Fake Ad Click Detection by Identifying Anomalies

Student Name :	Gautam Shanbhag
Student ID :	18210455
Email :	gautam.shanbhag2@mail.dcu.ie
Programme:	MCM (Data Analytics) Full-Time
Date of Submission :	11/Aug/19
Title :	Fake Ad Click Detection by Identifying Anomalies
Supervisor :	Mark Roantree

DISCLAIMER

A report submitted to Dublin City University, School of Computing for module CA685 Data Analytics Practicum, 2018/2019. I understand that the University regards breaches of academic integrity and plagiarism as grave and serious. I have read and understood the DCU Academic Integrity and Plagiarism Policy. I accept the penalties that may be imposed should I engage in practice or practices that breach this policy. I have identified and included the source of all facts, ideas, opinions, viewpoints of others in the assignment references. Direct quotations, paraphrasing, discussion of ideas from books, journal articles, internet sources, module text, or any other source whatsoever are acknowledged and the sources cited are identified in the assignment references. I declare that this material, which I now submit for assessment, is entirely my own work and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work. By signing this form or by submitting this material online I confirm that this assignment, or any part of it, has not been previously submitted by me or any other person for assessment on this or any other course of study. By signing this form or by submitting material for assessment online I confirm that I have read and understood DCU Academic Integrity and Plagiarism Policy (available at: http://www.dcu.ie/registry/examinations/index.shtml)

Name: Gautam Shanbhag

Date: 11/Aug/2019

Fake Ad Click Detection by Identifying Anomalies

Gautam Shanbhag
MCM Data Analytics
Dublin City University, School of Computing
Dublin, Ireland
gautam.shanbhag2@mail.dcu.ie

***Abstract*—Click Fraud is a technique of deceitfully increasing the number of clicks on a pay-per-click ad. There are two common driving factors for click fraud; one is an attempt by the advertisers to sabotage their competitors by increasing their costs and exhausting their budget caps early. Second is the practice in which Ad publishers try to generate more revenue for themselves by clicking on the ads displayed on their site. The aim of this research is to predict whether a publisher is fraud by analyzing the patterns in data using various features provided. This is achievable by measuring user click counts, flagging IP addresses that produce many clicks. Identifying key mobile agents used or specific country and ip combinations can be used for fraudulent clicks detection. Prediction of fraud publishers is performed by using various classification and machine learning models.**

Keywords—Fraud Detection, Anomalies, Pattern Recognition, Feature Engineering, Machine Learning, Decision Tree, CatBoost, Random Forest, Logistic Regression, Classification, Recursive Feature Elimination, Auc-Roc.

I. INTRODUCTION

Pay-per-click model is suffering from tremendous fraud issues. In many cases, the clickers click on advertisements without having actual interest in the ad's link. This enables them earn extra money for depleting the advertiser's budget. Click fraud refers to all such fraudulent means used for generating more clicks without being recognized by the search engine [1].

Loss estimation of Mobile app advertisers have doubled compared to 2017 with e-commerce the hardest hit category [14]. E-commerce accounted for 40% of the total installations rejected by Adjust in the 2018 report, up from 20% in 2017. The games category was last year's leader with 35%, but declined to 30% this year. A recent study from Juniper Research has forecasted that advertisers will lose around $42 billion of ad spend globally this year to deceitful activities committed via online, mobile and in-app advertising. This is a 21% increase from the $35 billion lost to advertising fraud in 2018. This growth will be driven by increasingly sophisticated techniques being implemented by fraudsters [15].

The frauds fall under two main categories:

(1) Bot-driven frauds employ bot networks or paid users to initiate fake ad impressions and clicks [2]

(2) Placement frauds manipulate visual layouts of ads to trigger ad impressions and unintentional clicks from real users (47% of user clicks are reportedly accidental [3])

Traditional search advertising click fraud was mainly triggered by an automated program. This kind of click fraud has evident characteristics, such as high traffic within a short time, single access mode, repetitive, etc. We can filter out this kind of click fraud through online detection rules [4].

A click farm is a form of click fraud, where the click fraudster hires large groups of workers to click on paid advertising links. The workers click the links, surf the target website for a certain amount of time and in some cases they also sign up for newsletters before moving to another link. In such cases, the workers can earn huge amount of money by clicking on enough ads per day to hit their target. This gives them an alternative source of income. It is extremely troublesome for an automated channel to detect such simulated traffic as fake as the user behavior appears the same as that of a genuine user [5].

