IJIRSET
International Journal of Innovative Research in
SCIENCE | ENGINEERING | TECHNOLOGY

e-ISSN: 2319-8753 | p-ISSN: 2347-6710

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

Volume 13, Issue 8, August 2024

ISSN INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.524

9940 572 462 | 6381 907 438 | ijirset@gmail.com | www.ijirset.com

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.524

|www.ijirset.com |A Monthly, Peer Reviewed & Referred Journal| e-ISSN: 2319-8753; p-ISSN: 2347-6710|

Volume 13, Issue 8, August 2024

|DOI: 10.15680/IJIRSET.2024.1308097|

Click Fraud Detection and Visualization

Anish V, Amit Kumar Yadav, Harshit Aryan, S Benny Solomon

U.G. Student, Department of Computer Science and Engineering, Rajarajeswari College of Engineering, Bangalore, Karnataka, India

ABSTRACT: This study focuses on detecting click fraud in mobile app advertisements using data analysis and interactive visualization techniques. Leveraging a dataset from Talking Data, we preprocess the data by removing anomalies and converting timestamps. We propose a model training approach that includes negative down sampling. The visualization of fraud rates by attributes such as IP address and OS is facilitated through D3 plotting. Our project aims to predict user app downloads post- click and provide a user-friendly API for fraud rate prediction. By combining data preprocessing, model training, and visualization, we aim to assist companies in reducing fraudulent charges associated with mobile app ads.

KEY WORDS: Click fraud detection, Mobile app advertise- ments, Data preprocessing, Model training, Fraudulent charges, Interactive visualization

I. INTRODUCTION

Motivated by the prevalence of fraudulent clicks in mobile app advertisements, our project delves into the detection of such deceptive practices. With a focus on mitigating the financial impact on companies, we leverage data analysis and interactive visualization techniques to predict user behavior post-click. Utilizing a dataset from Talking Data, we preprocess the data to ensure accuracy and reliability. By implementing a model training strategy that includes negative down sampling, we aim to enhance the efficiency and effectiveness of fraud detection. Through the visualization of fraud rates by key attributes, such as IP address and OS, we provide a user- friendly interface for clients to identify potential fraudulent activities and reduce associated charges.

II.SURVEY

Click fraud in online advertising is now recognized as a significant cyber scam, with an estimated fraudulent click rate ranging from 10 percent to 20 percent. The financial gains for website owners from ad-clicks have attracted fraudsters to engage in deceptive practices. An analysis of Zero Access activities revealed a staggering volume of approximately one million fraudulent clicks daily, diverting around USD 100,000 in ecosystem revenue. These findings underscore the critical need for effective fraud detection methods to combat such fraudulent activities in the digital advertising space

A. Historical Algorithms

In the realm of click fraud detection, historical algorithms have played a crucial role in identifying deceptive practices within network-oriented services based on user accessibility. One such early algorithm, Bluff Ads, heightened the chal- lenge for click fraudsters by scrutinizing individual clicking behavior to uncover fraudulent clicks, albeit facing scalability constraints. Recent trends have seen a surge in the popularity of learning-based prediction models. For instance, a study introduced online advertising algorithms capable of detecting duplicate clicks using decaying window models like sliding and jumping windows, leveraging timing bloom filters. An- other research effort proposed an ensemble learning approach, emphasizing innovative feature selection and embedding tech- niques to identify common patterns such as clicking time intervals and IP addresses. While these advancements have shown promise, challenges persist in ensuring the representativeness of merged attributes and optimizing models for online learning settings. Notably, initiatives by Google AI and FAIR have explored practical fraud detection challenges, introducing strategies like memory optimization, model selection, and the integration of gradient boosting decision tree (GBDT) and logistic regression. Despite the success in feature identification, it was noted that GBDT may be less suitable for online learning scenarios. Additionally, the development of open- source frameworks for GBDT training has provided scalability benefits for efficient training on sparse data, albeit with some trade-offs in accuracy.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.524

|www.ijirset.com |A Monthly, Peer Reviewed & Referred Journal| e-ISSN: 2319-8753; p-ISSN: 2347-6710|

Volume 13, Issue 8, August 2024

|DOI: 10.15680/IJIRSET.2024.1308097|

Network Systems and Mobile Applications

In 2010, a publication advocated for the fusion of academic algorithms with practical scenarios and authentic datasets to enhance fraud detection [11]. Subsequently, numerous stud- ies have integrated these algorithms into network systems tailored for industrial applications. A real-time system was proposed to combat click fraud by leveraging collaboration between server-side and client-side components [12]. Another initiative focused on addressing click fraud in mobile environments, enabling the system to validate user requests and flag potentially fraudulent activities. Additionally, a real-time ad-click fraud detection system analyzed click record data, assigning scores based on click quality, albeit necessitating a substantial dataset for evaluation. Furthermore, innovative applications have emerged, such as a 2014 tool enabling users to detect click fraud activities originating from click bots on the advertiser side [15]. This tool involved proactive functionality testing and passive monitoring of user behaviors to enhance fraud detection capabilities.

