


Review

AI-Based Techniques for Ad Click Fraud Detection and Prevention: Review and Research Directions

Reem A. Alzahrani ^{1,*} and Malak Aljabri ² 

¹ SAUDI ARAMCO Cybersecurity Chair, Department of Computer Science, College of Computer Science and Information Technology, Imam Abdulrahman Bin Faisal University, Dammam 31441, Saudi Arabia

² Department of Computer Science, College of Computers and Information Systems, Umm Al-Qura University, Makkah 21955, Saudi Arabia

* Correspondence: 2210500199@iau.edu.sa

Abstract: Online advertising is a marketing approach that uses numerous online channels to target potential customers for businesses, brands, and organizations. One of the most serious threats in today's marketing industry is the widespread attack known as click fraud. Traffic statistics for online advertisements are artificially inflated in click fraud. Typical pay-per-click advertisements charge a fee for each click, assuming that a potential customer was drawn to the ad. Click fraud attackers create the illusion that a significant number of possible customers have clicked on an advertiser's link by an automated script, a computer program, or a human. Nevertheless, advertisers are unlikely to profit from these clicks. Fraudulent clicks may be involved to boost the revenues of an ad hosting site or to spoil an advertiser's budget. Several notable attempts to detect and prevent this form of fraud have been undertaken. This study examined all methods developed and published in the previous 10 years that primarily used artificial intelligence (AI), including machine learning (ML) and deep learning (DL), for the detection and prevention of click fraud. Features that served as input to train models for classifying ad clicks as benign or fraudulent, as well as those that were deemed obvious and with critical evidence of click fraud, were identified, and investigated. Corresponding insights and recommendations regarding click fraud detection using AI approaches were provided.

Keywords: click fraud; artificial intelligence; machine learning; deep learning; click fraud detection features



Citation: Alzahrani, R.A.; Aljabri, M. AI-Based Techniques for Ad Click Fraud Detection and Prevention: Review and Research Directions. *J. Sens. Actuator Netw.* **2023**, *12*, 4. <https://doi.org/10.3390/jsan12010004>

Academic Editor: Mingjun Xiao

Received: 13 November 2022

Revised: 20 December 2022

Accepted: 25 December 2022

Published: 31 December 2022



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Advertising campaigns on websites and smartphone applications are examples of web services that have become common in people's lives. Campaign providers follow the pay-per-click (PPC) model, which charges advertisers for each click on an ad link. The PPC model is one of the fundamental approaches for funding and supporting websites. Advertising campaigns aim to publish ads on relevant web pages in order to increase profit; clicks on these ads may be natural and innocent from web users, but they may also be malicious clicks made by humans or automated software developed by competitors for fraudulent purposes, such as profiting from an organization or charging high fees from advertisers. According to Google Adwords statistics, the average cost of a click for Google Ads is \$0.89, and Google earned \$209.49 billion from advertising in 2021 [1].

With the growing number and diversity of malicious bots that exploit fraudulent auto-clicks, much research has explored this topic and attempted to identify and predict whether fraudulent clicks are made. Throughout the past few years, artificial intelligence (AI) techniques have become increasingly prevalent in cyber security, efficiently contributing to detect and prevent a broad range of threats and attacks [2–5]. Various AI models are used to detect click fraud and assess whether a click is legitimate or illegitimate when a user or computer program clicks on an ad. However, despite advertisers' efforts to stop fraud, recent statistics reveal that the problem of click fraud is widespread and is predicted

to grow in the future. According to the latest figures, the total cost for advertisers of fraudulent clicks reached around \$42 billion in 2021, with click fraud affecting 90% of PPC ad campaigns [6]. With the development of botnets that leverage fraudulent clicks, this issue must be investigated in depth in order to address it.

In this study, all solutions developed and published in the last 10 years that mainly use AI, including machine learning (ML) and deep learning (DL), for the detection and prevention of click fraud were examined. Various models and systems for PPC identification (benign or fraudulent) have been developed using different models. In this study, we also determined the features that were used as input to train models for classifying clicks, as well as which features were considered explicit and crucial indicators of click fraud.

The remainder of this paper is organized as follows. A background on ad clicks and click fraud is presented in Section 2. Related work on AI-based click fraud detection approaches is reviewed in Section 3. Commonly used public datasets in the field of click fraud detection are discussed in Section 4. The features used, and the most essential indicators for identifying click fraud are presented in Section 5, and the study recommendations are provided in Section 6. Finally, Section 7 concludes the study.

2. Background

2.1. E-Commerce

Electronic commerce, or e-commerce, is an electronic mode of business in which the selling and buying of products or services happen over the Internet [7]. This mode of business also allows for the transfer of funds and data using the Internet. According to Khan [7], e-commerce is a business transaction that happens over public electronic networks or the Internet. There has been widespread use of e-commerce in the last two decades, but it boomed particularly during the COVID-19 pandemic. E-commerce accounted for a total of 5% of retail sales before the pandemic. By the start of 2020, e-commerce accounted for 16% of overall retail sales, and this value continues to increase [8].

E-commerce offers business-to-business, consumer-to-consumer, and business-to-consumer business opportunities. A typical e-commerce framework works in the following ways:

- Businesses look for suitable online advertisers to advertise their brands, products, and services;
- Online advertisers, in conjunction with online advertisement providers, place the online ads of the business over the Internet;
- Providers then run different ad campaigns to target the end users of the products and services advertised by businesses;
- By following the links of businesses and companies provided in ad campaigns, users can browse the e-stores of these businesses and companies to purchase their products and services.

