$$

[Euclidian distance is good for the problems, where the features are of the same type. For the features of different types, it is advised to use. The value of k plays a crucial role in the prediction accuracy of the algorithm. However, selecting the k value is a non-trivial task. Smaller values of k will most likely result in lower accuracy, especially in the datasets with much noise, since every instance of the training set now has a higher weight during the decision process. As a general approach, it is advised to select k using the formula below]:

$$k = \sqrt{n}$$

E. Support Vector Machines:

Support Vector Machines (SVM) is another machine learning algorithm that is generally used for classification problems. The main idea relies on finding such a hyperplane, that would separate the classes in the best way. The term 'support vectors' refers to the points lying closest to the hyperplane, that would change the hyperplane position if removed. The distance between the support vector and the hyperplane is referred to as margin.

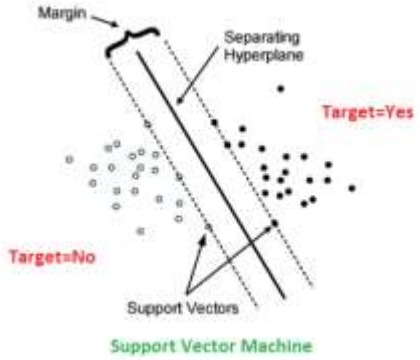


Fig-8: SVM Example
(http://www.1proga.info/support-vector-machine-algorithm-389128593a10a109ec0dd4/)

In the above figure, there is a dataset of two classes. Therefore, the problem lies in a two-dimensional space, and a hyperplane is represented as a line. In general, hyperplane can take as many dimensions as we want.

The algorithm can be described as follows:

- We define X and Y as the input and output sets respectively. $(x1,y1),\ldots,(xm,ym)$ is the training set.
- Given x, we want to be able to predict y. We can refer to this problem as to learning the classifier y=f(x, a), where a is the parameter of the classification function.
- F(x, a) can be learned by minimizing the training error of the function that learns on training data. Here, L is the loss function, and *Remp* is referred to as empirical risk.

$$R_{emp}(a) = \frac{1}{m}\sum_{i=1}^{m} l(f(x_i,a),y_i) = Training\ Error$$

We are aiming at minimizing the overall risk, too. Here, P(x,y) is the joint distribution function of x and y.

$$R(a) = \int l(f(x,a),y)dP(x,y) = Test\ Error$$

We want to minimize the Training Error + Complexity term. So, we choose the set of hyperplanes, so f(x) = (w·x)+b:

$$\frac{1}{m}\sum_{i=1}^{m} l(w\cdot x_i + b, y_i) + ||w||^2 \text{ subject to } min_i|w\cdot x_i| = 1$$

SVMs are generally able to result in good accuracy, especially on" clean" datasets. Moreover, it is good with working with the high-dimensional datasets, also when the number of dimensions is higher than the number of the samples. However, for large datasets with a lot of noise or overlapping classes, it can be more effective.

F. Random Forest

Random Forest is one of the most popular machine learning algorithms. It requires almost no data preparation and modelling but usually results in accurate results. More specifically, Random Forests are the collections of decision trees, producing a better prediction accuracy. That is why it is called a 'forest' – it is basically a set of decision trees. The basic idea is to grow multiple decision trees based on the independent subsets of the dataset. At each node, n variables out of the feature set are selected randomly, and the best split on these variables is found.

- Multiple trees are built roughly on the two third of the training data (62.3%). Data is chosen randomly.
- Several predictor variables are randomly selected out of all the predictor variables. Then, the best split on these selected variables is used to split the node. By default, the amount of the selected variables is the square root of the total number of all predictors for classification, and it is constant for all trees.
- Using the rest of the data, the misclassification rate is calculated. The total error rate is calculated as the overall out-of-bag error rate.
- Each trained tree gives its own classification result, giving its own" vote". The class that received the most" votes" is chosen as the result.

F. Classification Tree

For classification of instances, a decision tree is a simple representation. It is a supervised learning machine in which the information are separated continuously by a certain parameter.