The phrase 'invalid clicks' is similar to click fraud, however, in addition to clicks that are intentionally fraudulent, we also consider some accidental clicks which are generated by genuine user clicks as invalid clicks. Invalid clicks can be troublesome for advertisers. Large advertising networks analyze and validate clicks on behalf of the advertiser. In the event of detection of such invalid clicks, the networks flag such records and remove them from payments and reports [6].

The aim of this project is to examine various machine-learning algorithms for Click Fraud Detection.

The rest of the paper is organized as follows. Section II provides a background of the relevant work in this domain. In Section III, CRISP-DM approach is discussed as the research plan. Section IV discusses the approach taken for detecting Click Fraud and related work are present. This section is sub divided into dataset description, exploratory data analysis, feature extraction, steps taken to handle the dataset and preprocessing is discussed. Section V talks about the various algorithms finalized for analysis, their working and how it models our dataset. Evaluation methods such as recall, precision, accuracy and auc-roc are covered in Section VI which is used to compare and contrast between the different models used for prediction. Section VII concludes the results and understanding of the algorithm used to identify anomalies and classify publishers into real or fraud would be analyzed and identified. Finally, the paper concludes with references.

II. LITERATURE REVIEW

Recognizing Click-Fraud is a relatively new area of research. In minimalistic form, the advertisers detect frauds based on massive ad clicks crossing the set threshold. If a mobile app / web page is receiving a high number of clicks from the same IP address in a short interval, these clicks can be flagged as fraud. The detection becomes complicated if the clickers use proxy networks or global IP switching techniques. In such cases, the advertisers have to use global IP blacklists and periodically update the list as the number of bots increase, at the expense of losing income from genuine customer clicks [7].

Some research studies suggest using supervised learning methods for generic bot detection. Li et al presented a generalized botnet detection method using Particle Swarm Optimization (PSO) and K-means [8], where the authors used a dataset that contained a high proportion of bots, which is uncommon in real life. Zhao et al. [9] applied several machine-learning techniques to detect botnets of different types. However, this research used a merged dataset that was collected from different sources and contained 5.8% of malicious data. Merged dataset can be flawed, biased and may not represent the real world.

According to [10], prevention mechanisms are based on blocking suspicious traffic by IP, city, country, referrer, ISP, etc. An online database is maintained which contains these suspicious factors. This system is then tested using real world data from ad campaigns. It is concluded from the results that click fraud analysis quality can be improved using multi-level data fusion.

Antoniou D. and others in their IEEE Paper [11] on Exposing Click-fraud using a burst detection algorithm have used splay tree algorithm, which is a self-adjusting form of binary search tree by exploiting the time and space locality. They monitor the sequence of accesses to a webpage and the time span to design the splay tree.

Perera K.S. and others have worked on the same dataset which we have acquired. In their Paper [28] aggregate records over publisher information have been extracted and the novel approach for click fraud detection is evaluated using recall, precision, recall and area under the roc curve. For feature selection, Principal Component Analysis, Common Spatial Patterns and wrapper subset evaluation were tested. SMOTE and resampling were used to handle skewness in data. In their research, Bagging and Boosting were used for all the decision tree learning algorithms. Based on their results the best and most consistent method for classification was proven to be Bagging with decision tree.

In IEEE Paper [21], Taneja M. et al have used Recursive Feature Elimination (RFE) for feature selection and Hellinger Distance Decision Tree for classification. There framework was investigated on the Buzzcity dataset. The team compared J48, HDDT, logitboost and random forest. Results showed that RFE performed better than wrapper selection feature with all the classifiers except J48 and best results were given by combination of RFE + HDDT with an accuracy of 97.23%.