III. PROPOSED METHOD

Data Preprocessing: The dataset originates from Talking Data, China's leading independent big data service provider, encompassing both training and test sets. The raw training set comprises 1,000,000 click instances, while the raw test set contains over 1,000 click instances. Given the substantial size of the raw training set and the absence of ground truths in the raw test set, we opted to partition the raw training set for training and evaluation purposes in our project. This dataset is structured as a .csv file, with each row representing a click instance associated with seven attributes: user IP triggering the click ("IP"), the mobile app involved ("app"), the mobile device ("device"), the operating system of the device ("OS"), the channel number ("channel"), the timestamp when the click occurred ("click time"), and the timestamp of app download, if applicable ("attributed time"). The target variable we aim to predict is attributed," a binary label indicating whether the user downloaded the app after clicking the ad (1) or not (0). To ensure data quality, we initially eliminate extreme abnormal values that may result from erroneous data inputs. Subsequently, we convert the "click time" data type from string to date-time format and further transform it into numerical values by calculating the time difference between the current "click time" and the minimum "click time" in seconds. As most entries in the "attributed time" column are null, we conduct feature selection by excluding this column. Conversely, the remaining columns contain valuable information and significantly impact the final classification outcomes. Therefore, we retain the other six feature attributes. Additionally, we standardize the data format by obtaining the official names from each company or app's website and utilizing regex library to identify and replacevarious expressions with standardized terms.

Model Training

Negative Down Sampling is a technique employed to ad- dress the issue of training set imbalance, particularly when the number of negative samples significantly outweighs the number of positive training samples. The primary objective is to maintain or enhance the positive samples while reducing the number of negative samples in the training set, ensuring a balanced distribution of positive and negative instances. The rationale behind adopting negative down sampling during model training stems from observations during data analysis. Notably, over 95 percent of click instances have ground truth labels of 0, indicating that the majority of ad-clicks do not lead to app downloads. Training the model on this unbalanced dataset could result in learning from skewed positive and negative information. Such training may introduce a strong bias towards the prior label distribution, leading to poor performance in predicting minority samples, particularly the positive instances. By implementing negative down sampling, more positive sample information is exposed to the model within a fixed memory constraint.

Gradient Boosting Decision Trees (GBDT) is selected as one of the classifier candidates for the supervised binary classification task due to several reasons. Firstly, GBDT conducts splits up to the maximum depth through an algorithm and subsequently prunes the tree backward, eliminating splits that do not yield positive gains. This approach allows GBDT to explore deeper spaces compared to other gradient boosting machines that halt node splitting upon encountering negative loss. Given the presence of multiple attributes with high variance, deeper trees in GBDT can potentially provide more information. Additionally, GBDT offers regularization of leaf weight values using popular functions like L2 and L1, effec- tively mitigating overfitting caused by excessive exploration. Moreover, GBDT is adept at handling discrete attribute values by utilizing Classification and Regression Trees (CART).

Multilayer Perceptron (MLP) is another classifier candidate comprising a three-layer neural network with an input layer, a hidden layer, and an output layer. Each layer is fully connected to the subsequent layer, enabling MLP to

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.524

|www.ijirset.com |A Monthly, Peer Reviewed & Referred Journal| e-ISSN: 2319-8753; p-ISSN: 2347-6710|

Volume 13, Issue 8, August 2024

|DOI: 10.15680/IJIRSET.2024.1308097|

capture non-linear relationships through activation functions between layers. MLP is chosen for its capability in handling non- linearity in data, especially in scenarios where the training data size may not be sufficient for deep neural networks. Relu is utilized as the non-linear activation function, with cross- entropy serving as the loss function.In addition to GBDT and MLP, the performance of Support Vector Machine (SVM) and Random Forest algorithms is also explored in the context of the study [T3]. fraud rate grouped by IP address, OS, click time period of a day, etc. These requests can be sent via our RESTful API to the backend, where the group data will be handled and processed, then passed back in the response body to the frontend.We plan to utilize D3 as the tool for plotting. As mentioned in the proposal method, we will visualize the batch query result using D3. Once the backend passes the fraud rates, we will plot the statistics of the relation between fraud rates and each attribute.Clients will get an overview of each figure and can interact with the detailed plots.It's important to note that the click time here represents the relative time regarding the earliest click time in the training set, which is November 7, 2017, 00:00:00. Since the model is trained on the entire batch instead of certain instances, users will need to self-calculate the relative time value (number only) and input it into the corresponding field