A decision tree is a method used to support decision-making that uses a tree-like graph or template of decisions, including chance outcomes, the value of resources and utility. The decision tree is a flowchart structure in which each inner node is a "test" in a particular attribute (e.g. if the coin pad appears on the heads or tails). Every branch is the outcome of the test and every leaf node is a class tag (decision made after all attributes have been computerized). The root-to-leaf paths are category rules. One of the popular and mostly used supervised learning methods is trees-based learning algorithms. Tree-based methods allow high accuracy, stability and easy analysis of predictive models. They map non-linear relations rather well, unlike linear modelling. These are ideal for the resolution of any question (classification or regression). CART (Classification and Regression Trees) algorithms for Decision Tree.

Classification tree is a predictive model that maps an item's observations to its final value conclusions. The leaves are classifications of the tree structures (also called label), features are non-leaf nodes, and branches represent conjunctions of features leading to classifications [30]. It is simple to build a decision tree that fits a particular data set. The goal is to build good decision-making bodies, usually the smallest decision-making bodies. Overfitting can be used to avoid overfitting the tree for the training set only. This technique produces the tree for unmarked data and can accommodate some erroneously labelled training data.

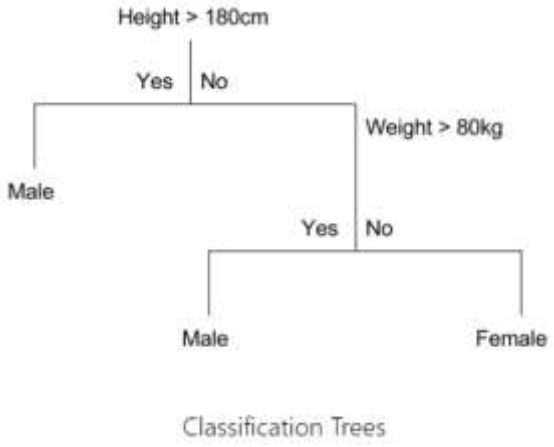


Fig-9: Classification Tree
(https://www.digitalvidya.com/blog/classification-and-regression-trees/)

G. Gradient Tree Boosting

From the application of boosting methods to regressions trees the algorithm for boosting trees was created. The general idea is to measure a series of simple trees, in which each subsequent tree is built to estimate the remains of the previous tree. This approach constructs binary trees, i.e., divides data into two samples at every divided node. At every step in the boost (algorithm for boosting trees), the data are simply (best) partitioned and variations in the observed values (residuals for each partition) measured from the respective means are calculated. In order to find another partition that reduces the rest (error) variance for the results, given the previous trees sequence, the next three node tree will then be fitted to these rest products [31].

It is shown that such "additive weighted expansions," even though the specific nature of the relationships between the predictor variables and the dependent interest variable is very complicated (not linear in nature), can eventually lead to an outstanding match of the expected values to the observed values. Therefore, a very common and efficient learning process is the gradient boosting approach – the adaptation of a weighted, additive distribution of simple trees.

IV. USE CASE

A. Dataset

The click information containing both valid and fraudulent click spam has been identified. First, it acquired a pre-label data set, consisting in controlled proportions both of legitimate clicks and of fraudulent click spam. In order to achieve this, traffic click spam has been processed within the university network; it has been filtered and distributed to test beds. As a consequence, clicks from both true and false clicks comprise the traffic leaving the Testbed.

B. Dataset Collection

The traffic monitors on backbone routers of the campus university network were set up to collect legal ad-click files. The following information was recorded in the application for each click: the URL, the IP address of the ad server, the publishing page (referrer URL), the IP address of source, the User agent string, and the time stamp. In addition, between August-2019 and October-2019, a total of 32,119 unique clicks were registered. Data was collected and all stored data

were encrypted following the due process of receiving ethical approval.