III. Research Plan

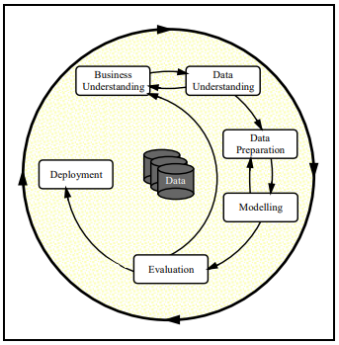

Figure 1 : Phases of the Current CRISP-DM Process Model for data Mining

In this research, we have followed Cross-industry standard process for data mining, known as CRISP-DM, which is an open standard process model that describes common approaches used by data mining experts. It is the most widely-used analytics model. CRISP-DM breaks the process of data mining into six major phases:
Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation, Deployment
The sequence of the phases is not strict and moving back and forth between different phases as it is always required. The arrows in the process diagram indicate the most important and frequent dependencies between phases. The outer circle in the diagram symbolizes the cyclic nature of data mining itself. The lessons learned during the process can trigger new, often more focused business questions, and subsequent data mining processes will benefit from the experiences of previous ones. [18]

IV. Approach

A. *Dataset Description*

For any given idea to be tested and refurbished, data forms the core of the analysis and deduction to be followed. The data set for this analysis was granted by LARC SMU-CMU Singapore Management University provided by BuzzCity [16] to analyze the behavior of publishers and classify fraudulent publishers from legitimate publishers. The dataset provided consist of 2 categories: publisher database and click database.
The publisher database records the publisher/partners profile, and consists of several fields as listed in Table 1. On the other hand, the click database captures the click traffic associated with various publishers. Table 2 lists the fields in the click database. Dataset is described by 3 distinct classes as status in publisher information ie. “OK”, “Fraud” and “Observation”.

Field	Description
`publisherid`	Unique identifier of a publisher
`bankaccount`	Bank account associated with a publisher (anonymized; may be missing/unknown)
`address`	Mailing address of a publisher (anonymized; may be missing/unknown)
`status`	Label of a publisher, which falls into three categories: • `OK`: Publishers whom BuzzCity deems as having healthy traffic (or those who slipped their detection mechanisms) • `Observation`: Publishers who may have just started their traffic or their traffic statistics deviates from system wide average. BuzzCity does not have any conclusive stand with these publishers yet • `Fraud`: Publishers who are deemed as fraudulent with clear proof. BuzzCity suspends their accounts and their earnings will not be paid

Table 1 : Fields in the Publisher database.

Field	Description
`id`	Unique identifier of a particular click
`numericip`	Public IP address of a clicker/visitor
`deviceua`	Phone model/agent used by a clicker/visitor
`publisherid`	Unique identifier of a publisher
`campaignid`	Unique identifier of a given advertisement campaign
`usercountry`	Country from which the clicker/visitor is
`clicktime`	Timestamp of a given click (in yyyy-mm-dd format)
`referredurl`	URL where ad banners are clicked (anonymized; may be missing/unknown)
`channel`	Publisher’s channel type, which consists of: • `ad`: Adult sites • `co`: Community • `es`: Entertainment and lifestyle • `gd`: Glamour and dating • `in`: Information • `mc`: Mobile content • `pp`: Premium portal • `se`: Search, portal, services

Table 2: Fields in the Click database.

B. *Exploratory data analysis and preprocessing*

Output variable consisted of 3 classes, namely ‘Ok’, ‘Fraud’ and ‘Observation’. However as the records with label ‘Observation’ were irrelevant for the research, those records were dropped from the dataset. The overall records count

after removal was 1,23,87,540 which is almost 12 million. Columns like publisherid, bankaccount, address, id, campaigned was dropped as they directly mapped to our predictor column ‘status’. Null values were cleaned by removing empty records which were present mostly in referrer and agent column thus reducing the final dataset to 69,38,523 ie 7 million with biased status column to be ‘Fraud’ – 444439 and ‘OK’ – 6494084. The following figure shows the count of unique features per attribute.

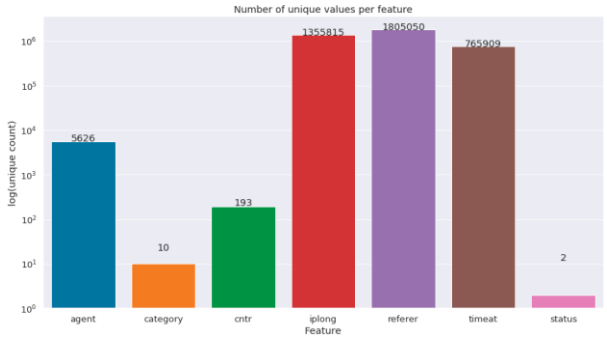

Figure 2 : Number of unique values per feature.