IV. EXPERIMENTS AND EVALUATION

Question 1: Is there any relationship between the conversion rate and click features?
Observations: We investigated the conversion rate across different IP addresses and discovered that the conversion rate is not correlated with the frequency of IP addresses. However, we identified 22 IP addresses that triggered clicks more than 100 times, which we labeled as suspect fraud clicks.Further exploration into the distribution of features such as App, device, OS, and channel revealed that fraud clicks predominantly occur within specific ranges of these attributes. For instance, we found that fraud clicks are concentrated within certain small ranges of App, device, OS, and channel.As an example, we display the cumulative distribution function (CDF) of OS to illustrate this point.

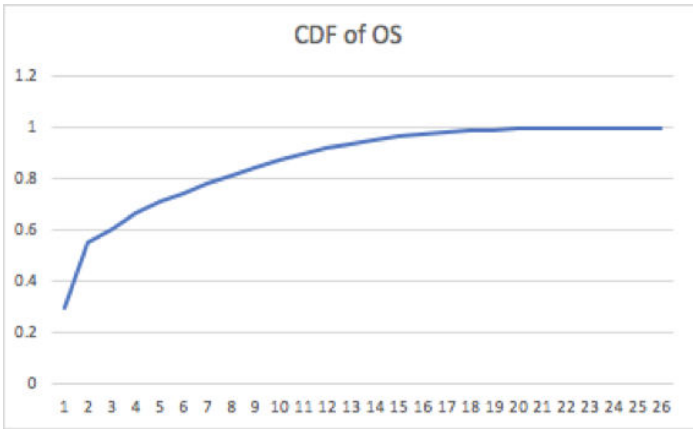


Fig.1

From the figure above, we can infer that the top 10 operating systems collectively account for approximately 68.8 percent of the clicks generated by the top 22 IP addresses. This suggests that the majority of suspect fraud clicks occur on a few specific operating systems. Similarly, our analysis indicates that the top 10 apps con- tribute to about 81.6 percent of the clicks produced by those 22 IP addresses. This implies that most suspect fraud clicks are concentrated within a narrow range of apps. Furthermore, we observed that the top 2 device types comprise approximately 97.68 percent of the total suspect fraud clicks, with type 1 device being particularly prominent. This suggests that the device type may play a crucial role in determining whether a fraud click is generated.However, we found that the distribution of suspect fraud clicks among channels is more evenly spread compared to the other three attributes. This suggests that channels may not serve as a significant discriminator for determining fraud clicks.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.524

|www.ijirset.com |A Monthly, Peer Reviewed & Referred Journal| e-ISSN: 2319-8753; p-ISSN: 2347-6710|

Volume 13, Issue 8, August 2024

|DOI: 10.15680/IJIRSET.2024.1308097|

Question 2: Are there any temporal patterns in these clicks? **observation:** To investigate potential time patterns, we analyzed the hourly click and conversion frequencies. While no distinct pattern emerged in the hourly conversion ratio, a cyclic trend was observed in the hourly click volume. The frequency of clicks remained relatively consistent throughout each day, gradually declining to its lowest point by the end of the day. The corresponding figures illustrating these patterns are presented below:

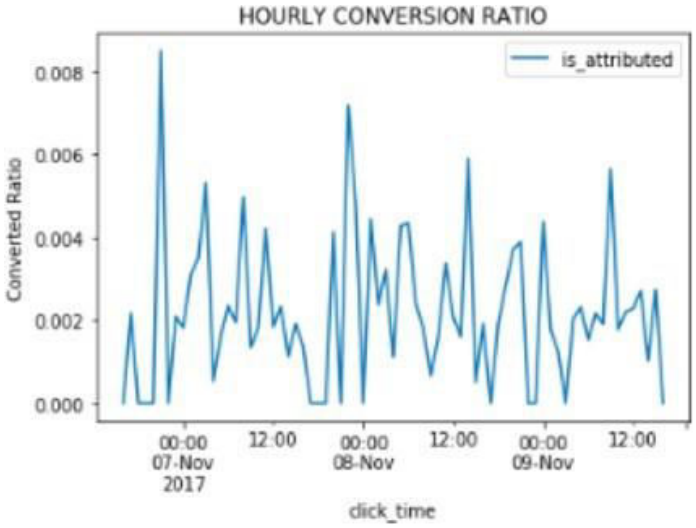
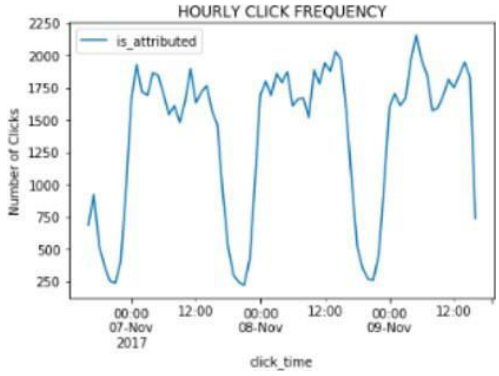


Fig.2

Question 3 : How can a balanced dataset be generated using negative down sampling?
Table 1 presents the statistics for the training and testing sets created in our study.
We developed two training sets: one utilizing negative down sampling (Set A) and the other without this technique (Set B). To ensure both training sets are of equal size, we followed the steps outlined below. For Set A, negative down sampling was applied to all available training data. Positive samples were retained, and negative samples were randomly selected without replacement from the remaining data until the number of positive and selected negative samples matched. Subsequently, the negative and positive samples in Set A were shuffled randomly. To create Set B, samples of the same size as Set A were randomly selected without replacement from all available training data. Given the substantial volume of training data, these steps were executed in batches sequentially. Ultimately, two training sets of approximately 40,000 instances each were obtained, as indicated in Table 1. Additionally, we formed five test sets of varying sizes with a train-test split ratio of approximately 4:1. It is important to note that while the original dataset includes ground truth labels for evaluation purposes, we opted not to apply negative sampling to the test sets. This decision was made because the test samples provided by users in practical scenarios typically adhere to the original data distribution, which inherently reflects an unbalanced distribution.



International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.524

|www.ijirset.com |A Monthly, Peer Reviewed & Referred Journal| e-ISSN: 2319-8753; p-ISSN: 2347-6710|

Volume 13, Issue 8, August 2024

|DOI: 10.15680/IJIRSET.2024.1308097|

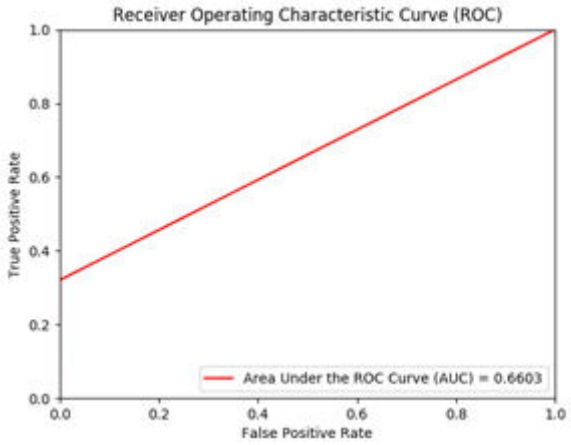
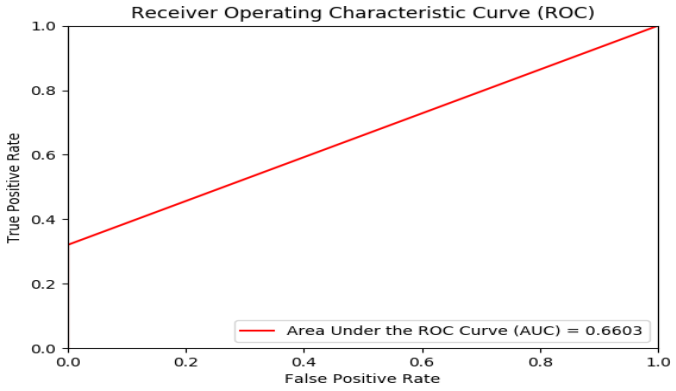


Fig. 3

		Size
Training	Train/Val Sets (10-fold cross validation)	38452
Testing	Test Set 1	9372
	Test Set 2	11149
	Test Set 3	12782
	Test Set 4	14012
	Test Set 5	14306

Table 1: Dataset Information

Question 4 : Which metrics should be selected to assess model performance on highly imbalanced datasets? In our evaluation, we opted for the Area Under the Receiver Operating Characteristic curve (AUC) as the primary metric. The AUC score quantifies the area under the Receiver Operating Characteristic (ROC) curve, which graphically represents the True Positive Rate (TPR) against the False Negative Rate FNR) at various classification thresholds. A high AUC score, closer to 1, indicates a model's effectiveness in distinguishing between positive and negative classes. TPR and FNR are crucial components in assessing the model's performance on imbalanced datasets FNR) at various classification thresholds. A high AUC score, closer to 1, indicates a model's effectiveness in distinguishing between positive and negative classes. TPR and FNR are crucial components in assessing the model's performance on imbalanced datasets.