C. Data Preparation

The data is prepared for effective analyses after data collection. The data set obtained consists of several attributes which are not required for study, so the data should be prepared according to the requirements so that the algorithm produces accurate results. Data is prepared by the data.table kit and fusion method in this research work. The data.table kit is supported with a data.frame upgrade version. This allows the user to manipulate data extremely quickly, and is commonly used for large data sets [32]. Merge function allows two databases to be merged by calling the data.frame method based on common columns or row names. When columns have been defined, names of columns are given by.x (first file column names) and by.y (second file column names) [33]. Next, the original data set is charged and, by setting the Order Date and Product ID, the number of occurrences per velocity parameter is determined. Then the output of all events of every velocity variable is combined by the merge function with the original data collection.

D. Metrics & Cross Validation

The following four common performance indicators for click traffics detection are used in this research paper:

- [True positive (TP): indicates that a click traffic is correctly predicted as a fraudulent ad.]
- [True negative (TN): indicates that a click traffic is detected as a legitimate ad correctly.]
- [False positive (FP): indicates that a click traffic is mistakenly detected as a fraudulent ad.]
- [False negative (FN): indicates that a click traffic is not detected and labelled as a legitimate ad.]

The effectiveness of our proposed methods are evaluated by using machine learning performance evaluation metrics which are "Accuracy, Recall, Precision and AUC". Accuracy is defined as the number of samples that a classifier can correctly detect, divided by the addition of number of all ransomware and good ware applications.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Recall value or the detection rate is the ratio of ransomware samples that are correctly predicted

$$Recall = \frac{TP}{TP + FN}$$

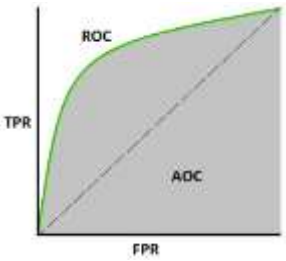
Precision is the calculated ratio of predicted ransomware that are correctly identified as a malware. Precision is defined below

$$Precision = \frac{TP}{TP + FP}$$

["AUC (Area Under the Curve) represents the probability that a true positive is positioned to the right of a true negative."] AUC ranges in values from 0 to 1. A model which predicts 100% wrong values has an AUC of 0.0 and one which predicts 100% correct values has an AUC of 1.0.

E. Performance of Algorithms

The leave-one-out technique for cross validation is used in this research paper. Below figure illustrate the network traffic usage graph due to fraudulent ads.



F. Exploratory Data Analysis

The following are the specifics of the button history and fraud tap. In this case, however, the data collection time is too short to display trends. So the attribute hour or minute is not here extracted from the time function of the click. The dataset is therefore distributed without regard to bias.

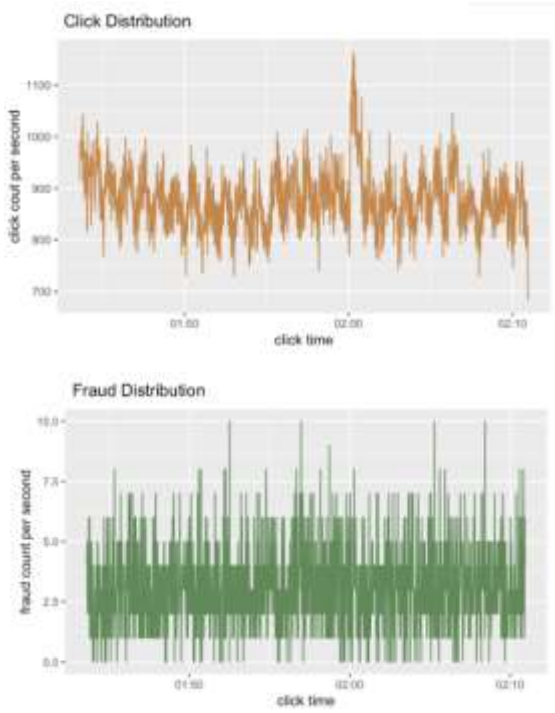


Fig-10: Dataset Distribution

The fraudulent versus non-fraudulent rate of traffic is measured as a fraudulent versus non-fraudulent proportion. Filtration speed x-axis is time, y-axis is ratio and in the time series described above indexed to ratio on the first date. The numbers show major releases of the material.

