IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Advertisement Fraud Detection Using Machine Learning

[1]Kamalesh M R, [2]Karna C, [3]Anandha Krishnan P, [4]Princy M

[1]UG Student, [2]UG Student, [3] UG Student, [4]Assistant Professor

[1]Information Technology,

[1]Sri Ramakrishna Engineering College, Coimbatore, India

Abstract: Online advertising is crucial to digital marketing because it allows businesses to effectively reach large audiences. However, because of the increasing number of fraudulent advertisements, there are significant risks for both consumers and advertisers. Cybercriminals commonly hide harmful content, including phishing links, deceptive images, and malicious video ads, behind advertisements that look real in order to trick consumers into watching dangerous content. These deceptive ads have the potential to harm user data, result in financial losses, and erode trust in digital advertising networks. To address this issue, we propose a system that can recognize and block malicious or phishing advertisements on websites. Our approach builds a machine learning model using a dataset of known phishing ad URLs. The trained model is then incorporated into a browser extension that users use while browsing the web, enabling the real-time identification and blocking of harmful ads. This method increases user safety, protects user privacy, and makes online advertising safer.

KEYWORDS—Phishing Detection, Malicious Advertisement, Online Ad Fraud, Browser Extension Security, Machine Learning for Security.

I. INTRODUCTION

Online advertising has become a key component of contemporary marketing tactics in the digital age, allowing companies to easily and effectively reach large audiences worldwide. Digital advertisements have revolutionized how businesses market their goods and services with sophisticated targeting tools, tailored content, and programmatic ad placements. Online advertisements, whether on websites, social media platforms, or search engines, are essential for raising brand awareness, bringing in visitors, and eventually making money.However, the risks that target the digital advertising environment are also evolving. The startling increase in misleading web ads is one of the industry's most urgent problems right now. Ad networks are being used more and more by cybercriminals to disseminate harmful content that looks like genuine ads.

These dangerous advertisements frequently include phishing links, deceptive images, or fake video material, all of which are meant to trick consumers into clicking on them. After engaging, consumers might be taken to malware downloads, scam websites, or phony login pages, endangering their devices, financial information, and personal information.In addition to users, legitimate advertisers and digital platforms are also impacted by this increasing wave of ad-based cyberthreats. The repercussions for consumers can include malware and spyware installation, financial fraud, and identity theft. Fraudulent advertisements erode user trust, reduce engagement, and harm a brand's reputation, according to advertisers and platform providers. Furthermore, if user data is compromised by harmful advertisements, platforms may potentially have compliance problems with data protection laws like the CCPA or GDPR.

Intelligent systems that can recognize and block phishing or fraudulent adverts in real time are desperately needed, given the scope and severity of this problem. Our initiative offers a machine learning-based approach to address this issue by identifying hazardous internet advertisements before users interact with them. We developed a prediction model to identify and flag questionable material by using a carefully selected dataset

of phishing ad links. After that, the trained model was made available as a browser extension, allowing users to easily and instantly identify phishing advertisements while they were online.By keeping an eye on website content and warning users when a potentially harmful advertisement is found, this browser plugin serves as a proactive barrier. Our technology aims to contribute to a more reliable and secure digital advertising environment in addition to improving user safety and data protection. By incorporating intelligent threat detection into routine online browsing, this project ultimately seeks to close the gap between cybersecurity and user experience.

II. LITERATURE REVIEW

Phishing detection and malicious advertisement prevention have been widely researched due to the increasing number of cyber threats emerging through digital advertising platforms. Before initiating our project, an in-depth study of existing systems and methodologies was conducted to understand the various approaches previously employed for detecting harmful advertisements. This included techniques involving URL analysis, content-based filtering, behavior-based models, and machine learning-based classifiers. Through this review, we were able to identify the strengths and limitations of the current systems in use. Most existing tools provide fundamental protection against known threats, however, they often fail to deliver effective real-time detection and have difficulty adapting to evolving attack methods, adaptability to new attack patterns or integration into user-friendly platforms like browser extensions. The insights gained from this study helped shape our approach by highlighting areas where improvements could be made particularly in terms of real-time detection, user accessibility, and proactive threat prevention.