International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.524

|www.ijirset.com |A Monthly, Peer Reviewed & Referred Journal| e-ISSN: 2319-8753; p-ISSN: 2347-6710|

Volume 13, Issue 8, August 2024

|DOI: 10.15680/IJIRSET.2024.1308097|

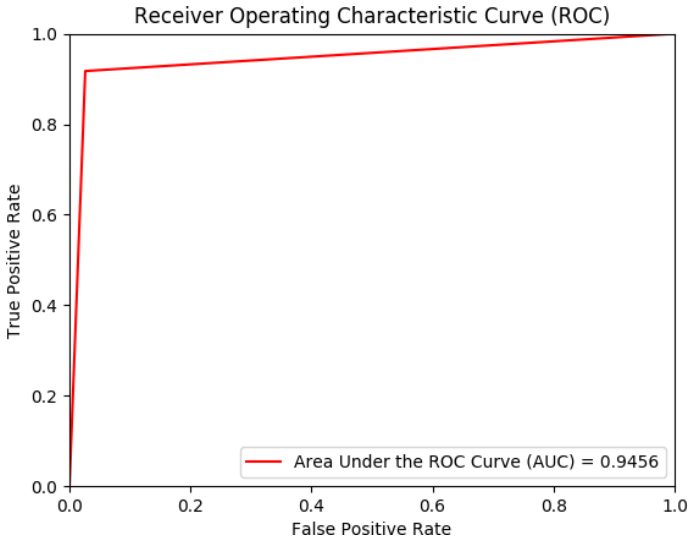


Fig. 4. GDBT training ROC using unbal- anced dataset

The calculations are as follows: True Positive Rate (TPR): TPR = True Positive (TP) / (True Positive (TP) + False Negative (FN)) False Negative Rate (FNR): FNR = False Positive (FP) / (False Positive (FP) + True Negative (TN)) It is important to note that our model evaluation does not rely on classification accuracy metrics. In the context of highly imbalanced datasets, achieving high classification accuracy under a fixed threshold does not guarantee adequate class differentiation by the classifier. For instance, a classifier could achieve over 95 percent accuracy by predicting all test samples as the dominant class, leading to incorrect predictions for mi- nority samples. Therefore, we refrain from using classification accuracy in our problem setting.

Additionally, we present the precision and recall scores in the experiments conducted.
Question 5 : How effective is negative sampling in model training?
Table 2 presents a comparison of the training and testing performance between two models: one trained on a dataset generated using negative down sampling (Model A) and the other trained without negative down sampling (Model B). The results from Table 2 indicate that Model A significantly outperforms Model B across various classifiers. Models trained on Model B exhibit training and testing scores around 0.50 to 0.60, suggesting that Model B struggles to learn meaningful information and tends to make random guesses. Visualization examples in Figure 4 and Figure 5 illustrate the training ROC generated by the Gradient Boosting Decision Trees (GBDT) model on both balanced and unbalanced datasets.

Furthermore, Table 3 showcases precision-recall statistics for testing performance on test sets 1 and 3. Notably, when testing with Model B, the recall for the positive class is approximately 1×10(power)-2, indicating that Model B tends to assign dominant class labels to all test samples due to the strong class conditional distribution prior assumption. Conversely, Model A achieves a recall of around 0.9 for the positive class, demonstrating the effectiveness of the negative down sampling approach.

Question 6 : What classifier should be selected?
A comparison of training and testing performance for four different classifiers is presented in Table 2. The parame- ters of Gradient Boosting Decision Trees (GBDT), Random Forest, and Support Vector Machine (SVM) were fine-tuned using 10-fold cross-validation. To expedite training speed, we normalized the training and testing data for SVM and Multilayer Perceptron (MLP). For MLP, a constant learning rate of 0.01 was employed along with the Adam optimizer with momentum, L2 regularization, and a maximum iteration of 200. The specific parameters for GBDT can be found in Table 5.

