International Journal of Engineering Research and Science & Technology

www.ijerst.org

ISSN : 2319-5991

Vol. 22 No. 1 (2026)



ijerst.editor@gmail.com

editor@ijerst.com

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com
Vol. 22, No. 1, 2026
© 2026 IJERST. All Rights Reserved
Peer Reviewed, Referred & Indexed Journal

Research

MACHINE LEARNING AND DEEP NEURAL NETWORK APPROACH FOR ONLINE AD CLICK FRAUD DETECTION

CH SRILAKSHMI[1], BURIKI ROOPITHA[2], AMANCHA HARIKRISHNA[3], BADAVATH BHASKAR[4], BOLLAVATHRI RATHAN[5], ENDIGA RANADEER[6]

ASSISTANT PROFESSOR[1], UG SCHOLAR[2,3,4,5&6]

DEPARTMENT OF CSE, NARSIMHA REDDY ENGINEERING COLLEGE (UGC- AUTONOMOUS) MAISAMMAGUDA (V), KOMPALLY, SECUNDERABAD, TELANGANA-500100

ABSTRACT

Online advertising has become one of the most important revenue sources for digital platforms, but it is increasingly threatened by **ad click fraud**, where malicious users or automated bots generate fake clicks on advertisements to manipulate advertising costs and revenues. These fraudulent activities lead to significant financial losses for advertisers and reduce the reliability of online advertising ecosystems. To address this challenge, this study proposes a **Machine Learning and Deep Neural Network–based approach for detecting online ad click fraud**. The proposed system utilizes historical clickstream data and user behavioral features such as **IP address patterns, click frequency, session duration, device information, and geographic location** to identify suspicious activities. Initially, data preprocessing and feature extraction techniques are applied to clean and transform raw click data into meaningful inputs for predictive models. Machine learning algorithms such as **Random Forest, Decision Tree, and Logistic Regression** are used to classify click activities as legitimate or fraudulent. In addition, a **Deep Neural Network (DNN)** model is employed to capture complex patterns and hidden relationships within large-scale advertising datasets. The hybrid approach combines the strengths of traditional machine learning and deep learning methods to improve detection accuracy and reduce false positives. Experimental results demonstrate that the proposed model effectively identifies fraudulent ad clicks and enhances the reliability of digital advertising systems. The system can assist advertising platforms and marketers in **minimizing financial losses, improving campaign transparency, and ensuring fair advertising practices**. This research contributes to the development of intelligent and scalable fraud detection mechanisms for modern online advertising

INTRODUCTION

Online advertising has become one of the most important revenue sources for digital platforms, businesses, and publishers. Advertising models such as **Pay-Per-Click (PPC)** allow advertisers to pay only when users click on their ads. However, the rapid growth of digital advertising has also led to the rise of **ad click fraud**, where fraudulent clicks are generated intentionally to manipulate advertising costs or increase revenue for publishers. These fraudulent activities are often performed by **bots, automated scripts, or malicious users**, causing significant financial losses to advertisers and reducing the effectiveness of online advertising campaigns. Traditional fraud detection methods rely mainly on **rule-based systems and manual monitoring**, which are often ineffective against modern and sophisticated click fraud attacks. Fraudsters continuously modify their strategies, making it difficult for conventional systems to detect abnormal click patterns. Therefore, advanced intelligent systems are required to analyze large volumes of advertising data and identify suspicious behaviors accurately. **Machine Learning (ML)** has emerged as a powerful approach for detecting fraudulent activities in online advertising. ML algorithms can analyze historical click data and identify patterns that distinguish legitimate user behavior from fraudulent activities. Techniques such as **Random Forest, Support Vector Machines (SVM), Decision Trees, and Logistic Regression** are commonly used to classify ad clicks as genuine or fraudulent based on features such as IP address, click frequency, user behavior, and device information. In recent years, **Deep Neural Networks (DNNs)** have further improved fraud detection capabilities by learning complex patterns and relationships within large datasets. Deep learning models can automatically extract high-level features from clickstream data and detect sophisticated fraud patterns that may not be easily identified using traditional machine learning methods. By combining machine learning and deep neural network approaches, it becomes possible to build a more accurate and scalable system for detecting fraudulent ad clicks. The