To identify anomalies or patterns in dataset, feature extraction is done primarily based on time feature. As per literature review it is observed that fraud clicks happen normally in burst mode. The ‘timeat’ feature is split into month, day, hour, minute, second for further processing.
Using the time features, we try to identify if there exists any hourly patterns for clicks. The below figure plots no. of clicks against ratio of fraud clicks per hour.

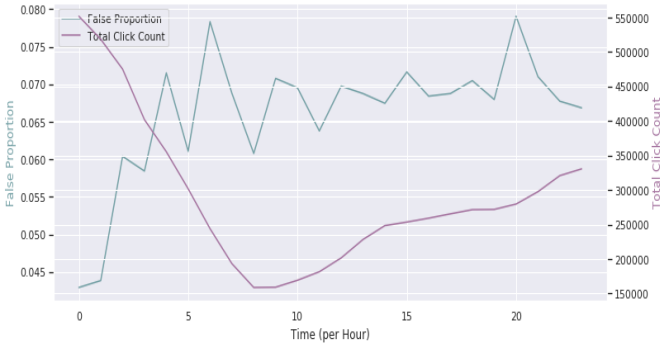

Figure 3 : Hourly Fraud Ratio to Total Click count

C. Feature extraction

Most machine learning algorithms can only read numerical values, therefore it is essential to encode categorical features into numerical values. One Hot Encoding is one of the optimum ways to deal with categorical features [19]. The complete dataset being categorical in nature had to be converted to numerical form for model to be implemented. But the presence of numerous classes in each of the features led to curse of dimensionality problem [20] and huge sparse matrix. This led to slowness in modelling plus out of memory problems. Thus encoding categorical features was avoided after initial tests.
Referring to [16] [21] feature extraction significantly improved accuracy and performance of the model.
Many parameters such as counts by each feature and then counts by grouping features such as 'ip_ref_cat_per_min', 'ip_ref_cat_per_hr' were extracted. The names of the features extracted are listed in the Table3.

Sr. No.	Type Of Features	Features Extracted
1	Time Features :	click_month click_day click_hour click_min click_sec
2	Count Features :	clicks_by_ip clicks_by_cntr clicks_by_referer clicks_by_category clicks_by_agent
3	Count feature By time :	clicks_per_minute clicks_per_hour
4	Count feature By Grouping :	ip_ref_cat_per_min ip_ref_cat_per_hr ip_cat_per_min ip_cat_per_hr ip_min_count agent_ip_count agent_cntr_count
5	Avg feature By Grouping :	ip_min_avg ip_ref_cat_min_avg ip_cat_min_avg

Table 3 : Details of Feature Explored From Given Attributes

D. Recursive Feature Elimination and UnderSampling

Multiple feature extraction from limited set of attributes can add multi collinearity. Recursive Feature elimination (RFE) algorithm is a greedy method that only hopes to find the best possible combination for classification. Although, RFE does not necessarily return an optimal solution due to its greedy nature, it is distinguished as one of the most effective methods of feature selection as the weakest feature is removed after each step rather than selecting the strongest feature. Using RFE and correlation matrix, we have selected the top 10 features which is further used for modelling.

The original categorical attributes and time components were dropped from the dataset. RFE selected features were used as final dataset. The machine learning models and algorithms to follow in the paper were applied on a training set (for building predictive model) 75% and a test set (for evaluating the models' generalization abilities) 25%. Due to imbalanced nature of dataset, the model would fail to identify minority class correctly, also the dataset being enormous, the approach of under sampling [22] was followed to balance both the classes.

E. Preprocessing

Most of the machine learning algorithms use Eucledian distance between two data points in their computations. If left alone, these algorithms only take in the magnitude of features. The features with high magnitudes will weigh in a lot more in the distance calculations than features with low

magnitudes. As in our case the count of clicks by different attributes vary thus to suppress this effect, we bring all features to the same level of magnitudes by Scaling [24].
Out of the many scaling techniques available, we have chosen the Min – Max Scaler method pre-loaded in sklearn package to scale the dataset.