This is how e-commerce plays an important role in the growth of businesses. With the help of e-commerce, businesses can target their customers beyond geographic limitations, which is nearly impossible without e-commerce. Online advertisements are critical components of e-commerce. They can help businesses reach their customers [9].

2.2. Online Advertising Architecture

Online advertisement is a marketing strategy that aims to target potential customers for businesses, brands, and companies over various online platforms, such as social media [10]. A typical online advertising architecture comprises the following components:

2.2.1. Advertiser

An online advertiser is an entity that uses website content and online ads for marketing purposes. The tasks of an online advertiser vary according to client requirements. However, the primary roles and responsibilities of an online advertiser include creating content and enhancing users' interactions with the company or brand using various social media platforms and search engine optimization [10].

2.2.2. Publisher/Provider

Online advertising controllers or providers are entities that provide pathways for companies to advertise their products and services [11]. They are often called digital marketing agencies. Google AdSense, Media.net, Advertising.com, and Propeller Ads are some of the most common online advertising controllers or providers.

2.2.3. Advertising Campaigns

An advertising campaign is a series of advertisements that are designed to target a specified audience at specific times and locations in order to promote brand awareness, improve sales, and so on. Advertising campaigns carried out on various online platforms, such as Google, Facebook, and Twitter, are called online advertising campaigns. According to [12], these campaigns are crucial for successful e-commerce strategies.

2.2.4. Ad Networks

Ad networks are interconnections between businesses and companies that are willing to advertise their brands, products, and services with online platforms that are ready to host these ads. One of the most promising features of ad networks is that they collect ad space and match it to advertisers' requirements [13].

2.2.5. Users

Users are the target of advertisers. Advertisers target specific communities of users to grow their awareness of particular brands, businesses, products, and services. Online advertisers use different strategies, often equipped with advanced AI and ML algorithms, to target the most relevant and accurate group of users who can positively contribute to the growth of a business [14]. For example, if a particular user searches for smartwatches on Facebook, the Facebook ad campaign will show them ads related to smart watches after analyzing the user's interests.

2.2.6. Attackers

Attackers are adversary entities with the primary aim of harming online advertisers, businesses, and users. They can perform various attacks, such as man-in-the-middle attacks, distributed denial of services, information theft, and backdoor access attacks. The motivations of attackers can be either the financial or reputational loss of businesses or online advertisers [15].

Figure 1 demonstrate the high-level architecture of online advertising, which carried out in the following order:

- Service request: Initially, when a user requests a service offered by a publisher (1), the publisher receives it immediately;
- Service response: Upon receiving that request, it returns the original content/service requested (2);
- Ad request: Moreover, it asks the ad network to present the user with ads that match their query and profile as results for their search (3);
- (4) Ad matching: In the real-time auction that takes place on the ad exchange, the ad network uses profile information to determine which ads will generate the most revenue;
- Ad redirects: Upon selecting an advertisement or series of advertisements, the ad network directs the provided information to the user, who then instructed by the provider to fetches the ads (5–6);
- Ad fetching and retrieval: A final step is for the user to select, fetch (7), retrieve (8) the advertisement they wish to view.

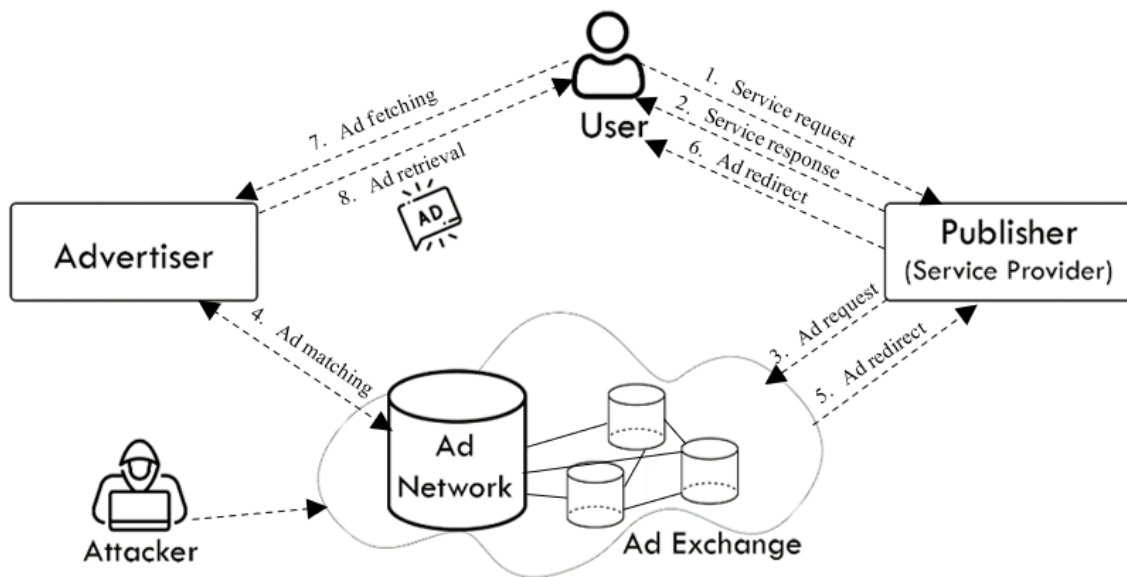


Figure 1. High-level architecture of online advertising.