proposed system focuses on developing an **intelligent ad click fraud detection framework** that leverages both machine learning and deep neural network algorithms. The system analyzes advertising data, identifies abnormal click behavior, and classifies clicks as legitimate or fraudulent in real time. This approach helps improve advertising transparency, reduce financial losses, and enhance the overall reliability of digital advertising platforms. In summary, the integration of **machine learning and deep neural networks** provides a robust solution for combating online ad click fraud by enabling accurate detection, automated analysis, and scalable fraud prevention in digital advertising ecosystems.

LITERATURE REVIEW

Online advertising has become a major revenue source for websites and digital platforms. However, the increasing use of **pay-per-click (PPC) advertising models** has also led to a rise in **click fraud**, where automated bots or malicious users repeatedly click advertisements to generate illegitimate revenue or exhaust competitors' advertising budgets. Click fraud can significantly impact the reliability and profitability of online advertising systems. Therefore, many researchers have explored **machine learning (ML) and deep learning (DL) techniques** to automatically detect fraudulent ad clicks.

Early Research on Click Fraud Detection

Early click fraud detection methods relied on **rule-based systems and statistical analysis** of user behavior. These systems monitored metrics such as **click-through rate (CTR), IP address repetition, and abnormal traffic patterns**. However, these traditional techniques were limited because sophisticated bots could mimic human browsing behavior. As digital advertising grew rapidly, researchers began exploring **data-driven techniques using machine learning algorithms** to identify fraudulent clicks more accurately.

Click fraud typically occurs when automated programs or malicious users intentionally click advertisements without genuine interest in the product or service, which can damage advertisers' revenue and trust in advertising platforms.

Machine Learning Approaches for Click Fraud Detection

Machine learning techniques have been widely used to detect fraudulent clicks by analyzing large volumes of advertising data. These methods use features such as:

- IP address patterns
- Time intervals between clicks
- Session duration
- User browsing behavior
- Device and geographic information

Several supervised learning algorithms have been applied, including:

- Support Vector Machine (SVM)
- Random Forest
- Decision Tree
- Logistic Regression
- k-Nearest Neighbor (KNN)

These algorithms learn patterns from labeled datasets and classify ad clicks as **legitimate or fraudulent**. Studies show that ensemble methods like **Random Forest and Gradient Boosting** often outperform basic classifiers due to their ability to handle complex and non-linear patterns in clickstream data.

In some research, user browsing behavior such as page views, scrolling actions, and interaction time was used to distinguish between human users and automated bots. These behavioral features significantly improved detection accuracy.

Deep Learning Techniques for Ad Click Fraud Detection

Recent research has explored **deep learning models** to improve fraud detection performance. Deep learning methods can automatically learn hidden patterns from large datasets without requiring extensive manual feature engineering.

Common deep learning architectures used include:

- Artificial Neural Networks (ANN)
- Deep Neural Networks (DNN)
- Convolutional Neural Networks (CNN)
- Recurrent Neural Networks (RNN)

These models analyze sequential click behavior, temporal patterns, and complex relationships between user interactions. Studies show that deep learning models can achieve very high detection accuracy, with some experiments reporting **accuracy levels above 97%** in identifying fraudulent clicks.

Deep learning approaches are particularly effective in identifying **bot-generated traffic**, coordinated click campaigns, and advanced fraudulent strategies that traditional models struggle to detect.

Hybrid Machine Learning and Deep Learning Models

To improve detection performance, many recent studies combine **machine learning and deep learning techniques** in hybrid frameworks. These systems integrate feature extraction from deep neural networks with classification models such as logistic regression or decision trees.

For example, some research combines **Artificial Neural Networks with Logistic Regression, Naïve Bayes, and Decision Trees**, achieving very high performance in detecting fraudulent ad clicks. Such hybrid models can capture both complex patterns and interpretable decision boundaries.