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$$

This scaling brings the value between 0 and 1. The training dataset is transformed and fitted using this package and test set has been fitted on the train model created.

V. Algorithm & Implementation

A. Algorithm 1 : CatBoost Classifier

CatBoost is a machine learning algorithm that uses gradient boosting on decision trees. It is available as an open source library [17]. CatBoost comes from two words "Category" and "Boosting". The library works well with multiple Categories of data, such as audio, text, image including historical data. Gradient boosting is a powerful machine learning algorithm that is widely applied to multiple types of business challenges like fraud detection, recommendation items, forecasting and it performs better than other algorithms.

Catboost introduces two critical algorithmic advances - the implementation of ordered boosting, a permutation-driven alternative to the classic algorithm, and an innovative algorithm for processing categorical features. Both techniques are using random permutations of the training examples to fight the prediction shift caused by a special kind of target leakage present in all existing implementations of gradient boosting algorithms.

The original data set had categorical variables on which the analysis and prediction was to be made. Without any feature extraction and conversion, CatBoost which works directly on categorical dataset was used to make base model predictions. The dataset was split into training / test and the train dataset was undersampled to handle the bias.
CatBoost was able to achieve Accuracy of 95% with a ROC value of 91%

B. Algorithm 2 : Logistic Regression

Logistic regression is a statistical model that in its basic form uses a logistic function to model a binary dependent variable. The logistic function is a sigmoid function, which takes any real input and outputs a value between zero and one [23]. The standard logistic function $\sigma : \mathbb{R} \rightarrow (0,1)$ is defined as follows :

$$\sigma(t) = \frac{e^t}{e^t + 1} = \frac{1}{1 + e^{-t}}$$

When it comes to classification, we are determining the probability of an observation to be part of a certain class or not. Therefore, we express the probability with a value between 0 and 1. Logistic Regression computationally takes less time and less resources to get to the final output. As in our classification problem we classify the publisher as either 'Ok' or 'Fraud', thus logistic regression was used as a base model. The accuracy obtained using LR was 66% with a ROC score of 65%.

C. Algorithm 3: ADA Boost with Decision Tree Classifier

A decision tree is a flowchart-like tree structure where an internal node represents feature (or attribute), the branch represents a decision rule, and each leaf node represents the outcome. It partitions the tree in recursive manner called recursive partitioning. Decision Tree selects the best attribute using Attribute Selection Measures (ASM) to split the records, makes that attribute a decision node and breaks the dataset into smaller subsets. Most popular attribute selection measures are Information Gain, Gain Ratio, and Gini Index. By default Decision Tree Classifier present in sklearn package uses Gini Index for ASM [25].
AdaBoost, short for "Adaptive Boosting" focuses on classification problems and aims to convert a set of weak classifiers into a strong one [26]. Adaboost selects a training subset randomly. It iteratively trains the AdaBoost machine learning model by selecting the training set based on the accurate prediction of the last training. It assigns the higher weight to wrong classified observations so that in the next iteration these observations will get the high probability for classification. Also, it assigns the weight to the trained classifier in each iteration according to the accuracy of the classifier. The more accurate classifier will get high weight. This process iterates until the complete training data fits without any error or until reached to the specified maximum number of estimators. To classify, perform a "vote" across all of the learning algorithms you built.
Using ADA boost with Decision Tree we achieved an accuracy of 89% with a ROC score of 90%.