2.3. Typical Online Advertising Services

Services that advertise businesses online are called online advertising services. They use social media platforms and search engines. The range of typical online advertising services covers the PPC model to social media and programmatic advertising services [16]. The creation of ads, the management of ads, and performance reporting are some of the common functionalities included in online advertising services. They have variable price ranges depending on several factors, such as the business industry, ad agency, and ad network [17]. Some common online advertising services include the following:

- PPC;
- Programmatic advertising services;
- Social media advertising services.

2.3.1. Pay-per-Click Advertising Services

In PPC, marketers place bids on phrases and keywords that can be used to trigger their advertisements. Thus, PPC is a paid advertising service. The optimization of PPC is a good option for marketers who want rapid qualified traffic to their online stores or sites [18]. This is because PPC starts giving results soon after an ad goes live. A common example of a PPC advertising service is search engine advertising.

2.3.2. Programmatic Advertising Services

Programmatic advertising services display ads to users on the basis of their interests [19]. This advertising service is automated. It learns the online behavior of users to display ads of targeted items that are of interest to them. Therefore, programmatic advertising services are sometimes called online behavioral advertising [20]. This type of advertising service can target individual users and seems to be a perfect marketing tool for marketers.

2.3.3. Social Media Advertising Services

Social media advertising services are a great way to build an interactive relationship with users who have an active presence on social media platforms. The use of social media sites, such as Instagram and Facebook, continues to increase [21]. Therefore, social media platforms are attractive markets for advertisers. Most social media platforms, such as Facebook, YouTube, and Instagram, have introduced their own advertising services.

2.4. Online Advertising and Fraud

Online advertising fraud is a false representation of the number of times a digital advertisement is clicked on or displayed [22]. This has an impact on online advertising campaigns because they are designed to show information about a product or service to only those people who are most likely to want it. In exchange, the publisher receives an advertising fee, and the advertiser's revenue increases. Competitors or dishonest publishers frequently engage in digital/online ad fraud by using automated bot traffic to click on advertisements repeatedly. In this case, the advertiser pays the publisher and the advertising platform, so this practice has the potential to scam a company with substantial amounts of money.

Many websites depend on advertising sales to make money, and many other businesses spend much money on marketing initiatives in an effort to boost sales [13]. The system relies heavily on automated exchanges that pair advertisements with potential clients. These automated procedures are used by digital ad fraudsters, who pose as real users and steal money by showing advertisements to fictitious consumers; this is how both advertisers and publishers are impacted by scams or fraud.

2.4.1. Click Fraud

For most businesses, digital marketing—and PPC in general—has replaced traditional forms of promotion. From small businesses to multinational corporations, everyone can benefit from having access to a broad market online. Roughly four billion individuals who use the Internet every day make purchases, so a well-targeted PPC campaign can mean the difference between drowning and swimming [23]. Five billion searches are made on Google every day, indicating the significance of PPC advertising; this enormous amount of traffic and the money involved make it an ideal target for fraud. Furthermore, click fraud has surpassed credit card fraud as the most expensive type of fraud performed annually.

The issue of fraudulent ad clicks is serious, especially when we consider that the cost of some keywords in Google Ads can be as much as \$50 or over \$100 per click [24]. In fact, the volume of click fraud can soon cause issues for the average advertiser, even with clicks costing around \$1 each. In 2017, one in five clicks on a PPC ad campaign was thought to be false in some way [25]. In the intervening years, both the methods and the scale of Internet fraud have evolved significantly.

2.4.2. Source of Click Fraud

Large-scale click fraud is frequently automated using a bot or another computer that copies a real person visiting a website. The bot continually clicks on an advertisement in an effort to fool a platform into believing that it is a person who wants to buy whatever the advertisement is selling [26]. Large numbers of clicks coming from one computer are likely to be noticed by a click fraud victim, and advertising networks and advertisers may also find this traffic suspicious. However, fraudsters can circumvent this by using a virtual private network (VPN) to route bot communication across a variety of constantly changing Internet protocol (IP) addresses. Additionally, they can engage in click fraud by using numerous computers in various locations, which are the main sources of high-, medium-, or low-volume clicks [27].

3. Literature Review

Along with other notable existing solutions for detecting click fraud, previous efforts have specifically used AI-based techniques, such as ML and DL. The purpose of these is to protect advertisers from being charged with fraudulent click expenses that cost them significantly and thus affect the quality of their ad campaigns. Previous findings have shown that the application of AI techniques to identify legitimate and illegitimate ad clicks yields good results, so they are deployed on both server and user sides to detect and avoid fraudulent clicks.

In this section, we review previous studies conducted to detect and prevent click fraud. Our review is based on the following criteria:

- Case studies that leveraged AI techniques, including ML and DL;
- Studies that investigated the detection of click fraud in the advertising industry (There are several types of fraud that harm the advertising industry, including impressions fraud, publisher fraud, and click fraud. We will mainly examine studies on click fraud);
- Conducted in the last 10 years (2012–2022);
- Written in English.

The taxonomy followed in this paper is illustrated in Figure 2.