Hybrid architectures also help address challenges such as:

- Imbalanced datasets
- Large-scale advertising data

- Real-time detection requirements

Feature Engineering and Datasets

The effectiveness of ML and DL models largely depends on the quality of **feature engineering and training datasets**. Researchers typically use features derived from:

1. **User behavior features** – click frequency, browsing duration, mouse movements
2. **Network features** – IP addresses, device fingerprints, location
3. **Temporal features** – time intervals between clicks
4. **Advertisement interaction features** – click-through rate and conversion rate

These features help models distinguish **human browsing patterns from automated bot activity**.

Challenges in Ad Click Fraud Detection

Despite significant progress, several challenges remain in click fraud detection systems:

- **Sophisticated bots** that mimic human browsing behavior
- **Large-scale real-time data processing requirements**
- **Imbalanced datasets**, where fraudulent clicks are much fewer than legitimate clicks
- **Adversarial attacks** that attempt to bypass detection models

Researchers continue to explore **advanced deep learning models, explainable AI, and hybrid detection frameworks** to address these challenges.

Research Gap and Future Directions

Although machine learning and deep learning techniques have significantly improved ad click fraud detection, some gaps remain:

- Limited real-time detection systems for large-scale advertising networks
- Lack of explainable AI models to interpret fraud detection decisions
- Need for cross-platform fraud detection systems
- Limited availability of publicly available datasets

Future research may focus on **explainable AI, graph-based learning, and blockchain-based ad verification systems** to further enhance the security and transparency of online advertising.

SYSTEM ANALYSIS

EXISTING SYSTEM

In current online advertising platforms, detecting fraudulent ad clicks is a major challenge. Traditional systems mainly rely on **rule-based methods and basic statistical analysis** to identify suspicious activities. These methods monitor parameters such as repeated clicks from the same IP address, abnormal click patterns, and unusual user behavior.

Many advertising networks use **manual monitoring systems and threshold-based techniques** to detect click fraud. In some cases, basic machine learning algorithms like **Logistic Regression or Decision Trees** are used to classify click activity as legitimate or fraudulent.

However, these systems have several limitations:

- Rule-based systems cannot effectively detect **complex or evolving fraud patterns**.
- Traditional machine learning models rely heavily on **manual feature engineering**.
- Fraudsters often use **botnets, proxy servers, and automated scripts** to mimic human behavior.
- Detection accuracy may decrease when dealing with **large-scale advertising data**.
- Existing systems often lack **real-time fraud detection capability**.

Because of these limitations, many fraudulent clicks remain undetected, resulting in **financial losses for advertisers and reduced trust in digital advertising platforms**.

PROPOSED SYSTEM

The proposed system introduces an **advanced fraud detection framework using Machine Learning and Deep Neural Networks (DNN)** to improve the accuracy and reliability of online ad click fraud detection.

In this system, large volumes of ad click data such as **IP address, timestamp, device type, click frequency, user behavior patterns, and geographic location** are collected and processed. After preprocessing and feature extraction, the data is fed into machine learning models and deep neural networks for classification.

Deep learning models are capable of learning **complex hidden patterns and relationships** in click behavior that traditional models may fail to detect.

The proposed system includes the following key features:

- **Data preprocessing and feature engineering** to improve data quality
- **Machine Learning models** such as Random Forest or Support Vector Machine for initial classification
- **Deep Neural Networks (DNN)** for detecting complex fraud patterns
- **Real-time fraud detection system** for monitoring ad clicks

- **Automated alert generation** when suspicious activities are detected

Advantages of the proposed system include:

- Higher **fraud detection accuracy**
- Ability to detect **sophisticated bot-based click fraud**
- Improved **scalability for large advertising datasets**
- Reduced **financial loss for advertisers**
- Enhanced **security and reliability of digital advertising platforms**

IMPLEMENTATION

MODULES

The proposed system for online ad click fraud detection is organized into several functional modules that work together to identify fraudulent clicks in digital advertising platforms using machine learning and deep neural networks.

Data Collection Module

This module gathers ad click data from various sources such as **advertising networks, web servers, and user activity logs**. The collected dataset may include:

- User IP address
- Device type
- Browser information
- Click timestamp
- Ad ID and publisher ID
- Geographic location

This information forms the base dataset for training and testing fraud detection models.