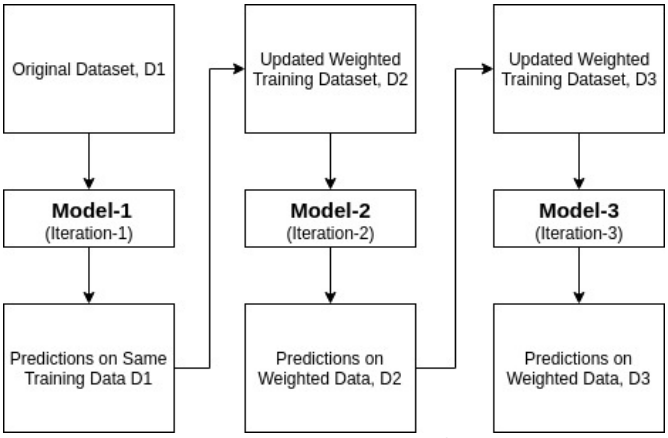

Figure 4 : ADA Boost Classifier

D. Algorithm 4 : Random Forest Classifier

Random Forests (RF) is a robust ensemble algorithm that can be used for both classification and regression. Performance of random forests is on par with other machine learning algorithms but it is much easier to use and more forgiving with regards to over fitting and outliers than other algorithms, so it is always a good option to consider. Each individual tree in the random forest spits out a class prediction and the class with the most votes becomes our model's prediction [28]. It is non-parametric, capable of using continuous and categorical data sets, easy to parameterize, not sensitive in case of over-fitting, good at dealing with outliers, and it calculates ancillary information such as classification error and variable importance.

A large number of relatively uncorrelated models (trees) operating as a committee will outperform any of the individual constituent models.

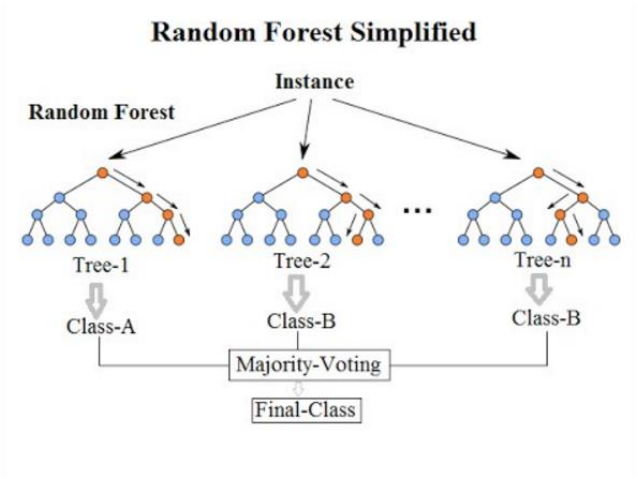

Figure 5 : Random Forest Working

The key to the success of random forests is how it creates each of the decision trees that make up the forest. Decisions trees are very sensitive to the data they are trained on. Small changes to the training set can result in significantly different tree structures. Random forest takes advantage of this by allowing each individual tree to randomly sample from the dataset with replacement, resulting in different trees. This process is known as bagging. Random forest, we end up with trees that are not only trained on different sets of data (bagging) but also use different features to make decisions. Random Forest provided the best accuracy and ROC value on our dataset very much similar to CatBoost Algorithm with Accuracy being 94% and ROC value being 93%.

VI. Evaluation

For the evaluation purpose multiple parameters like precision, recall, accuracy and auc-roc value on the test data were considered. Considering our scenario, identifying Fraud publisher holds more importance than mislabeling the publishers deemed 'Ok'. We consider Recall to be a better evaluating factor with respect to others.

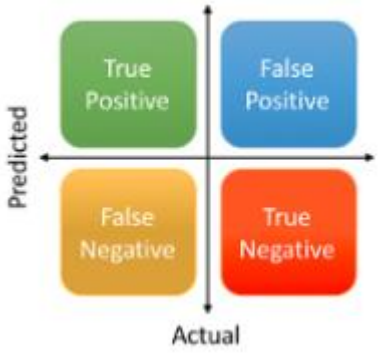

Figure 6 : Type 1 & Type 2 Error

Let us understand the basic types of predictions that can occur from the above given figure. True Positives are those classes which are actually positives and correctly predicted as positives. In our case the publishers which were labelled 'Fraud' and the models predicted as 'Fraud' similarly True Negatives are classes which are actually negative and correctly predicted as 'Ok'.

Coming to False Positive or Type I Error is the incorrect rejection of a true null hypothesis. As per our example, Type 1 Error / False Positive are those records which are labelled 'Ok' but our model predicted them as 'Fraud'.

A False Negative or Type II Error is the failure to reject a false null hypothesis. For our built model Type II errors are those where the classes are predicted as 'Ok' whereas it initially were 'Fraud'. We need to minimize the Type II error for better model predictions.

$$\text{Precision} = \frac{\text{True Positive}}{\text{Actual Results}} \text{ or } \frac{\text{True Positive}}{\text{True Positive + False Positive}}$$

$$\text{Recall} = \frac{\text{True Positive}}{\text{Predicted Results}} \text{ or } \frac{\text{True Positive}}{\text{True Positive + False Negative}}$$

$$\text{Accuracy} = \frac{\text{True Positive + True Negative}}{\text{Total}}$$

Figure 7 : Precision, Recall & Accuracy

Above three formulae show how precision, recall and model accuracy are calculated.