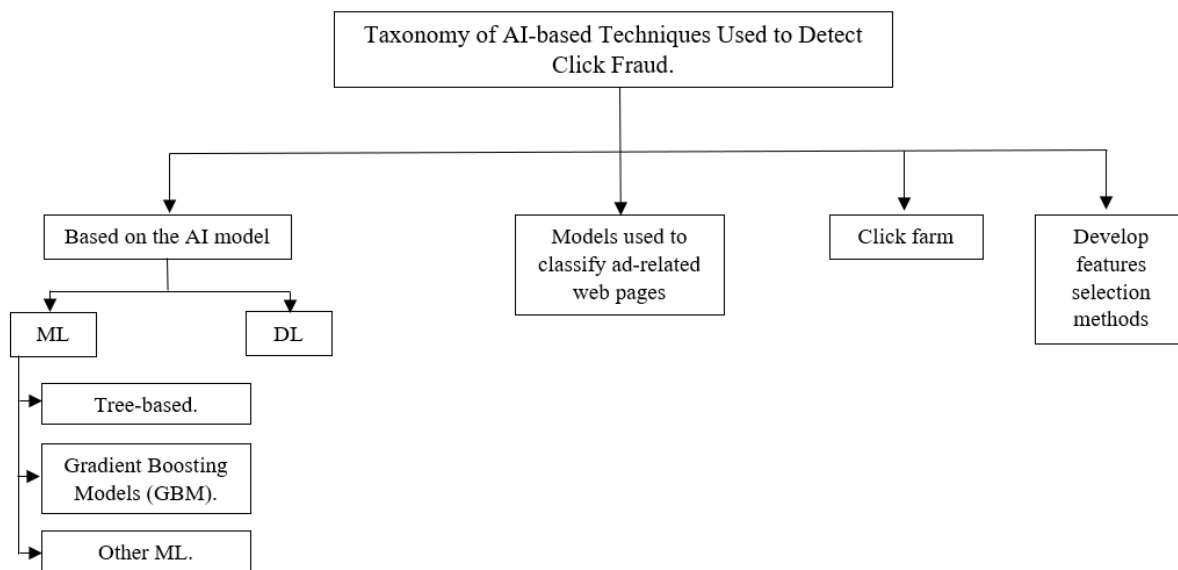


Figure 2. Taxonomy of the Literature Review.

3.1. ML-Based Click Fraud Detection Techniques

3.1.1. Tree-Based Techniques

To determine the source of a click, most previous solutions relied on ML techniques. There were many tree-based models that performed extremely well. Traditional tree-based models can be either Decision Trees (DT) or Random Forests (RF) or Extremely Randomized Trees (ERT), etc. The former can be used to build all tree-node models, while the two latter builds many decision trees in parallel by impling different strategies. Moreover, other versions of tree-based approaches have been employed as well.

Li et al. [28] developed the MadTracer system, which considers ad infrastructure and involves many ad-related parties, such as ad delivery paths and ad context. The system detected a number of attacks, which were later classified as drive-by download, scam, and click fraud, by using detection rules based on a DT classifier. A previously unknown type of click fraud was observed (by Google Safe Browsing and Microsoft Forefront combined)—a fake ad click that leads the visitor to a landing page without the user seeing or clicking on the ad. MadTracer which is a browser-based solution uses knowledge about harmful ad paths and the behavioral features associated with them to activate an alarm and detect when a user reaches a suspect ad path. Berrar [29] used Random Forests (RFs) with skewed bootstrap sampling to determine a publisher’s status (fraudulent vs. legal) based on clicks associated with its ad campaigns. Additional features, such as the click profile (the time gap between subsequent clicks), were extracted and were available in two stages: (1) long click profile and (2) short click profile; clicks made with the same IP and URL were also investigated. Two tests were carried out to assess the usefulness of the provided features; the first model contained the raw features in the dataset and the long click profile, whereas the second model included the short click and URL profiles. With an average accuracy of

49.99%, the first model performed best in the validation set, and in the test set, it had an accuracy of 42.01%. Yan and Jiang [30] conducted a study to determine suitable classifier models for detecting click fraud in commercial and advertising logs. Several classifiers were trained using numerical features, such as IP and the number of clicks at different time intervals during the day, as well as statistical features, such as the standard deviations of clicks in various periods (morning, night, etc.); RFs, Bayesian networks (BNs), decision tables, REPTree, and naive Bayes (NB) were then used to classify clicks. The MapReduce paradigm was applied to develop the preprocessing and feature extraction modules, and the findings showed that tree-based approaches outperformed the Bayes approach because of the imbalanced distribution of fraudulent versus valid clicks (the majority class). In a study by Perera et al. [31], a novel ensemble model was introduced based on user behavior from click patterns, and valuable new features derived from raw ones that could not be used in their original forms to detect click fraud, such as 20 features derived from time at (i.e., average, max, variance, and skewness), were calculated for clicks across various time intervals. Several classifiers were attempted to build an ensemble model that includes the six best-performing classifiers: bagging with J48, bagging with repetition tree (REPTree), bagging with RF, MetaCost with J48, LogiBoost with J48, and random subspace with J48; these proved successful on validation and test sets and demonstrated their generalizability with an accuracy of 59.39%. Oentaryo and Lim [32] focused on extracting various types of features, including basic features, such as publisher account and click-per-ad ratio, as well as spatial features, such as click fraction of the top 20 countries, as input when training ML models, such as Logistic Regression (LR) and extremely randomized trees. Overall, the findings indicated that temporal (time series) and spatial (clicks from specific risky countries) features are effective fraud markers, and ensemble methods provide beneficial solutions to highly unbalanced classification tasks with varied attribute values and noisy/missing patterns. Perera [33] indicated that understanding click patterns would greatly aid in the challenging task of detecting and preventing fraudulent clicks. Support vector machines (SVMs), LR, BNs, multilayer perceptron (MLP), and several DT algorithms, including C4.5, REPTree, and RF, were used. Under-sampling was also applied to overcome the problem of dataset imbalance, in which majority class records are used with all minority class records. Addresses, bank accounts, and other features unrelated to publisher or click behavior were excluded. A number of fraud detection features used in Google's fraud detection system, as well as most fraud detection mechanisms applied to PPC ads, were considered to extract useful information. Feature selection (FS) techniques were ignored because using all features resulted in the highest accuracy. It was found that using ensemble and sampling techniques separately does not achieve good results; however, when these techniques were combined, performance improved significantly. Bagging with C4.5 + cluster-based sampling was selected as the final model. The author stated that temporal features (e.g., the click ratio in 5 min intervals) and spatial features are critical indicators of fraud detection. Oentaryo et al. [34] discussed the top five entries (for the top five teams) in a click fraud detection competition using the FDMA 2012 dataset, including what we reviewed earlier in [26,28,29,32,35]. One of the teams pointed out that to identify multiple fraud patterns, they must first extract the publisher's characteristics from a variety of statistics, such as mean, standard deviation, and count, and then use single ML models, such as the Functional Trees and Reduced Error Pruning Tree (REP tree), as well as ensemble models, such as RF and rotation forest. However, the ensemble of ensemble models achieved the highest accuracy at 52.3%. Xu et al. [36] presented an interesting method for distinguishing between human and bot clicks, which included checking whether the user's browser supported JavaScript and the tracking and monitoring of mouse movement. If there is mouse movement and JavaScript support, they are real users (and several functionality tests will be performed to validate this); otherwise, they are bots. The authors also evaluated clicks from suspicious websites and IP addresses listed on blacklists, along with the user's browsing behavior, as important indicators for detecting bot clicks. On the advertiser's side, the C4.5 model was used to classify and