Data Preprocessing Module

Raw clickstream data often contains missing, duplicate, or irrelevant values. This module performs preprocessing tasks to prepare the dataset for analysis.

Key operations include:

- Data cleaning and removal of duplicate records
- Handling missing values
- Data normalization and transformation
- Encoding categorical variables

These steps improve data quality and enhance model performance.

Feature Engineering and Extraction Module

This module extracts important features that help distinguish **legitimate clicks from fraudulent clicks**.

Examples of engineered features include:

- Click frequency per IP address
- Time interval between clicks
- Device and browser patterns
- Abnormal click behavior
- Geographic click distribution

Feature engineering improves the predictive power of machine learning models.

Machine Learning Model Training Module

In this module, traditional machine learning algorithms are trained using the prepared dataset to detect suspicious click patterns.

Common algorithms used include:

- Random Forest
- Support Vector Machine (SVM)
- Logistic Regression
- Decision Tree

These models classify clicks as **fraudulent or legitimate** based on historical patterns.

Deep Neural Network Detection Module

This module applies deep learning models to capture complex patterns in large-scale clickstream data.

Deep learning techniques may include:

- Deep Neural Networks (DNN)
- Convolutional Neural Networks (CNN)
- Recurrent Neural Networks (RNN)

These models analyze **temporal patterns, user behavior, and high-dimensional data** to detect advanced fraud activities.

Fraud Classification and Decision Module

This module integrates the outputs of machine learning and deep learning models to make final decisions.

Functions include:

- Classifying clicks as **fraudulent or genuine**
- Calculating fraud probability scores
- Flagging suspicious activities for further analysis

This hybrid approach improves detection accuracy.

Alert and Reporting Module

When fraudulent clicks are detected, this module generates alerts for advertisers and administrators.

Outputs include:

- Fraud detection reports
- Suspicious activity logs
- Real-time fraud alerts

This helps prevent financial loss in digital advertising campaigns.

System Monitoring and Dashboard Module

This module provides visualization and monitoring tools for system administrators.

Features include:

- Real-time fraud analytics dashboard
- Performance metrics of detection models
- Click behavior visualization

This module helps maintain system efficiency and transparency.

ALGORITHMS

DECISION TREE CLASSIFIERS

Decision tree classifiers are used successfully in many diverse areas. Their most important feature is the capability of capturing descriptive decision making knowledge from the supplied data. Decision tree can be generated from training sets. The procedure for such generation based on the set of objects (S), each belonging to one of the classes C1, C2, …, Ck is as follows:

Step 1. If all the objects in S belong to the same class, for example Ci, the decision tree for S consists of a leaf labeled with this class

Step 2. Otherwise, let T be some test with possible outcomes O1, O2,…, On. Each object in S has one outcome for T so the test partitions S into subsets S1, S2,… Sn where each object in Si has outcome Oi for T. T becomes the root of the decision tree and for each outcome Oi we build a subsidiary decision tree by invoking the same procedure recursively on the set Si.

GRADIENT BOOSTING Gradient boosting is a machine learning technique used in regression and classification tasks, among others. It gives a prediction model in the form of an ensemble of weak prediction models, which are typically decision trees.[1][2] When a decision tree is the weak learner, the resulting algorithm is called gradient-boosted trees; it usually outperforms random forest.A gradient-boosted trees model is built in a stage-wise fashion as in other boosting methods, but it generalizes the other methods by allowing optimization of an arbitrary differentiable loss function.

K-NEAREST NEIGHBORS (KNN)

- Simple, but a very powerful classification algorithm
- Classifies based on a similarity measure
- Non-parametric
- Lazy learning
- Does not “learn” until the test example is given
- Whenever we have a new data to classify, we find its K-nearest neighbors from the training data

Example

- Training dataset consists of k-closest examples in feature space
- Feature space means, space with categorization variables (non-metric variables)
- Learning based on instances, and thus also works lazily because instance close to the input vector for test or prediction may take time to occur in the training dataset