AUC-ROC curve is a performance measurement for classification problem at various thresholds settings. ROC is a probability curve and AUC represents degree or measure of separability. It tells how much model is capable of distinguishing between classes. Higher the AUC, better the model is at predicting Fraud publisher as 'Frauds' and labels 'OK' as 'OK'.

The ROC curve is plotted with TPR against the FPR where TPR is on y-axis and FPR is on the x-axis.

$$\text{TPR /Recall / Sensitivity} = \frac{TP}{TP + FN}$$

$$\text{FPR} = \frac{FP}{TN + FP}$$

Figure 8 : True Positive Rate and False Positive Rate

Algorithm	Recall	Precision	Accuracy	AUC-ROC
Random Forest	94	54	94.44	94.14
CatBoost	88	57	95.04	91.84
ADA Boost + Decision Tree	92	36	89.22	90.64
Logistic Regression	65	12	66.67	65.95

Table 4 : Comparative Evaluation

Amongst the mentioned algorithms, Logistic Regression performed poorly but was the fastest to model. CatBoost was the slowest in computation as it performs 1000 iterations by default, followed by ADA Boost + Decision

Tree. Best results were achieved using Random Forest with maximum recall and highest auc-roc value.

VII. CONCLUSION

There are various algorithms that can be used for fraud detection and classification; however, the efficiency of every algorithm differs while looking at different factors. After reading different published journals and papers, we have a clearer picture of the working of different machine learning algorithms. This has helped me pick beneficial machine-learning algorithm, which have better accuracy in identifying click fraud.

Logistic Regression works well when the data points are independent of each other. This is a major disadvantage, because our research relies on research techniques involving multiple observations of the same individuals in a short span of time. Thus logistics regression performed badly.

Ensembles of decision trees were the most widely used approach which we saw clearly from Literature reviews as it provided fairly fast learning of data, well suited for noisy nonlinear patterns, and mixed variable types. Decision Tree performed good for us and even better when boosted using ADA boost to achieve recall of 92% and accuracy of 89%.

CatBoost is a recently launched algorithm which primarily worked on gradient boosting over decision trees and directly on categorical variables where it is by default tuned for optimum results thus it proved better than ADA boost + Decision Tree.

Random Forest gave us the best recall and accuracy as in case of such classification problems, bagging techniques works better with decision trees.

As future work, we plan to identify more features based on the combined attributes which best describe the behavior of a partner. More experiments on feature selection techniques over created features are needed to identify the best features and hence avoid over-fitting and tuning the top performing algorithms to acquire better accuracy and recall as in our case for better estimation.

REFERENCES

[1] K C Wilbur, Y Zhu, D S. Anderson, "Click Fraud[J]", Access & Download Statistics, vol. 28, no. 2, pp. 293-308, 2009.

[2] Urbanski Al., (01 May 2013). “Bots Mobilize”, DMN [Online]. Available: http://www.dmnews.com/bots-mobilize/article/291566/.

[3] Liu, B & Nath, S & Govindan, R & Liu, J. (2014). DECAF: Detecting and characterizing ad fraud in mobile apps. Proc. of the 11Th USENIX Symposium on Networked Systems Design and Implementation (NSDI’14). 57-70.

[4] X. Jiarui and L. Chen, "Detecting Crowdsourcing Click Fraud in Search Advertising Based on Clustering Analysis," 2015 IEEE 12th Intl Conf on Ubiquitous Intelligence and Computing and 2015 IEEE 12th Intl Conf on Autonomic and Trusted Computing and 2015 IEEE 15th Intl Conf on Scalable Computing and Communications and Its Associated Workshops (UIC-ATC-ScalCom), Beijing, 2015, pp. 894-900.

[5] Munson, Lee (October 15, 2007). "What is a Click Farm?". Security-FAQs. Retrieved January 10, 2017.