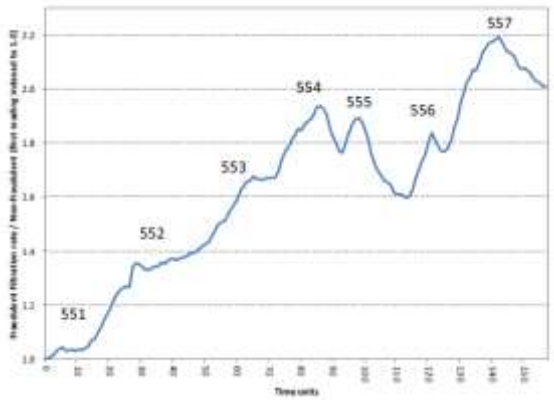


Fig-11: Fraudulent versus Non-Fraudulent Rate

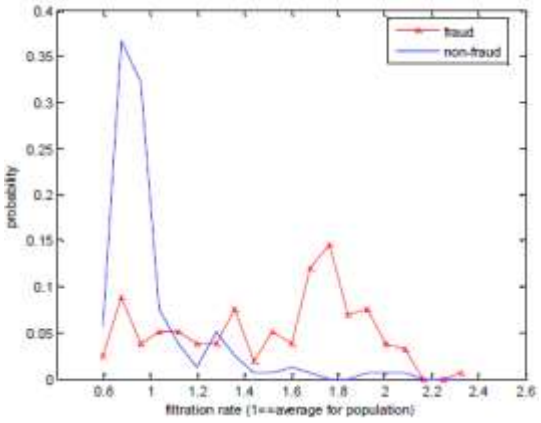


Fig-12: Filtration Rate Ratio for fraudulent versus non-fraudulent click spams

[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.]

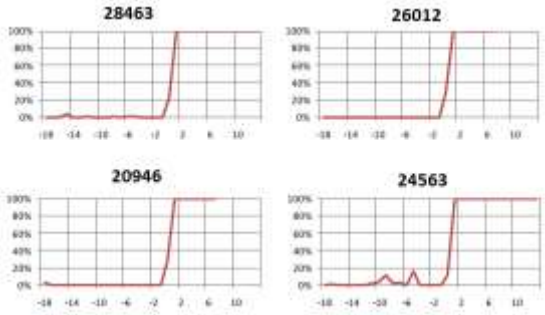


Fig-13: Filtration Rates for Four Fraudulent Click Spam

G. Evaluation

The three-phase analysis of classifier implementations has been done in order to determine the suitability of the various classification approaches for this application scenario. This was also done to address the absence of open classification studies in the area of the issue. In this section the results of the assessment are presented. During the First Evaluation Phase, one candidate algorithm was evaluated from each approach to determine its exactness (i.e., percentage of properly classified cases) on a small number of prelabelled data. A brief choice was made of candidates who ran the majority of the available classificatory and weakened those who produced too poor results (note that not all classification systems are relevant to the type of data with which we operate, e.g. those which require strict nominal input).

The analysis was performed by partitioning a collection of pre-labelling data into two separate sets used for classification learning, whether false or valid, and by assessing the effects of classifies on the pre-labelling data respectively. Rather than doing a simple percentage split, the test results were improved with a so-called n-fold cross-validation technique. A model is built with the same size n-1 partitions in the data set in n-fold cross-validation. On the remaining partition, the template is then evaluated. It is repeated n times, until each partition is used exactly once for evaluation. Listing 3 explains the cross validation algorithm. n=10 has been used for the experiments shown below.

Require: A set D of data points prelabelled with a class
$P = \{p_1, p_2, \ldots, p_n\}$, a set of equally sized partitions of D
for $i = 1$ to n **do**
$S = \{p_i\}$
$T = D \setminus S$
Build a classifier c using T as the training set
Let r_i be the result of evaluating c on test data S
end for
return The average of all results $\{r_1, r_2, \ldots, r_n\}$