evaluate clicks in real time for a 10-day campaign, achieving an impressive accuracy of 99.1%. He et al. [37] reported that selecting the right features and models is essential for the efficient detection of click fraud. They built a solid model by combining DTs and LR. These features were divided into two categories: historical and contextual. The latter included features such as the device used and the page on which the advertisement was displayed, whereas the former included data related to the user's behavior and interactions with the advertisement, such as the CTR in the previous week. Contextual features are especially important to new users or ads, but historical features play a more significant role and provide a more detailed description of click identification. The results showed that as the time difference between the training and test sets increased (weekly training and evaluation), the prediction accuracy decreased, whereas it increased with daily training and testing. Ravi [38] investigated touch fraud in mobile gaming apps by using the C4.5 model to classify ads that are tapped or pressed without the user's knowledge and that are uninteresting to them. In terms of features, the app's features, such as app ID, release date, current version, and rating, were extracted, as well as the app developer's features, such as the number of apps developed and the ratings of these apps. The author also introduced a new set of features related to ad control location-based features, which included observing ad constraints in the apps and verifying their implementation in each app. For example, there is a constraint state in which the number of ads on a mobile screen should not exceed one ad, and there is a need to ensure the visibility of the ad so that it is not hidden behind a button, an image, or other similar objects. Violation of these constraints will almost certainly result in touch fraud. The proposed features improved the classifier's performance across all metrics. Beránek et al. [39] used ML techniques, including NBs, DTs, and SVMs, in their work. They analyzed click time from the available features in the dataset and extracted multiple comprehensive features from it, such as time of day, period of day, and type of day. These features were then combined into a timeprint. They ran several experiments to evaluate how accurate a user's timeprint was in modeling the behavior of the user for the purpose of identifying click fraud. The results demonstrate that the timeprint is an effective solution for enhancing the quality of click fraud detection and that the proposed approach may be used as a pre-processing step. Berrar [40] conducted experiments using the FDMA2012 dataset and developed a model to detect suspicious publishers and clicks. In the RF approach, the average precision of the final ensemble model on the blinded test set was 36.20%. The experiments also revealed that features relating to click time might provide the most useful evidence concerning fraudulent activities. Guo et al. [41] claimed that most false clicks come from malicious data centers and that bots are designed for this fraudulent purpose in such centers, which are called CloudBot. A traffic-based near real-time approach was developed using ML techniques. As raw traffic was used at first instead of logs, log-based sampling and partitioning methods were inapplicable; therefore, a novel approach, IP-based sample partitioning, was proposed in which traffic flow data are sampled based on a user's IP address by hour. To maintain user privacy, the construction model did not include any information from the application layer; instead, most of the features were from the transport layer, such as the maximum segment size and time to live, in addition to statistical traffic features, such as packet variance in each flow. The study's findings demonstrated that the proposed system could spot click fraud while preserving privacy and that tree-based models achieve effective performance. Lia and Jia [42] evaluated the effectiveness of the RF algorithm in spotting fraudulent clicks using an ad click dataset, which needed to address imbalance; the purpose was to minimize information loss caused by undersampling or overfitting because of oversampling. As a result, the following approach was used: oversampling positive samples and undersampling negative samples. Following an analysis of the raw features in the dataset, more than 100 features were calculated, derived, and used as input to the RF, which was then compared with the SVM, NB, and DT, with an accuracy of 93% for predicting positive samples (i.e., legitimate clicks) and 91% for predicting negative samples (i.e., illegal clicks).

3.1.2. Gradient Boosting Models (GBM) Techniques

GBM and its variants, on the other hand, have been widely employed in previous studies due to their effectiveness in extracting both feature extraction and click classification. Gradient boosting models are an ensemble method that builds many decision trees sequentially. The extreme gradient boosting (XGBoost) variant is a more regularized version of Gradient Boosting.