[6] Beal V. “Click Fraud”, Webopedia [Online]. Available: https://www.webopedia.com/TERM/C/click_fraud.html

[7] H. Haddadi, "Fighting Online Click-Fraud Using Bluff Ads", ACM SIGCOMM, vol. 40, n. 2, pp. 22-25, April 2010.

[8] S.H. Li, Y.C. Kao, Z.C. Zhang, Y.P. Chuang, and D.C. Yen, “A network behavior-based botnet detection mechanism using PSO and K-means,” ACM TMIS, vol. 6, p. 3, April 2015.

[9] D. Zhao et al., “Botnet detection based on traffic behavior analysis and flow intervals,” Comput. Secur., vol. 39, pp. 2-16, November 2013.

[10] C. Walgampaya, M. Kantardzic, R. Yampolskiy, "Real Time Click Fraud Prevention using multi-level Data Fusion", WCECS 2010, October 20-22, 2010, San Francisco, USA

[11] D. Antoniou et al., "Exposing click-fraud using a burst detection algorithm," 2011 IEEE Symposium on Computers and Communications (ISCC), Kerkyra, 2011, pp. 1111-1116.

[12] https://onion.derby.ac.uk/

[13] Lavin, A. & Ahmad, S., 2015. Evaluating Real-time Anomaly Detection Algorithms—the Numenta Anomaly Benchmark. Miami, IEEE.

[14] Mobile ad fraud rates double, with shopping apps hardest hit [Online]. Available: https://www.retaildive.com/news/mobile-ad-fraud-rates-double-with-shopping-apps-hardest-hit/523240/

[15] Advertising Fraud Losses To Reach $42 Billion In 2019, Driven By Evolving Tactics By Fraudsters [Online]. Available : https://www.juniperresearch.com/press/press-releases/advertising-fraud-losses-to-reach-42-billion

[16] R. J. Oentaryo, E.-P. Lim, M. Finegold, D. Lo, F.-D. Zhu, C. Phua, E.-Y. Cheu, G.-E. Yap, K. Sim, M. N. Nguyen, K. Perera, B. Neupane, M. Faisal, Z.-Y. Aung, W. L. Woon, W. Chen, D. Patel, and D. Berrar, "Detecting click fraud in online advertising: A data mining approach," Journal of Machine Learning Research, vol. 15, pp. 99-140, 2014.

[17] Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A.V. and Gulin, A., 2018. CatBoost: unbiased boosting with categorical features. In Advances in Neural Information Processing Systems (pp. 6638-6648).

[18] Wirth, R. and Hipp, J., 2000, April. CRISP-DM: Towards a standard process model for data mining. In Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining (pp. 29-39). Citeseer.

[19] Encoding Categorical Features [Online]. Available : https://towardsdatascience.com/encoding-categorical-features-21a2651a065c

[20] The Curse of Dimensionality [Online]. Available : https://towardsdatascience.com/the-curse-of-dimensionality-50dc6e49aa1e

[21] Taneja, M., Garg, K., Purwar, A. and Sharma, S., 2015, August. Prediction of click frauds in mobile advertising. In 2015 Eighth International Conference on Contemporary Computing (IC3) (pp. 162-166). IEEE.

[22] Class Imbalance in Credit Card Fraud Detection - Part 2 : Undersampling in Python [Online]. Available : http://blog.madhukaraphatak.com/class-imbalance-part-2/

[23] Logistic regression – Wikipedia [Online]. Available : https://en.wikipedia.org/wiki/Logistic_regression#Logistic_regression_vs._other_approaches

[24] Why, How and When to Scale your Features [Online]. Available : https://medium.com/greyatom/why-how-and-when-to-scale-your-features-4b30ab09db5e

[25] Decision Tree Classification in Python [Online]. Available : https://www.datacamp.com/community/tutorials/decision-tree-classification-python

[26] AdaBoost Classifier in Python [Online]. Available : https://www.datacamp.com/community/tutorials/adaboost-classifier-python

[27] Understanding Random Forest [Online]. Available : https://towardsdatascience.com/understanding-random-forest-58381e0602d2

[28] Perera, K.S., Neupane, B., Faisal, M.A., Aung, Z. and Woon, W.L., 2013. A novel ensemble learning-based approach for click fraud detection in mobile advertising. In Mining Intelligence and Knowledge Exploration (pp. 370-382). Springer, Cham.