There are some definitions used in the evaluation of this study before the results are reported. In the following text, a positive is the equivalent of a fraudulent instance, while a negative refers to a non-fraudulent example. A true positive is a positive statement, while a false positive is a negative that the classification evidence has been made positive. Likewise, a true negative is an advertised negative, but a false negative is a positive, which is marketed as a negative. The words used are as follows:

$$TPR\ (\text{True Positive Rate}) = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}}$$

$$FPR\ (\text{False Positive Rate}) = \frac{\text{false positives}}{\text{true negatives} + \text{false positives}}$$

$$TNR\ (\text{True Negative Rate}) = \frac{\text{true negatives}}{\text{true negatives} + \text{false positives}}$$

$$FNR\ (\text{False Negative Rate}) = \frac{\text{false negatives}}{\text{true positives} + \text{false negatives}}$$

$$ACC\ (\text{Accuracy}) = \frac{\text{true positives} + \text{true negatives}}{\text{all instances}}$$

	TPR	FPR	TNR	FNR	ACC	AROC
Random Forest	95.40%	14.30%	85.70%	4.60%	89.40%	0.959
Classication Trees	94.40%	7.60%	92.40%	5.60%	93.20%	0.975
Support vector machines	95.40%	9.10%	90.90%	4.60%	92.70%	0.962
knn Classification	69.00%	69.00%	31.00%	31.00%	45.70%	0.975
Gradient Tree Boosting	2.80%	76.90%	23.10%	97.20%	15.20%	0.975

Data set size: 32119 instances (8713 fraudulent, 23406 legitimate)

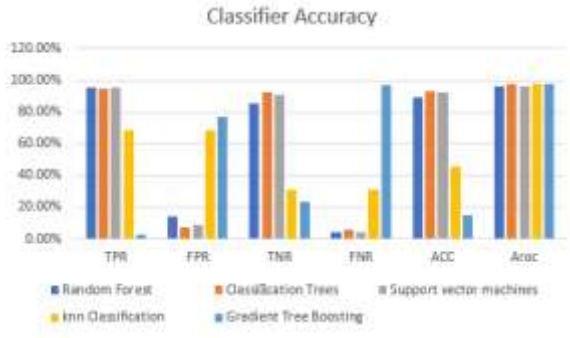


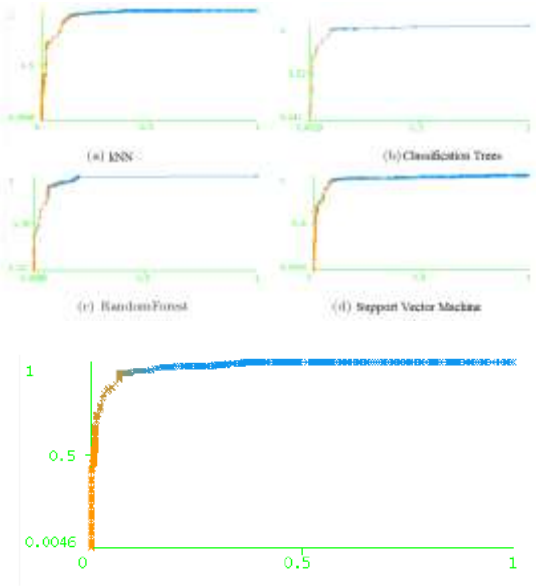
Fig-14: Classifier Accuracy

A low FPR does not necessarily imply a satisfactory outcome, as the FPR must be taken into account in relation to the actual data positive part. If the actual positive percentage in comparison with the FPR is relatively low, many of the

recorded positive warnings can be assumed to be incorrect. For example, it should be presumed that there is a data set of 10,000 users, 100 of whom actually showed an act of fraud (PAP= 1%) and the other 9.900 users show no fraud. The process reports correctly on average0:944 by means of the Random Forest steps. 100= users 94:4, and 0:076 wrongly. As fraudulent, 9900= 752:4 clients. As can be seen, because of the low portion of actual positive data, the number of false-classified positive. Positive elements are significantly higher than that of the correctly classified positives. Thereby, 88:9 percent of all positive reports are false alarm! It can be inferred! More specifically, the following formula can describe the portion of all positive reports that are false alarms:

$$PFA = \frac{\text{false alarms}}{\text{reported positives}} = \frac{FPR \cdot (1 - PAP)}{PAP \cdot TPR + FPR \cdot (1 - PAP)}$$

This is an overall difficulty in detecting fraud. But it's not entirely lost. It should be possible to approach more satisfactory results by rigorously tweaking the parameters of each algorithm. The accuracy of the training data should also be improved when optimized. Because there is a larger number of points in the "black" region between the two categories for the studies, a lower FPR can be predicted when the real data is graded. The PFA is still a concern, however, provided that real data includes signed instances that are considerably less fake than the data set used in those studies.