R. A. Alzahrani et al., [1] suggested a method for digital marketing ad click fraud detection based on machine learning and deep learning. They conducted a thorough feature engineering study to find subtle behavioral characteristics that differentiate between authentic and fraudulent user clicks. Nine machine learning (ML) and deep learning (DL) models were tested in the study; Decision Tree (DT) and Random Forest (RF) models achieved accuracy rates higher than 98.99%. Additional models such as XGBoost, LightGBM, and Gradient Boosting also shown remarkable accuracy levels above 98.90%. Artificial Neural Networks (ANN), Convolutional Neural Networks (CNN), Deep Neural Networks (DNN), and Recurrent Neural Networks (RNN) are examples of deep learning models that have shown considerable precision; RNN's accuracy was 97.34%. The results demonstrate how well tree-based and sophisticated learning models detect and prevent click fraud, providing important information for creating trustworthy anti-fraud systems for online advertising.

Vandavasi Baba Mahesh et al., [2] created a machine learning-based method to identify fraud clicking in online advertisements, a problem that has grown more important as social media and digital platforms have grown in popularity. Due to their heavy reliance on pay-per-click (PPC) models, advertising networks are increasingly at risk from fake clicks that deplete advertiser budgets or exaggerate publisher revenue. By developing several machine learning models intended to differentiate between human users and bots, this study tackled the problem. To ascertain the models' performance, analytical methods were applied. The study offers a data-driven strategy to improve efficiency and trust in digital advertising by highlighting the importance of artificial intelligence in cybersecurity and showcasing the ability of machine learning models to successfully combat click fraud.

Benjamin Kirkwood et al., [3] proposed a machine learning-based method for identifying click fraud in online advertising, a problem that is becoming more and more prevalent as internet-based ad platforms grow in size. According to the study, fake clicks, which are frequently produced by automated scripts, can defraud marketers by distorting marketing metrics or draining budgets.

The system differentiates between authentic and fraudulent user interactions by examining the time intervals between clicks and using a variety of machine learning algorithms, including logistic regression, random forest, and neural networks. The TalkingData Ad Tracking Fraud Detection dataset was used to train models, proving to be a successful strategy for improving accuracy and trust in digital advertising campaigns.

Bingzhou Dai et al., [4] presented GemmaWithLoRA, a cutting-edge method for employing large language models (LLMs) to detect click fraud. Using the Gemma-2b model optimized using LoRA technology, the study overcomes the drawbacks of conventional fraud detection techniques, including their low generalization and labor-intensive feature engineering. The model's capacity to extract features and adapt to the pay-per-click

(PPC) advertising ecology is improved by this combination. The model's 83% accuracy rate in extensive testing using the TalkingData2017 dataset demonstrated the promise of LLMs in addressing intricate fraud tendencies in digital advertising contexts.

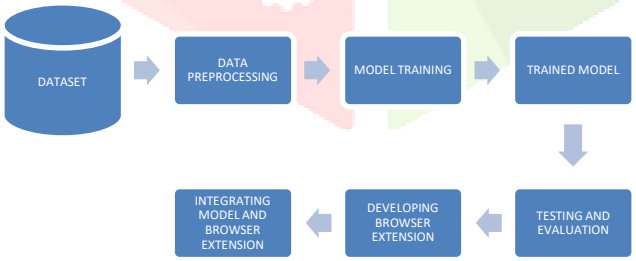
Marouane Dirchaoui and Abdallah Abarda, [5] offered a method for classifying images based on deep learning that can be used to identify phony Arabic online ads on social media sites. The study assessed the performance of three models: a standard CNN, MobileNetV2, and NASNetMobile. It presented the first dataset, which included 914 photos, 457 of which were authentic and 457 of which were fake. The NASNetMobile model fared better than the others, obtaining an F1 score and accuracy of 87.43%, 90.91% precision, and 84.21% recall. Despite their apparent similarities, the study shows how well deep learning can differentiate between genuine and fraudulent advertisements. By growing the dataset and creating algorithms that can analyze both photos and videos, the scientists suggest potential advancements.

Fumin Zhu et al., [6] suggested a novel method based on a tensor recovery mechanism based on Locality-Sensitive Hashing (LSH) for identifying click fraud in online advertising. This approach reconstructs click data into high-rank tensors in order to investigate more intricate linkages and patterns, in contrast to conventional models that interpret multi-dimensional data as flat vectors. Tensor decomposition and transformation techniques are used by the system to more precisely identify fraudulent behavior. The method showed better accuracy and recall rates than current machine learning-based fraud detection systems when tested on real-world datasets, which makes it a viable way to deal with the enormous complexity and scope of click fraud in the Pay-Per-Click (PPC) ecosystem.