LOGISTIC REGRESSION CLASSIFIERS

Logistic regression analysis studies the association between a categorical dependent variable and a set of independent (explanatory) variables. The name *logistic regression* is used when the dependent variable has only two values, such as 0 and 1 or Yes and No. The name *multinomial logistic regression* is usually reserved for the case when the dependent variable has three or more unique values, such as Married, Single, Divorced, or Widowed. Although the type of data used for the dependent variable is different from that of multiple regression, the practical use of the procedure is similar. Logistic regression competes with discriminant analysis as a method for analyzing categorical-response variables. Many statisticians feel that logistic regression is more versatile and better suited for modeling most situations than is discriminant analysis. This is because logistic regression does not assume that the independent variables are normally distributed, as discriminant analysis does. This program computes binary logistic regression and multinomial logistic regression on both numeric and categorical independent variables. It reports on the regression equation as well as the goodness of fit, odds ratios, confidence limits, likelihood, and deviance. It performs a comprehensive residual analysis including diagnostic residual reports and plots. It can perform an independent variable subset selection search, looking for the best regression model with the fewest independent variables. It provides confidence intervals on predicted values and provides ROC curves to help determine the best cutoff point for classification. It allows you to validate your results by automatically classifying rows that are not used during the analysis.

NAÏVE BAYES

The naive bayes approach is a supervised learning method which is based on a simplistic hypothesis: it assumes that the presence (or absence) of a particular feature of a class is unrelated to the presence (or absence) of any other feature .Yet, despite this, it appears robust and efficient. Its performance is comparable to other supervised learning techniques. Various reasons have been advanced in the literature. In this tutorial, we highlight an explanation based on the representation bias. The naive bayes classifier is a linear classifier, as well as linear discriminant analysis, logistic regression or linear SVM (support vector machine). The difference lies on the method of estimating the parameters of the classifier (the learning bias).While the Naive Bayes classifier is widely used in the research world, it is not widespread among practitioners which want to obtain usable results. On the one hand, the researchers found especially it is very easy to program and implement it, its parameters are easy to estimate, learning is very fast even on very large databases, its accuracy is reasonably good in comparison to the other approaches. On the other hand, the final users do not obtain a model easy to interpret and deploy, they does not understand the interest of such a technique. Thus, we introduce in a new presentation of the results of the learning process. The classifier is easier to understand, and its deployment is also made easier. In the first part of this tutorial, we present some theoretical aspects of the naive bayes classifier. Then, we implement the approach on a dataset with Tanagra. We compare the obtained results (the parameters of the model) to those obtained with other linear approaches such as the logistic regression, the linear discriminant analysis and the linear SVM. We note that the results are highly consistent. This largely explains the good performance of the method in comparison to others. In the second part, we use various tools on the same dataset (Weka 3.6.0, R 2.9.2, Knime 2.1.1, Orange 2.0b and RapidMiner 4.6.0). We try above all to understand the obtained results.

RANDOM FOREST

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time. For classification tasks, the output of the random forest is the class selected by most trees. For regression tasks, the mean or average prediction of the individual trees is returned. Random decision forests correct for decision trees' habit of overfitting to their training set. Random forests generally outperform decision trees, but their accuracy is lower than gradient boosted trees. However, data characteristics can affect their performance. The first algorithm for random decision forests was created in 1995 by Tin Kam Ho[1] using the random subspace method, which, in Ho's formulation, is a way to implement the "stochastic discrimination" approach to classification proposed by Eugene Kleinberg. An extension of the algorithm was developed by Leo Breiman and Adele Cutler, who registered "Random Forests" as a trademark in 2006 (as of 2019, owned by Minitab, Inc.).The extension combines Breiman's "bagging" idea and random selection of features, introduced first by Ho[1] and later independently by Amit and Geman[13] in order to construct a collection of decision trees with controlled variance. Random forests are frequently used as "blackbox" models in businesses, as they generate reasonable predictions across a wide range of data while requiring little configuration.