Question 7 : How can the clicks be effectively visualized? **Observation:** The analysis reveals a significant bias in the clicks, with a majority being fraudulent. Visualizing the entire dataset directly may lead to confusion due to this

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.524

|www.ijirset.com |A Monthly, Peer Reviewed & Referred Journal| e-ISSN: 2319-8753; p-ISSN: 2347-6710|

Volume 13, Issue 8, August 2024

|DOI: 10.15680/IJIRSET.2024.1308097|

imbalance. To address this, we opt to focus on highlighting the most prominent characteristics of fraudulent clicks by visualizing the top occurrences of each feature. This targeted approach aims to provide clients with insights into the key attributes of fraudulent clicks.to quickly get some idea about what kind of clicks is likely fraudulent.

V. EVALUATION

In this project, we offer users an API that prompts them to input the necessary features for predicting the fraud rate.

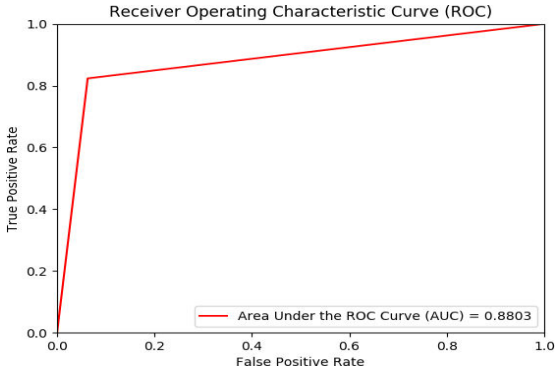


Fig. 5. GBDT ROC: test set 1

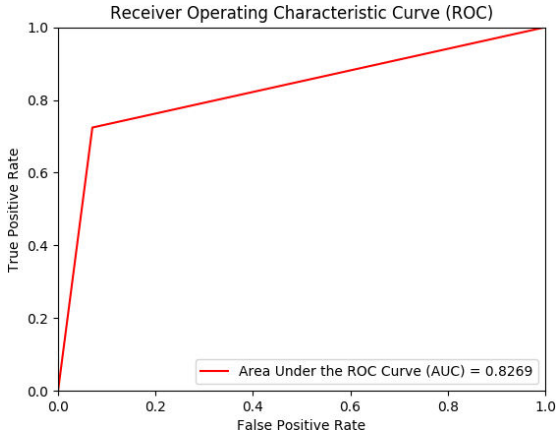


Fig. 6. GBDT ROC: test set 2

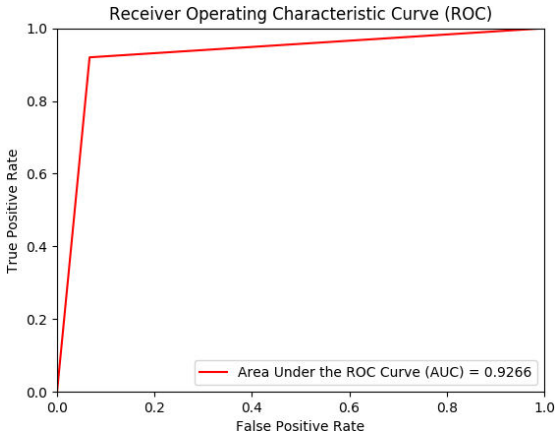


Fig. 7. GBDT ROC: test set 3

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.524

|www.ijirset.com |A Monthly, Peer Reviewed & Referred Journal| e-ISSN: 2319-8753; p-ISSN: 2347-6710|