Sejal Badere et al., [7] suggested a clever approach that uses deep learning models (CNN-BiGRU and CNN-BiLSTM) in conjunction with Natural Language Processing (NLP) methods like tokenization and count vectorization to identify fraudulent job postings. The models showed excellent accuracy in differentiating between real and fake job listings after being trained using data gathered from Google Dataset Search. In particular, the system outperformed earlier models by 5%, achieving an accuracy rate of 97%. This strong performance demonstrates the system's ability to give job seekers a safer online experience and guides the creation of cutting-edge fraud detection systems for industries including social media, government, and digital employment.

III. Model Specifications

A. *Block Diagram*



B. *Data Collection*

Gathering a thorough and organized dataset of links to both genuine and phishing advertisements was the first stage of the research. In order to guarantee efficient model training and precise classification, the dataset has to have a balanced representation of dangerous and secure URLs in addition to pertinent attributes that mirror typical phishing attempt patterns.

Data Source:

There are 235,795 URLs in the sample, of which 134,850 are authentic and 100,945 are phishing. Since the majority of the URLs in the dataset are current, it is extremely valuable for identifying contemporary phishing tactics.

Contents:

100,945 phishing URLs and 134,850 authentic URLs make up the dataset's total of 235,795 URLs. Since the majority of the included URLs are recent, the dataset is extremely pertinent for identifying contemporary phishing tactics.

Feature Representation:

The associated web pages' source code and URL structure were both examined for features. These characteristics aid in identifying structural and behavioral trends frequently observed in phishing advertisements. Among the noteworthy aspects are:

- The URLCharProb analyzes the probability distribution of characters in the URL.
- The URLTitleMatchScore compares the page title with the domain to identify inconsistencies.
- The CharContinuationRate measures character repetition or unusual sequences in the URL.
- The TLDLegitimateProb estimates the legitimacy based on the top-level domain (TLD).

C. Data Preprocessing

Cleaning the dataset and extracting valuable features for model training came next once it was collected.

Data Cleaning:

- To avoid biased learning, duplicate records were eliminated. If necessary, rows were discarded, or mean or median imputation was used to address missing values.
- To maintain uniformity, URL fields were normalized by changing the content to lowercase and removing any extraneous spaces or characters.

Feature Extraction:

- To assist the algorithm in identifying phishing tendencies, pertinent features were taken from every URL.
- These consist of the URL length, the number of dots (.), and the slashes (/); larger or more complicated URLs are frequently a sign of questionable activity.
- HTTPS is present; trustworthy websites usually employ secure protocols.
- Top-Level Domain (TLD) — encoded numerically using LabelEncoder to help detect uncommon or less often used domains.
- Commonly utilized in malicious or obfuscated URLs include special characters, the amount of numbers, and equals signs (=).

D. Model Training

a. Extreme Gradient Boost (XGBoost)

Model Selection and Justification:

The XGBoost classifier was used as the main model in order to efficiently identify fraudulent and phishing adverts. XGBoost is very well-suited for problems involving unbalanced datasets, such as phishing detection, and is quite effective for structured data. It is a reliable option for high-accuracy classification due to its capacity to recognize non-linear patterns and support for regularization approaches (L1 and L2).

Feature Representation:

Features taken from each URL, such as URL length, number of dots or slashes, HTTPS presence, Top-Level Domain (TLD), encoded with Label Encoding, special characters, digit count, and usage of "=" symbols, were used to train the model. These characteristics are typical markers of phishing or dubious ad links.

Model Initialization and Training:

In order to maximize performance, the model was first set up with default hyperparameters and then adjusted by cross-validation. By reducing the error of earlier iterations, gradient boosting was utilized to gradually enhance model predictions. L1 and L2 regularization were also used in training to lessen overfitting and enhance generalization to fresh data.

Optimization Strategy:

Algorithms based on gradient descent were employed to maximize the learning process. To guarantee that the model achieved excellent precision and recall without compromising training stability, the learning rate and other crucial hyperparameters were carefully adjusted.