SVM

In classification tasks a discriminant machine learning technique aims at finding, based on an *independent and identically distributed* (*iid*) training dataset, a discriminant function that can correctly predict labels for newly acquired instances. Unlike generative machine learning approaches, which require computations of conditional probability distributions, a discriminant classification function takes a data point x and assigns it to one of the different classes that are a part of the classification task. Less powerful than generative approaches, which are mostly used when prediction involves outlier detection, discriminant approaches require fewer computational resources and less training data, especially for a multidimensional feature space and when only posterior probabilities are needed. From a geometric perspective, learning a classifier is equivalent to finding the equation for a multidimensional surface that best separates the different classes in the feature space. SVM is a discriminant technique, and, because it solves the convex optimization problem analytically, it always returns the same optimal hyperplane parameter—in contrast to *genetic algorithms* (*GAs*) or *perceptrons*, both of which are widely used for classification in machine learning. For perceptrons, solutions are highly dependent on the initialization and termination criteria. For a specific kernel that transforms the data from the input space to the feature space, training returns uniquely defined SVM model parameters for a given training set, whereas the perceptron and GA classifier models are different each time training is initialized. The aim of GAs and perceptrons is only to minimize error during training, which will translate into several hyperplanes' meeting this requirement.

CONCLUSION

In the landscape of online advertising, the persistent threat of click fraud looms large, undermining the industry's integrity and siphoning funds away from legitimate advertisers. Our study, aimed at addressing this challenge, embarked on a comprehensive exploration of feature engineering and extraction techniques to discern subtle nuances in click behavior, crucial for distinguishing between fraudulent and genuine clicks. We meticulously evaluated nine ML and three DL models to identify the most effective approaches in combating click fraud. Our findings underscore the robust performance of ML models, particularly after RFE. Notably, DT and RF models surpassed 98.99% accuracy, while GB, Light GBM, and XGBoost achieved accuracy rates of 98.90% or higher. Impressively, models such as

ANN exhibited precision scores exceeding 98%, indicating their adeptness at accurately identifying fraudulent clicks. Simultaneously, our exploration into DL models revealed promising results, with CNN, Deep DNN, and RNN demonstrating robust performance. RNN, in particular, achieved an accuracy of 97.34%, highlighting its efficacy in click fraud detection. Yet, every research has some limitations that motivate further research, and our study is not an exception. However, looking ahead, our findings pave the way for future research directions in ad click fraud detection. These findings are considered the starting point for further improving our approach proposed in this study. A possible future work is by refine and optimize ML and DL models so that we will be able to enhance detection capabilities and reduce the impact of fraudulent clicks. Moreover, in order to develop multi-level security mechanism, advertisers and cyber security experts could be able to use the proposed fraud detection model on both the server and client sides. The model enables real-time tracking and analysis of click patterns across all ad campaigns and may be server-side linked into the ad platform's backend. This would then be handled on the server-side setup using incoming click data, which would then be passed through the model's algorithms to identify any anomalies and immediately flag any possible fraudulent activity. To supplement the server-side analysis and insights, additional information could be obtained from the client side by monitoring the browser or application. In order to provide a much higher level of detail on user activity, the scripts can then collect metadata such as user interaction, device finger printing, and session behavior. This client-side data may then be compared to server-side results to validate the legitimacy of clicks and identify sophisticated fraud patterns that could otherwise go undetected. By taking these aspects into account, an effective proactive fraud detection solution for advertisers is provided by server-side and client-side implementations together. Online learning or incremental learning which allows a periodic updating of the model and adaptiveness in learning from real-time data will further enhance the model to continue evolving toward emerging threats. This dual approach would guarantee that advertisers' investments are adequately safeguarded while creating a more transparent and resilient digital advertising environment. Additionally, more efforts are needed to investigate advanced fraud patterns, including adversarial attacks and evolving strategies, to enhance models' robustness. Incorporating unsupervised methods especially in case of limited labeled data, including anomaly detection and clustering algorithms. Moreover, hybrid models that combine supervised and unsupervised techniques may achieve a balance between accuracy and adaptability in practice. Attention also must be focused on real-world challenges like imbalanced data, latency, and scalability for deployment in various environments

REFERENCES

[[1] Juniper Research, Hampshire, U.K. Quantifying the Cost of Ad Fraud: 2023–2028. Accessed: Jul. 12, 2024. [Online]. Available: https://fraud blocker.com/wp-content/uploads/2023/09/Ad-Fraud-Whitepaper_Juniper - Research.pdf

[2] X. Zhu, H. Tao, Z. Wu, J. Cao, K. Kalish, and J. Kayne, Fraud Prevention in Online Digital Advertising. Cham, Switzerland: Springer, 2017.