Using generalized boosted regression models, Phua et al. [35] found that fraudulent clicks exhibit distinct spatio-temporal patterns. The authors emphasized that adopting simple statistical methods to extract features related to click behavior, click frequency, and high-risk click behavior would significantly increase model performance and reduce the overfitting of training data. Wang et al. [43] proposed a framework that combines two modules of supervised learning with rule-based statistical methods to deal with click fraud issues in both the user and traffic layers, as well as to identify as many click patterns as possible. Once the model is applied in both layers, users and their clicks are classified in the user layer, and fraud traffic is identified directly in the traffic layer. The outcomes show that when the gradient boosting decision tree (GBDT) classifier is trained with the derived hybrid features, such as time-related features (e.g., frequency of the cookie and IP recurrence feature), in a certain time window, there is an effective improvement in performance. Jianyu et al. [44] took the first step in detecting fraudulent activity in huge advertising platforms. Because nominal characteristics made up over half of the features in their dataset, they developed a unique coding method to convert nominal attributes to numeric ones while preserving the most useful information for fraud detection. They thoroughly examined features and determined the key features (i.e., if a feature with its frequency can identify more than 50% of the fraud activity in a dataset, it is a key feature). Furthermore, the key feature is combined with its frequency and stored in the dataset as a new nominal feature. Finally, to classify clicks in online advertising platforms, they constructed ML models, including LR, DT, GBDT, and XGBoost. In their study, Minastireanu and Mesnita [45] used a state-of-the-art ML algorithm, the light gradient boosting (LightGBM) decision tree-type approach, to explore the journeys of visitors who regularly click on ads but do not download apps (referenced in the ad link). As they engineered the features, they extracted time-related features from click time and calculated additional features, such as count (grouped by multiple features) and group-by-count unique values. They confirmed that the chosen algorithm outperforms XGBoost and stochastic gradient boosting (GB) in terms of computational speed and memory consumption; the outcomes of their experiments illustrated the effectiveness of the proposed method, achieving an accuracy of 98%. The vast number of raw click data, complexity, high cardinality, and unbalanced class distribution, according to Singh and Sisodia [46], are some of the obstacles that make click fraud in the advertising industry crucial. This issue requires the use of a cautious algorithm. Therefore, they used the gradient tree boosting (GTB) model, which has been evaluated on multiple datasets and compared against 11 other ML algorithms; it demonstrated usefulness in detecting fraudulent publishers and overcoming the stated challenges. To investigate the behavior of users who repeatedly click on an ad without performing the required action (e.g., download an app or make a specific purchase), Dash and Pal [47] conducted a study to develop a robust, adaptive, and scalable feature set for click fraud detection. They trained models, including SVM, KNN, DT, RF, and GBDT, with input features, such as URL, ad server IP address, and referrer URL. Nevertheless, there were some limitations in this study because the authors anticipated more detailed features, such as those explaining information about the user (e.g., geographical region), as well as features describing click behavior, such as mouse movement patterns; in addition, detailed click time features, such as day, hour, and minute, cannot be extracted. The GBDT method achieved an accuracy of 97.20%. Viruthika et al. [48] constructed a model based on the XGBoost method, which is commonly used in both the feature extraction and feature selection (FS) stages. The method demonstrated effective performance compared with other classification algorithms after the completion of multiple massive trials and testing. This study included the main

features of click time, IP, app, OS, and device, which may be gathered from each click to train the model. The method achieved 91% accuracy, effectively handled missing data, and prevented overfitting. This is because it is preferable for the click fraud detection system to be optimized at a low cost while maintaining good precision and recall scores. Srivastava [49] developed an approach based on ML and heuristics. First, to select the features of interest to include in the model, the relationship of conversions (i.e., the action computed when a visitor interacts with an ad) with each feature (e.g., IP, app, and device) was analyzed. For example, the authors demonstrated that the relationship between click and rate conversions for benign clicks is that the latter drops significantly as the former increases because a legitimate user is likely to click on the ad several times but will only perform specific actions (e.g., download an app) once. The study's key objective was to exploit a minimal feature set with high accuracy and efficiency by applying the following models: linear support vector classifier (SVC), GB model, KNN, MLP, and NB. Furthermore, the categorical features were converted into one hot encoding to enhance effectiveness. Thejas et al. [50] developed the cascaded forest and XGBoost classifier (CFXGB) model, which is based on a combination of the cascaded forest to handle the original dataset with additional features and then XGBoost for classification. Their results showed that it is a feasible method for detecting click fraud. Experiments were carried out on three widely known public datasets: TalkingData, Avazu, and Kad. In all these datasets, click time was split into month, day, and hour, with certain features that had nothing to do with identifying the click excluded. Finally, meta-analytical comparisons with previous studies that used ensemble models, such as RF and GBDT, revealed that the proposed method is more effective in terms of performance than the other models. Gohil and Meniya [51] explored the capability of the XGBoost classifier to recognize malicious click cases as a feasible solution. After execution, the model was verified to be highly successful without diminishing the final accuracy; in addition to the existing features of the datasets, the time-to-click feature was split into month, day, and hour, and the IP feature was paired with all other features. When classifying clicks in the first dataset, XGBoost obtained a high accuracy of 96%. A further study by Singh and Sisodia [52] found that transferring from appropriate learning domains to new ones can improve model performance while saving training time. Their approach focused on the integration of 1D user click time-series non-image features into a 2D graphical image. Next, classification was performed using five main ML models: SVM, RF, DT, AdaBoost, and GBT. The latter demonstrated effective performance, with 79.8% precision.