Batch Processing:

The dataset was divided into manageable portions during the data pretreatment and training pipeline to increase computing efficiency and enable smoother evaluation, even though XGBoost doesn't rely on batch processing like deep learning models do.

b.K-Nearest Neighbors (KNN) Algorithm

As a baseline model for phishing detection, KNN is an easy-to-understand algorithm. By determining the distance (such as Euclidean) between data points, it classifies data and makes predictions based on the nearest neighbors' majority class.

However, feature scaling is crucial to KNN performance; if features are not appropriately normalized, predictions may be skewed. Furthermore, because KNN needs to calculate distance for each prediction, it might be computationally wasteful, particularly in real-time applications. XGBoost, on the other hand, is a better option for phishing detection since it creates a model up front, enabling predictions that are significantly quicker and more accurate.

E. Design and Structure of Browser Extension

Real-time scanning of links and ads for phishing detection is made possible by the browser extension's efficient operation within the browser environment. The extension is made up of a number of parts, each of which is in charge of carrying out particular duties to guarantee user interaction and seamless operation.

1. background.js (Service Worker)

The background.js file serves as the extension's service worker, managing necessary background tasks. It controls installation events, making sure that when a user downloads or updates an extension, it is initialized correctly. In order to ensure smooth functioning without interfering with the user's experience, it also manages background operations including state maintenance, API interaction, and user setting storage.

2. content.js

The user's visited URLs are immediately injected with the content.js script. Its main duty is to continuously search for possible phishing content by scanning links and advertising. In order for this script to function, it must communicate with the document object model (DOM) of the page, extract URLs, and compare the results to the phishing detection model that has been trained. The detection of a questionable link or advertisement may result in the user being notified or the material being highlighted, among other suitable steps.

3. popup.html, popup.js, and styles.css

These files make up the extension's user interface (UI), which gives users an easy-to-use and straightforward way to interact with it:

popup.html: Specifies the buttons, toggles, and status indicators that make up the user interface's structure and layout.

popup.js: Includes the logic for the user interface's interactivity, such as the ON/OFF toggle feature that lets users turn on or off phishing detection whenever they choose.

styles.css: Provides the visual styling, guaranteeing that the user interface is both aesthetically pleasing and easy to use, with unambiguous signs of the status of the extension.

4. manifest.json

A crucial configuration file that specifies the metadata and permissions for the extension is manifest.json. It contains important details including the name, version, and description of the extension in addition to outlining the permissions needed to operate it (e.g., access to specified websites or reading content on a page). Along with ensuring that the extension loads and integrates with the browser correctly, this file also specifies the extension's configuration, including background and content scripts.

5. Real-time Interaction with Web Content

In order to scan online sites, the plugin looks for important HTML elements like pictures, iframes, and anchor tags (), which could include harmful links or advertisements. These components are carefully inspected in order to identify any possibly dangerous URLs.

After identifying URLs, the extension forwards them to a Flask-based phishing detection API, where a trained model is used to classify them. The extension immediately alerts the user to the questionable advertisement or link on the webpage if it detects phishing elements. By ensuring that phishing risks are quickly detected and reported, this real-time interaction improves user security when browsing.

F. Integrating Browser Extension and Model

For real-time phishing ad detection, integrating the browser extension with the machine learning model is essential. The process guarantees that users are swiftly informed and that suspected dangerous information is promptly discovered. The XGBoost model, a Flask-based API, and the browser extension communicate seamlessly to accomplish this integration.

1. Capturing and Sending URLs

The browser plugin keeps looking for any phishing advertisements on websites. It detects questionable ad elements including iframes, anchor tags (), and pictures that can have harmful links in them. The extension uses fetch and POST requests to transmit the URLs to the Flask API in real-time after detecting these components. This guarantees that there is no discernible delay for the user and that the URLs are analyzed for classification right away.

2. Feature Extraction and Model Prediction

The Flask API processes the data by extracting pertinent information from the URLs after receiving them. These characteristics may consist of:

- Length of URL
- The quantity of unique characters
- TLD, or top-level domain
- HTTPS is present.

3. User Notification and Real-time Alerts

The browser extension processes the answer once it has received the prediction from the Flask API. The model's output causes the browser's popup interface to instantly display an alert. The extension alerts the user to the presence of a phishing advertisement on the page if the URL is deemed to be phishing. The plugin gives the user confidence that the advertisement is secure if the URL is authentic. This enables people to interact with potentially harmful content on the internet in an informed manner.