(e) Gradient Tree Boosting

Fig-15: ROC Curves for Classification Algorithms

V. FUTURE WORK

About future improvements to the process that can be made. The adaptive character of the system means that the learning data are continually improved. Nevertheless, there are additional ways of improving identification system accuracy. In this study, we covered a wide range of classification algorithms to classify who you are [34].

Performance improvements are also available. The current system bottleneck is the move to aggregate user data as shown in the above tests. In addition, this part of the system should therefore concentrate on efforts to improve overall system performance.

We have described some ideas which have been investigated but left out because the necessary data cannot be obtained (such as the analysis of premium clicks or mouse patterns). Those characteristics, such as consumer geographical location, were not included in the existing classification process [35]. To this end, training data would need to be developed for every campaign, so that a warning flag is lifted if most viewers for an ad suddenly comes from a new location. We think these ideas should be discussed further as they may be helpful input attributes to the classification system (when information can be obtained).

VI. CONCLUSION

The financing of millions of websites and mobile apps on-line ads is a template. Digital advertising with special purpose attack methods, called click malware, is constantly targeted by criminals. An important security challenge is click fraud created via malware. The state-of - the-art techniques can easily detect static attacks involving large attack volumes. Nonetheless, current methods fail to detect complex attacks involving steady click-spam that match the app user's actions. Timing analysis has been found to have a crucial role to play in isolating click scams, both static and dynamic.

This research paper applies a technique that detects click-spam using relative uncertainty between click-spam and valid clicks-streams. It does this by identifying repeated patterns from valid click-spam in the ad network. A malware corpus is also analysed in an instrumented environment which can handle click-spam generation by exposing malware to legitimate click-spams. We have tested a passive technique that is promising. An effective protection has also been tested, wherein the analytical system is better functioning when injecting watermarked click traffic. Although timing analysis has been well studied for its ability to discover supernatural interaction in the field of data hiding, its potential still has to be fully explored when understanding fraud attacks through stealthy clicks. Our work shows that time analysis may be important in order to improve the detection of fraud by clicking.

REFERENCES

[1] M. R. Berthold, N. Cebron, F. Dill, T. R. Gabriel, T. K • otter, T. Meinl, P. Ohl, K. Thiel, and B. Wiswedel. KNIME{The Konstanz information miner: Version 2.0 and beyond. SIGKDD Explorations Newsletter, 11(1):26{31, 2009.

[2] G. E. P. Box. Non-normality and tests on variances. Biometrika, 30(3/4):318{335, 1953.

[3] L. Breiman. Bagging predictors. Machine Learning, 24:123{140, 1996.

[4] L. Breiman. Random forests. Machine Learning, 45(1):5{32, 2001.

[5] C. Chambers. Is click fraud a ticking time bomb under Google? Forbes Magazine, 2012. URL http://www.forbes.com/sites/investor/2012/06/18/is-click-fraud-a-ticking-time-bomb-under-google/.

[6] C. C. Chang and C. J. Lin. LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems Technology, 2(3):27:1{27:27, 2011.

[7] A. Chao and T. Shen. Nonparametric estimation of shannon's index of diversity when there are unseen species in sample. Environmental and Ecological Statistics, 10:429{443, 2003.

[8] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. SMOTE: Synthetic minority over-sampling technique. Journal of Articial Intelligence Research, 16:321{357, 2002.

[9] C. Chen, A. Liaw, and L. Breiman. Using random forests to learn imbalanced data. Technical report, Technical Report No. 666, Department of Statistics, University of California, Berkeley, 2004.

[10] W. Cohen. Fast effective rule induction. In Proceedings of the International Conference on Machine Learning, pages 115{123, Tahoe City, California, 1995.

