Advertisement Click Fraud Detection

Yameeni Mhaske[1], Vaishnavi Bhosale[2], Ankita Gupta[3]
[1, 2, 3] Dept of Information Technology
[1, 2, 3] PCE, Navi Mumbai, India - 410206

Abstract- *Click-fraud happens in pay-per-click ad networks where the ad network charges advertisers for every click on their ads. In a click-fraud attack, a user or an automated software clicks on an ad with a malicious intent and advertisers need to pay for those valueless clicks. Among many forms of click-fraud, botnets with the automated clickers are the most severe ones. In this project, we will present a method to detect such fraud clicking by developing an effective fraud detection algorithm essential for online advertising businesses. We will focus on the issue while using various single and ensemble-typed classification algorithms for the fraud detection task. The main goal is to assess the journey of a user's click across their portfolio and flag IP addresses who produce lots of clicks. We will employ a decision tree-based ensemble classifier, which will be used in data mining. This algorithm is Random Forest (If The dataset has a sufficient number of fraud examples*

Keywords- Click Fraud, Ip-Address,Number of clicks, click_time.

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. More than "4 billion people use the internet on a daily basis and more than 2 billion people" use the 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. This technique works by manipulating your PPC campaigns, causing you to lose money, miss valuable sales opportunities, and possibly even destroy your business . 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.

II. LITERATURE SURVEY

The 1st we referred to is A hybrid and effective learning approach for Click Fraud detection(2021)Authors: Thejas G.S., Surya Dheeshjith , S.S. Iyengar , N.R. Sunitha , Prajwal Badrinath.It uses the Cascaded Forest to concatenate the original dataset with additional columns and uses the XG Boost model for final classification. It uses the Cascaded Forest to concatenate the original dataset with additional columns and uses the XG Boost model for final classification. Using various click fraud detection models, we infer that CFXGB performs outstandingly well on performing comparative analysis. We also experimented with Parent node values and inferred that this parameter must be treated as a hyper parameter. Nevertheless, this model can be used as a generic model to solve other machine learning classification problems.

The 2nd paper referred to FC Fraud: Fighting Click-Fraud from the User Side(2016) Authors: Shahrear Iqbal, Mohammad Zulkernine, Fehmi Jaafar, Yuan Gu. In this paper, we develop a technique, FC Fraud, that protects innocent users by detecting the fraudulent processes that perform click-fraud silently. FC Fraud executes as a part of the operating system's anti-malware service. It inspects and analyzes web requests and mouse events from all the user processes and applies Random Forest algorithm to automatically classify the ad requests. After that, it detects fraudulent ad clicks using a number of heuristics. In our experimental evaluation, FC Fraud successfully detects all the processes running in the background and performing click fraud. It blocks them from further accessing the network thus removing the machine from the botnet. Most major operating system vendors own or operate ad networks and advertising is one of their major sources of income. As a result, we believe that adding FC Fraud to the operating system's anti-malware service can greatly serve the interests of the operating system vendors and online advertisers and it can be a valuable addition to the server-based detection techniques.

The 3rd paper referred is Detection of Advertisement Click Fraud Using Machine Learning.(2020) Authors: B. Viruthika, Suman Sangeeta Das, E Manish Kumar, D PrabhuThe percentage of advertisement click fraud is found significant. Recent statistics have proven that the cause is major and would be increasing in the future. Thus, the proposed model has been developed to detect and minimise the malwares that monetises using click fraud. It is in need as criminals get profit out of it and the problem is on a large scale. Hence, the proposed system has overcome the problem to maximum extent and provides the result accurately.

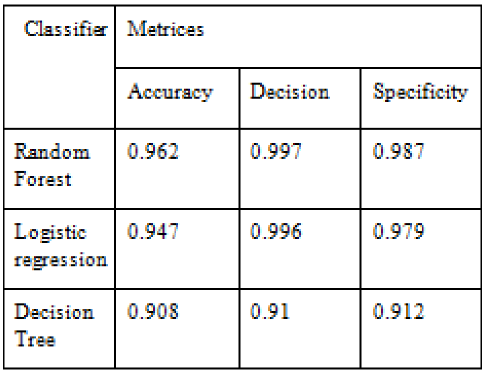
The 4th paper is A Method for Detecting Fraud Advertisement (2020) Authors: Keesara Sravanthi, Batturi Pavankalyan, Sharath

Chandra.We had done the advertising click fraud detection by using neural networks and we got a result of accuracy 91% with less false positives. We had used Gaussian naïve bayes classifiers before normalization. We got more false positives so then we had normalized the data and we tested the data so we got less number of false positives. So we conclude that the neural network is the best technique to find the frauds in advertisements.

2.1 COMPARISON

Performance of all learning algorithms used for Advertisement click fraud detection are compared in table 1. The comparison is based on their accuracy, precision and specificity.

Table 1
COMPARISON OF MACHINE LEARNING TECHNIQUES



Classifier	Metrices		
	Accuracy	Decision	Specificity
Random Forest	0.962	0.997	0.987
Logistic regression	0.947	0.996	0.979
Decision Tree	0.908	0.91	0.912

The overview of comparison of different parameters are given

III. PROPOSED WORK

We have gathered data from China’s largest independent big data service platform, covering over 70% of active mobile devices nationwide. They handle 3 billion clicks per day, of which 90% are potentially fraudulent. According to the references we are planning to implement the techniques like Logistic Regression,, random forests with decision trees, feature encoding and engineering, and sequence modeling. We are implementing Jupyter Notebook scikit Learn and a confusion matrix to sort the data. We have discussed the model of our project using block diagrams and hardware software details. If this algorithm is applied into advertisement click fraud detection systems, the probability of fraud transactions can be predicted soon after click fraud occurs. Thereafter a series of anti-fraud strategies can be adopted to prevent advertisers from great losses and reduce risks.