IV. CONCLUSIONS AND FUTURE SCOPE

i. Conclusion

This project effectively illustrates how to combine a browser extension and a machine learning model to identify fraudulent and phishing ads in real time. The extension effectively searches websites for dubious links and gives users prompt feedback by utilizing XGBoost for classification and connecting it with a Flask-based API. Because the method strikes a balance between accuracy and performance, consumers can interact with online material safely without being concerned about hazardous or misleading advertisements. By proactively identifying phishing risks, this technology seeks to improve consumer confidence and web security while making browsing safer. The user-friendly interface of the browser extension offers unambiguous notifications, and the real-time interaction between the extension and the phishing detection model guarantees that users are shielded from dangerous advertisements. This research provides a workable way for regular people to stay safe from phishing scams by laying the foundation for integrating cutting-edge machine learning algorithms into browser-based security measures.

ii. Future Scope

Even while the current system detects phishing ads effectively, there are a few possible areas for future development and enhancement:

Model Improvement:

To further increase accuracy and manage intricate phishing patterns, the present XGBoost model could be improved by adding more sophisticated machine learning methods, such as deep learning models (e.g., CNNs for URL categorization or LSTMs for sequential data).

Feature Expansion:

Other elements, including picture recognition, textual content analysis, or behavioural patterns that could help detect harmful ads, could be taken from the actual content of the webpage. This would offer a detection method that is more thorough.

Cross-browser Compatibility:

Currently, the browser extension can be tailored for a specific browser, such as Chrome. Reaching a wider audience would be facilitated by making the extension compatible with more browsers, including Firefox, Edge, and Safari.

Real-time Feedback and Continuous Learning:

The model may be continuously improved over time by putting in place a feedback loop where users could submit false positives or negatives. This could involve active learning strategies, in which the model is frequently re-trained using fresh data or user input.

User Customization:

Giving users the choice to alter the phishing detection level (low, medium, or high sensitivity, for example) would increase the extension's adaptability to user preferences and better meet a range of security requirements.

Phishing Detection for Other Platforms:

The system's reach would be expanded to detect phishing in email links, social media posts, and other digital content in addition to web adverts. This would give users a more complete protection tool.

Collaboration with Ad Networks:

By combining a greater range of phishing-ad data, collaborating with advertising networks may assist collect more information and enhance the detection model. This would help combat online fraud on a worldwide scale