Volume 13, Issue 8, August 2024

|DOI: 10.15680/IJIRSET.2024.1308097|

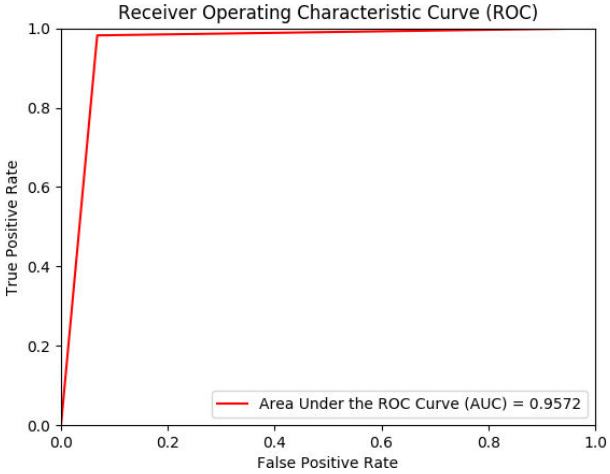


Fig. 8. GBDT ROC: test set 4

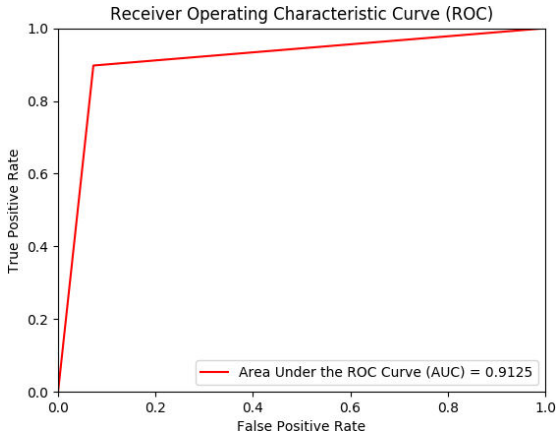


Fig. 9. GBDT ROC: test set 5

The frontend design utilizes D3 for creating visualizations and raw JavaScript for data processing. On the backend side, we implement a local service using the Python framework Flask. Within the backend system, the service invokes our trained model to forecast the fraud rate. Users can also access statistics detailing the most frequent values for each feature in fraudulent clicks. The typical workflow is outlined below:

1. The user provides the values of the queried click feature

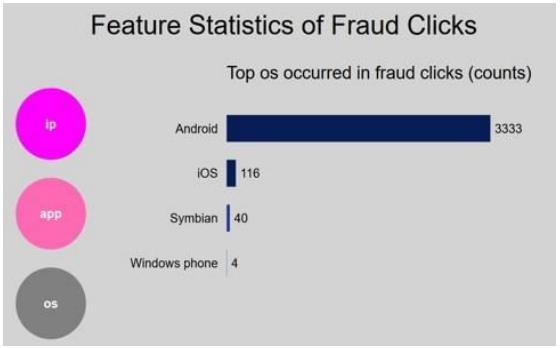


Fig. 10. Bar Chart of Statitics

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.524

|www.ijirset.com |A Monthly, Peer Reviewed & Referred Journal| e-ISSN: 2319-8753; p-ISSN: 2347-6710|

Volume 13, Issue 8, August 2024

|DOI: 10.15680/IJIRSET.2024.1308097|

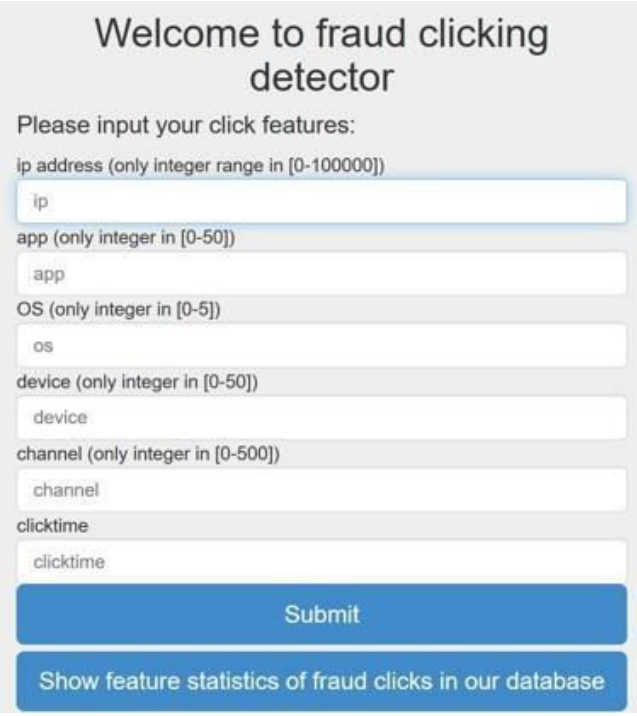


Fig. 11. User Interface

(such as OS, device) for a single click instance. 2. Upon inputting the values, the user clicks the ”submit” button to transmit the input data to the backend. 3. The backend receives the feature values of the click instance and utilizes our trained model to predict a binary label indicating whether the instance is a fraudulent click. 4. The backend sends the prediction result back to the frontend, updating the user interface to display the prediction outcome.

For users visiting the statistics page, an interactive visualiza- tion feature is available. Users can hover over the features of interest, triggering a bar chart that showcases the top occurring values of that feature among fraudulent clicks. It is important to note that the prediction statistics presented in our project are derived from the prediction results of test set 3. Users who wish to submit their own batch of data can package their data in a pickle file and directly submit it to the backend server.