3.1.3. Other ML Techniques

Moreover, there are other ML models that can be exploited in order to effectively perform the click classification.

Gabryel [53] deployed JavaScript on an advertiser's site for a month to gather all accessible click data and determine the origin of clicks (humans or bots). He developed an analytical approach that uses the term frequency-inverse document frequency (TF-IDF) algorithm, weights all features, and then leverages the KNN model to distinguish benign or fraudulent clicks. Compared to using only the TF-IDF technique, assigning weights to each feature enhanced the classification results. In a two-stage model, Almahmoud et al. [54] used two types of fingerprints: one rule-based fingerprint and one ML-based fingerprint. When a user requests a web page, in the first stage, the fingerprint is retrieved from deterministic information about the user and web page navigation, whereas the other fingerprint is generated in the second stage by using the features of the click behavioral patterns during and after clicking on the ad and then classifying the click as legitimate or benign. Several classifiers have been examined, but the KNN classifier has proven to be the most accurate, with a precision of 97.6%. Pan et al. [55] exploited top-rank-k frequent pattern mining in combination with an SVM classifier to improve the effectiveness and coverage of click fraud detection. The proposed method depends on click abstraction, click frequency description, and other aspects regarding the ad click. A simulation was carried

out in a practical setting to assess the efficiency of the method, and the results showed that it is useful and efficient, as it considerably decreases time complexity. To effectively handle imbalanced datasets, which are common in click fraud detection data, causing the classifier to be used to be either overfitted or underfitted, scholars have developed several resampling techniques to address this issue. According to Thejas et al. [56] developed an algorithm with the goal of modeling temporal click fraud data in terms of probabilities, which takes into consideration time scales of minutes, hours, and seconds. This algorithm can predict click fraud behavior in these scales. The auto-regression and moving average models were constructed, and they used only time-series features and labels/classes (fraud/benign) without requiring a huge number of features to be involved; subsequently, the LR classifier was applied to differentiate between fraud and legitimate clicks. Borgi et al. [57] tested the extent to which extraordinary timeprints can be indicators of click fraud detection. The proposed rule-based method used offline rules, such as the blacklist rule, which recognizes fraud by considering the time spent by the human/bot to click on the ad, and then goes through the classification using the LR classifier. This model considers an important aspect that the authors believe is the most crucial predictor of whether a click is fraudulent: measuring the amount of time a person spent surfing the Internet prior to clicking on the ad. The study concluded that timeprints might be significant indicators of false clicks. Li et al. [58] established the first model to detect click fraud using multimodal and contrastive learning approaches, taking into consideration the variations in demographic information, behavior patterns, devices, and media used by actual users and fraudsters while clicking on ads. The MCCF model was developed based on the use of three modules: wide and deep features, behavior sequences, and heterogeneous networks. It required integrating multimodal information into each layer of the MLP and then using contrastive learning during the model training phase to overcome the imbalance problem. Several attempts have been made to test the proposed model in varied forms, such as model validation after eliminating wide and deep features and after excluding the behavior sequences. The three-component MCCF demonstrated its efficiency by achieving the greatest performance and surpassing all preceding models by a significant margin at 98.7% precision. Aberathne and Walgampaya [59] devised a framework known as the real-time mobile ad investigator. The system is knowledgeable and capable of detecting fraudulent clicks using an all-new supervised ML model known as the hidden Markov scoring model (HMSM), which was developed by combining the hidden Markov model, a rule engine, and pattern recognition algorithms. The entropy-based binning approach was also used to extract and transform new categorical features. The HMSM classifier was beneficial as a supervisor learning method; with an accuracy of 94%, it minimized complexity during training while performing well while classifying blind data. The findings showed that the model is effective in detecting fraud not only in clicks but also in impressions and user sessions in the field of advertising. Mikkili and Sodagudi [60] used a dataset along with its raw features to calculate the number of clicks on the same ad per day, as well as the frequency of each click because, as the authors noted, these are vital indicators to determine where legitimate and illegitimate clicks originate. The proposed method for detecting invalid clicks, which result in significant illegal gains, depends on the usage of two classifiers, LR and Gaussian naive Bayes (GNB), which obtained exceptional accuracy of 99.23% and 99.78%, respectively. Dekou et al. [61] highlighted a recurring issue with detecting fraud in e-commerce—datasets are commonly very imbalanced, and fraudulent humans or bots constantly adapt their behavior to remain undetected. The model used depends on the powerful open-source libraries H2O and Catboost (a GB technique for DT libraries), as well as AutoML, which attempts to collect numerous basic models in order to enhance prediction quality and produce a more efficient aggregation model. It was notable that the ensemble models outperformed the individual models, with the stacked ensemble model achieving the greatest performance with an area under the curve metric of 98.85%. Singh and Sisodia [62] developed a method called the quad division prototype selection-based K-nearest neighbor classifier (QDPSKNN) to handle the dataset

imbalance issue, with the objective of minimizing capacity requirements and enhancing execution performance. This process is based on dividing the data into four quarters and then performing undersampling. The performance and generalizability of the proposed method were evaluated using the FDMA2012 dataset of ad clicks, in which 103 features derived from previous work [35] and numerous trials on 15 other common imbalanced datasets were used, with the sizes and quantities of the features in the datasets varying. The approach was compared with the traditional KNN model. The results revealed that QDPSKNN enhanced classification and was successful in addressing data imbalance, as it gained the best accuracy in 13 out of 16 datasets and obtained a precision rate of 75.1% in the click-related dataset.