V. RESULTS AND DISCUSSION

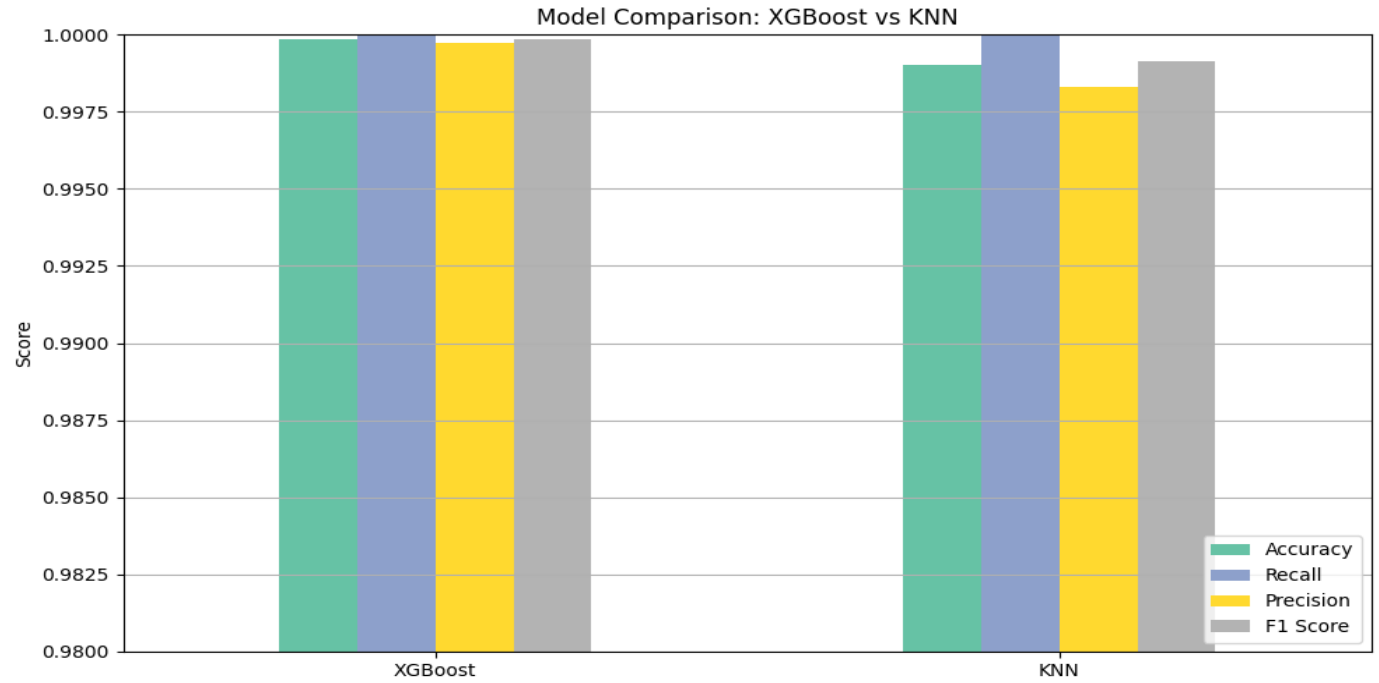


Figure 1.1 Comaprison Between XGBoost and KNN

REFERENCES

[1] A. Dash and S. Pal. (2020) "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.

[2] A. Purwar, A. K. Jain, I. Chawla, I. Gupta, M. Raj and D. Jain. (2024) "Click fraud detection using ensemble classifier", Proc. Int. Conf. Artif.-Bus. Anal. Quantum Mach. Learn., pp. 15-23.

[3] B. Kirkwood, M. Vanamala and N. Seliya. (2024) "Click Fraud Detection of Online Advertising Using Machine Learning Algorithms" on IEEE International Conference on Electro Information Technology, Eau Claire, WI, USA.

[4] D. Sisodia and D. S. Sisodia. (2021) "Gradient boosting learning for fraudulent publisher detection in online advertising", Data Technol. Appl., vol. 55, no. 2, pp. 216-232.

[5] J. D. Kelleher, B. Mac Namee and A. D'arcy. (2020) Fundamentals of Machine Learning for Predictive Data Analytics: Algorithms Worked Examples and Case Studies, Cambridge, MA, USA:MIT Press.

[6] L. Singh, D. Sisodia, K. Shashvat, A. Kaur and P. C. Sharma. (2023) "A reliable click-fraud detection system for the investigation of fraudulent publishers in online advertising" in Applied Intelligence in Human-Computer Interaction, Boca Raton, FL, USA: CRC Press.

[7] M. Aljabri and R. M. A. Mohammad. (2023) "Click fraud detection for online advertising using machine learning", Egyptian Informat. J., vol. 24, no. 2, pp. 341-350.

[8] Malak Aljabri, Rami Mustafa, A. Mohammad. (2023) "Click fraud detection for online advertising using machine learning"on Egyptian Informatics Journal Volume 24, Issue 2.

[9] Neeraja, Anupam, Sriram, Subhani Shaik and V. Kakulapati. (2023) "Fraud Detection of Ad Clicks Using Machine Learning Techniques"on Journal of Scientific Research and Reports, Volume 29, Issue 7, Page 84-89.

[10] R. A. Alzahrani, M. Aljabri and R. A. Mustafa Mohammad. (2025) "Ad Click Fraud Detection Using Machine Learning and Deep Learning Algorithms" in IEEE Access, vol. 13, pp. 12746-12763.

[11] R. Dekou, S. Savo, S. Kufeld, D. Francesca and R. Kawase. (2021) "Machine learning methods for detecting fraud in online marketplaces", Proc. CEUR Workshop, vol. 3052, pp. 3-7.

[12] R. Mouawi, M. Awad, A. Chehab, I. H. E. Hajj and A. Kayssi. (2018) "Towards a machine learning approach for detecting click fraud in mobile advertizing", Proc. Int. Conf. Innov. Inf. Technol. (IIT), pp. 88-92.

[13] S. Hong and H. S. Lynn. (2020) "Accuracy of random-forest-based imputation of missing data in the presence of non-normality non-linearity and interaction", BMC Med. Res. Methodol., vol. 20, no. 1, pp. 1-12.

[14] S. Shaik and V. Kakulapati . (2023) "Fraud detection of AD clicks using machine learning techniques", J. Sci. Res. Rep., vol. 29, no. 7, pp. 84-89.

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