3.1 System Architecture

The system architecture is given in Figure 1. Each block is described in this Section.

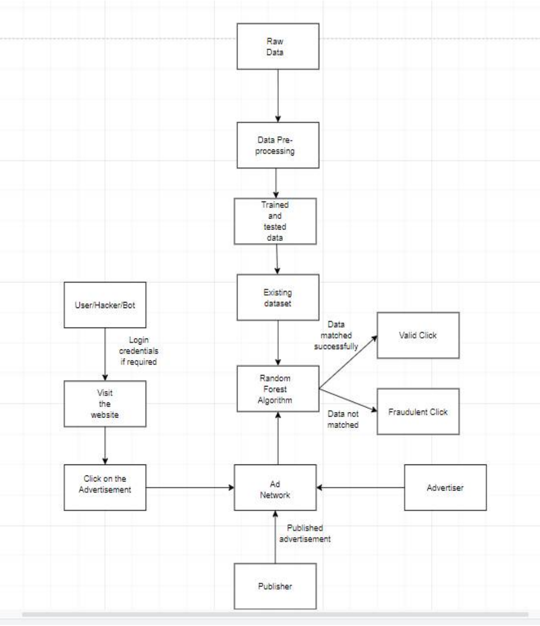


Fig. 1 Proposed system architecture

This is a block diagram for our system. It is the actual representation of the algorithm which we wish to implement. The 1st step is to read the data set & then it is sent for sampling. Training & testing of the data set is done.

After the feature selection, the data will be sent to the algorithm which is the combination of the Logistic Regression , decision tree, random forest & MCC.

The resultant data is stored in test sample data. The prediction of outcome is done Based on test sample data & the result of the combined algorithm.

Later the performance & accuracy results are plotted. It has a feature which validates the results if the click is legitimate thenthe transaction is said to be true or else it is false. In case of false transactions the bank is made aware of it.

III. REQUIREMENT ANALYSIS

The implementation detail is given in this section.

3.1 Software

1. Operating system : Windows 8/10.
2. IDE Tool : Anaconda
3. Coding Language : Python 3.6 & up
4. APIs : Numpy, Pandas ,Seaborn, Matplotlib

3.2 Hardware

1. Processor : Pentium i3 or higher.
2. RAM : 4 GB or higher.
3. Hard Disk Drive : 20 GB (free).
4. Peripheral Devices : Monitor, Mouse and Keyboard

3.3 Data Preprocessing

In this module selected data is formatted, cleaned and sampled. The dataset that we've considered is from talking data which is China's largest independent big data service platform, covering over 70% of active mobile devices nationwide. They handle 3 billion clicks per day, of which 90% are potentially fraudulent. The data preprocessing steps includes following:

a. Formatting: The data which has been selected may not be in a suitable format. The data may be in a file format and we may like it in relational databases or vice versa.
b. Cleaning: Removal or fixing of missing data is called cleaning. The dataset may contain records which may be incomplete or it may have null values. Such records need to be removed.
c. Sampling: As the number of frauds in the dataset is less than the overall transaction, class distribution is unbalanced in credit card transactions. Hence sampling method is used to solve this issue.

IV. ACKNOWLEDGMENT

We would like to thank our guide and mentor for this project Prof. Keishnendu Nair who mentored us throughout our "Advertisement Click Fraud Detection" project and cleared our concepts and helped us understand all the topics.

We would also like to thank the Head Of Department Of IT Dr. Satishkumar Varma for giving us an opportunity to understand and implement this project, which helped us a lot in understanding deep concepts of Software Designing, and how it works.

We thank our principal Dr. Sandeep Joshi for providing us with all the facilities and required equipment for this project. We would like to thank our guide and mentor for this SDL lab Prof. Smita Kadam who mentored us throughout our "Credit Card Fraud Detection" project and cleared our concepts and helped us understand all the topics.

We would also like to thank the Head Of Department Of IT Dr. Satishkumar Varma for giving us an opportunity to understand and implement this project, which helped us a lot in understanding deep concepts of Software Designing, and how it works.

We thank our principal Dr.. Sandeep Joshi for providing us with all the facilities and required equipment for this project.

V. CONCLUSION

We have considered various security vulnerabilities in the most prominent online advertising. We systematically examine the click fraud and its types. Click Fraud is already a threat to business online and has a great potential to increase your attack radius. New technologies like the Internet of Things can collaborate indirectly to that. The need to sharpen defenses against this coup is urgent. The financial losses resulting from this danger are very significant. We analyze and survey some detection techniques. used presently indifferent solution domains. Click fraud not only disturbs the budget advertisers but also how bots are used to corrupt your valuable data. Hence its important to be aware and evolving to come up with solutions to circumvent and prevent them. Defence Strategies must be further improved

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. Bio metrika, 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 supporting 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. Kegel meyer. SMOTE: Synthetic minority over-sampling technique. Journal of Artificial 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, Berk eley, 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 classification. 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.