VI. CONCLUSION

Through our data analysis, we identified that the features of App, device, OS, and channel exert a significant influence on the final prediction outcome. Additionally, we uncovered a cyclic pattern in the hourly volume of clicks.Our model, constructed on Gradient Boosting Decision Trees (GBDT), demonstrates robust performance on highly imbalanced datasets, achieving an average testing Area Under the Curve (AUC) score exceeding 0.90.To enhance clarity and highlight key insights for clients, we have developed a user-friendly interface. This interface aids clients in identifying fraudulent clicks and provides a summary of statistics that enable clients to swiftly identify prominent characteristics associated with fraudulent activities.

REFERENCES

1. Nir Kshetri. The economics of click fraud.IEEE Security and Privacy, 8(3):45–53, 2010.
2. Hamed Haddadi. Fighting online click-fraud using bluff ads. ACM SIGCOMM Computer Communication Review, 40(2):21–25, 2010.
3. Juha Salo and Ahti Muhonen. Method and apparatus for detecting click fraud, August 17 2010. US Patent 7,779,121.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.524

|www.ijirset.com |A Monthly, Peer Reviewed & Referred Journal| e-ISSN: 2319-8753; p-ISSN: 2347-6710|

Volume 13, Issue 8, August 2024

|DOI: 10.15680/IJIRSET.2024.1308097|

4. Paul Pearce, Vacha Dave, Chris Grier, Kir- ill Levchenko, Saikat Guha, Damon McCoy, Vern Paxson, Stefan Savage, and Geoffrey M Voelker. Characterizing large-scale click fraud in zeroaccess. In Proceedings of the 2014 ACM SIGSAC Conference on Computer and Com- munica- tions Security, pages 141–152. ACM, 2014.
5. H Brendan McMahan, Gary Holt, David Scul- ley, Michael Young, Dietmar Ebner, Julian Grady, Lan Nie, Todd Phillips, Eugene Davy- dov, Daniel Golovin, et al. Ad click prediction: a view from the trenches. In Proceedings of the 19th ACM SIGKDD international confer- ence on Knowledge discovery and data mining, pages 1222–1230. ACM, 2013.
6. Kasun S Perera, Bijay Neupane, Mustafa Amir Faisal, Zeyar Aung, and Wei Lee Woon. A novel ensemble learning-based approach for click fraud detection in mobile advertising. In Mining Intelligence and Knowledge Explo- ration, pages 370–382. Springer, 2013.
7. [5] Linfeng Zhang and Yong Guan. Detecting click fraud in pay-per-click streams of online adver- tising networks. In Distributed Computing Sys- tems, 2008. ICDCS'08. The 28th International Conference on, pages 77–84. IEEE, 2008
8. Kenneth C Wilbur and Yi Zhu. Click fraud. Marketing Science, 28(2):293–308, 2009
9. Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining, pages 785–794. ACM, 2016.
10. Richard Oentaryo, Ee-Peng Lim, Michael Fine- gold, David Lo, Field Zhu, Clifton Phua, Eng-Yeow Cheu, Ghim-Eng Yap, Kelvin Sim, Minh Nhut Nguyen, et al. Detecting click fraud in online advertising: a data mining approach. The Journal of Machine Learning Research, 15(1):99–140, 2014.
11. Haitao Xu, Daiping Liu, Aaron Koehl, Hain- ing Wang, and Angelos Stavrou. Click fraud detection on the advertiser side. In European Sym- posium on Research in Computer Security, pages 419–438. Springer, 2014.
12. Mehmed Kantardzic, Chamila Walgampaya, Brent Wenerstrom, Olek- sandr Lozitskiy, Sean Higgins, and Darren King. Improving click fraud detection by real time data fusion. In Signal Processing and Information Technology, 2008. ISSPIT 2008. IEEE International Sym- posium on, pages 69–74. IEEE, 2008.
13. Arvind Gupta, Ashutosh Tiwari, Gopalakrish- nan Venkatraman, Do- minic Cheung, Stacy R Bennett, and Douglas B Koen. Mobile click fraud prevention, August 5 2014. US Patent 8,799,069.
14. Li Ge and Mehmed Kantardzic. Real-time click fraud detecting and blocking system, Novem- ber 1 2007. US Patent App. 11/413,983.

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi cross ref

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

9940 572 462 6381 907 438 ijirset@gmail.com

www.ijirset.com

Scan to save the contact details