[3] A. K. Wood and A. M. Ravel, "Fool me once: Regulating fake news and other online advertising," S. Cal. L. Rev., vol. 91, p. 1223, Jan. 2017.

[4] B. Stone-Gross, R. Stevens, A. Zarras, R. Kemmerer, C. Kruegel, and G. Vigna, "Understanding fraudulent activities in online ad exchanges," in Proc. ACM SIGCOMM Conf. Internet Meas. Conf., Nov. 2011, pp. 279–294.

[5] (2024). Wasted Ad Spend Report 2024. [Online]. Available: https://lp. lunio.ai/wp-content/uploads/2023/09/Lunio_Wasted_Ad_Spend_Report_ 2024_V2.pdf

[6] D. Berrar, "Random forests for the detection of click fraud in online mobile advertising," in Proc. Int. Work. Fraud Detect. Mob. Advert. (FDMA), Singapore, 2012, pp. 1–10. [Online]. Available: http://berrar.com/ resources/Berrar_FDMA2012.pdf

[7] J. H. Yan and W. R. Jiang, "Research on information technology with detecting the fraudulent clicks using classification method," Adv. Mater. Res., vol. 859, pp. 586–590, Dec. 2013, doi: 10.4028/www.scientific.net/amr.859.586.

[8] K. S. Perera, B. Neupane, M. A. Faisal, Z. Aung, and W. L. Woon, "A novel ensemble learning-based approach for click fraud detection in mobile advertising," in Mining Intelligence and Knowledge Exploration (Lecture Notes in Computer Science: Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 8284. Berlin, Germany: Springer, 2013, pp. 370–382, doi: 10.1007/978-3-319-03844-5_38.

[9] C. Phua, E.-Y. Cheu, G.-E. Yap, K. Sim, and M.-N. Nguyen, "Feature engineering for click fraud detection," in Proc. Work. Fraud Detect. Mob. Advert., 2012, pp. 1–10. [Online]. Available: http://palanteer.sis.smu.edu. sg/fdma2012/doc/FirstWinner-Starrystarrynight-Paper.pdf%5Cnpapers2: //publication/uuid/9290A6CF-A861-4058-99F4-D39706B0619A

[10] E.-A. Minastireanu and G. Mesnita, "Light GBM machine learning algorithm to online click fraud detection," J. Inf. Assurance Cybersecur., vol. 2019, pp. 1–12, Apr. 2019, doi: 10.5171/2019.263928.

[11] D. Sisodia and D. S. Sisodia, "Gradient boosting learning for fraudulent publisher detection in online advertising," Data Technol. Appl., vol. 55, no. 2, pp. 216–232, Apr. 2021, doi: 10.1108/dta-04-2020-0093.

[12] A. Dash and S. Pal, "Auto-detection of click-frauds using machine learning Auto-detection of click-frauds using machine learning," Int. J. Eng. Sci. Comput., vol. 10, pp. 27227–27235, Sep. 2020.

[13] R. Mouawi, M. Awad, A. Chehab, I. H. E. Hajj, and A. Kayssi, "Towards a machine learning approach for detecting click fraud in mobile advertizing," in Proc. Int. Conf. Innov. Inf. Technol. (IIT), Nov. 2018, pp. 88–92, doi: 10.1109/INNOVATIONS.2018.8605973.

[14] M. Aljabri and R. M. A. Mohammad, "Click fraud detection for online advertising using machine learning," Egyptian Informat. J., vol. 24, no. 2, pp. 341–350, Jul. 2023, doi: 10.1016/j.eij.2023.05.006.

[15] S. Shaik and V. Kakulapati, "Fraud detection of AD clicks using machine learning techniques," J. Sci. Res. Rep., vol. 29, no. 7, pp. 84–89, Jun. 2023, doi: 10.9734/jsrr/2023/v29i71762.