[11] T. Cover. Nearest neighbor pattern classiffication. IEEE Transactions on Information Theory, 13(1):21{27, 1967.

[12] V. Dave, S. Guha, and Y. Zhang. Measuring and fingerprinting click-spam in ad networks. In ACM SIGCOMM Computer Communication Review, volume 42, pages 175{186, Helsinki, Finland, 2012.

[13] P. Domingos. MetaCost: A general method for making classiffiers cost-sensitive. In Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 155{164, 1999.

[14] R. E. Fan, K. W. Chang, C. J. Hsieh, X. R. Wang, and C. J. Lin. LIBLINEAR: A library for large linear classiffication. Journal of Machine Learning Research, 9:1871{1874, 2008.

[15] Amazon EC2 Instance Types. Retrieved March 18, 2010, from http://aws.amazon.com/ec2/instance-types/.

[16] Google AdWords Tra-c Estimator. Retrieved February 1, 2010, from https://adwords.google.com/select/TrafficEstimatorSandbox.

[17] Invalid Clicks - Google's Overall Numbers. Retrieved May 10, 2010, from http://adwords.blogspot.com/2007/02/invalid-clicks-googles-overall-numbers.html, February 2007.

[18] Apache Lucene Mahout: k-Means. Retrieved April 6, 2010, from http://cwiki.apache.org/MAHOUT/k-means.html, November 2009.

[19] Dhruba Borthakur. HDFS architecture. Retrieved April 29, 2010, from http://hadoop.apache.org/common/docs/current/hdfs_design.html, February 2010.

[20] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly Detection: A Survey. ACM Computing Surveys, 41, 2009.

[21] C.C. Chang and C.J. Lin. LIBSVM: a library for support vector machines, 2001.

[22] D. Chang, M. Chin, and C. Njo. Click Fraud Prevention and Detection. Erasmus School of Economics e Erasmus University Rotterdam, 2008.

[23] N. Daswani and M. Stoppelman. The anatomy of Clickbot. A. In Proceedings of the rst conference on First Workshop on Hot Topics in Understanding Botnets, page 11. USENIX Association, 2007.

[24] J. Dean and S. Ghemawat. Map Reduce: Simplied data processing on large clusters. Communications of the ACM-Association for Computing Machinery-CACM, 51(1):107114, 2008.

[25] Peter Eckersley. A primer on information theory and privacy. Retrieved April 28, 2010, from https://www.eff.org/deeplinks/2010/01/primer-information-theory-and-privacy, January 2010.

[26] Tristan Fletcher. Support vector machines explained. 2009. D. Foregger, J. Manuel, R. Ramirez-Padron, and M. Georgiopoulos. Kernel similarity scores for outlier detection in mixed-attribute data sets. 2009.

[27] M. Gandhi, M. Jakobsson, and J. Ratkiewicz. Badvertisements: Stealthy click-fraud with unwitting accessories. Journal of Digital Forensic Practice, 1(2):131-142, 2006.

[28] Z. He, S. Deng, X. Xu, and J. Huang. A fast greedy algorithm for outlier mining. Advances in Knowledge Discovery and Data Mining, pages 567-576, 2005.

[29] 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 communi-cations security, October 26, 2007, pp. 421 – 431 (2007)

[30] 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)

[31] Jansen, B., Flaherty, T., Baeza-Yates, R., Hunter, L., Kitts, B., Murphy, J.: The Compo-nents 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)

[32] Kantarcioglu, M., Xi, B., Clifton, C.: A Game Theoretic Approach to Adversarial Learn-ing, 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)

[33] Kitts, B.: Regression Trees, Technical Report, http://www.appliedaisystems.com/papers/RegressionTrees.doc (2000)

[34] Kitts, B. Laxminarayan, P. and LeBlanc, B.: Cooperative Strategies for Keyword Auc-tions, First International Conference on Internet Technologies and Applications, Wales. September 2005. (2005)

[35] Wellman, M., Greenwald, A., Stone, P. and Wurman, P. (2003a) 'The 2001 Trading Agent Competition', Electronic Markets 13(1): 4–12.