3.2. DL-Based Click Fraud Detection Techniques

Additionally, DL methods have been investigated in some studies as a solution for detecting click fraud. Either alone, or in combination with other ML techniques.

Mouawi et al. [63] developed the CFC click fraud detection model, which uses custom-designed features along with ML models, including KNN, artificial neural network (ANN), and SVM. The model distinguishes itself by allowing a third party to supervise and control the entire click fraud detection process through crowdsourcing ad requests, compared to usual models that place this task in the hands of an ad network or advertiser. The classifiers were evaluated with different kernels and values; the findings showed that KNN with $K = 5$ (number of neighbors) produces impressive outcomes, with an accuracy of 98.26%. The work of Renström and Holmsten [64] was one of the few that used unsupervised ML techniques in which they used autoencoders (AEs) because of a shortage of labeled data. The procedure was carried out in three steps. First, a simple three-layered AE was used; second, a stacked AE was proposed, which was simply a three-layered AE, but each layer was trained separately before each output served as an input to the subsequent classification model. Finally, a variational autoencoder (VAE) model was used, which operates similarly to the second model, but statistical approaches, such as mean, standard deviation, and latent vector, are applied. A click fraud dataset was one of the datasets used to evaluate the proposed models. The results revealed that when time- and date-related features are eliminated from the models, their performance suffers considerably, with the stacked model outperforming the others. In another investigation, Zhang et al. [65] developed a method called CSBPNN-ABC to detect fake clicks, and the findings were promising. They primarily applied the back propagation neural network (NN) technique, which they combined with the novel artificial bee colony (ABC). They used the ABC algorithm to find the optimal feature set, and then the CSBPNN-ABC approach was used to classify ad clicks across mobile platforms. Throughout the experiments, the authors found that the basic features included in the dataset are insufficient for the direct creation of the click fraud detection model. As a result, they relied on Oentaryo's [34] earlier work, in which he retrieved more than 100 features using statistical methods. Only 13 characteristics were used in the investigation to enhance performance and minimize training complexity. Compared to cutting-edge techniques on real-world click datasets, the proposed model performed effectively. According to Thejas et al. [66], there is a shortage of studies in the existing literature that determine and develop various techniques for detecting and preventing click fraud, notably because a bot can be an intelligent attacker that constantly learns from the classification training method in order to mislead the fraud detection model training process. As a result, the authors proposed a model that is a combination of an ANN and an AE. Furthermore, the dataset used is highly unbalanced, so a semi-supervised generative adversarial network was included as a solution to improve the accuracy of the ANN; the ANN contained several layers in which each layer learns the pattern and distribution of the click for a specific extent and then passes the understanding to the next layer and so on. The results showed that the method is effective for detecting fraud. However, because of a lack of illegitimate clicks, the AE was unable to recognize bot clicks that mimic the pattern or distribution of human clicks, and these illegitimate clicks were misclassified as benign.

In 2020, Shi et al. [67] created ClickGuard, a mobile solution that leverages mobile sensors, such as an accelerometer and gyroscope, to enter signal-related data because the pattern of vibration signals differs greatly between legitimate click actions and fraud actions. Mel frequency cepstral coefficients were chosen to extract features and build strong features (e.g., formants of the signal spectrum) in order to distinguish fraudulent click signals, and then classify the data using classifiers, such as RF, logistic model tree, and convolutional neural networks (CNNs). The latter performed best, followed by the RF classifier. Hu et al. [68] introduced a hybrid model that combines a novel weighted heterogeneous graph and deep neural network (DNN). In the first phase, a weighted heterogeneous graph was established to display the behavioral patterns, apps, devices, and other characteristics of an advertisement's clickers. Then, the time was separated into numerous windows, and statistical methods were used to derive new features based on each window, with the log data for each app split into 24 parts every day (i.e., the time window was 1 h). Compared with other models, such as SVM, RF, and fully connected neural network (FCNN), the developed model performed excellently. Liu et al. [69] introduced a method to detect the distribution of fraudulent clicks among human clicks, in which they first used a priori probability. As the raw features in the dataset do not provide an overview of users' behavior when they click on an ad (as the authors stated), they relied on earlier features derived by Oentaryo [34]; they then calculated the entropy value to choose the new included features, transformed it into a feature matrix over many time windows, and then divided it into two primary windows with different temporal patterns. They also generated statistically new features from the given characteristics that were appropriate when used to train a cost-sensitive CNN method. To demonstrate the effectiveness of the proposed method's performance, they compared it with multiple ML models, such as J48, RF, and SVM; their algorithm outperformed the aforementioned models with a precision of 89%. Gabryel et al. [70] used FCNNs in an experiment to classify clicks as legitimate or illegitimate (click monitoring) and to examine visitors' behavior after clicking, such as whether they would fill out a form or purchase a specific service (lead monitoring). Commonly available data from the user's browser were collected, such as browser type, OS, number of page views, and mouse movement, to train the model. Three models were tested: an MLP network, a multilayer network model, and a particular neural network (NN) with one hidden layer in which the weights of the underlying layers were trained first. The last model had the most efficient performance, with 99.5% accuracy. Zhu et al. [71] focused on a type of click fraud known as a humanoid attack, in which human click behavior and click patterns are fully mimicked to dodge detection by earlier works. The authors developed the ClickScanner tool, which uses VAEs to detect the stated type of attack, along with establishing a data dependency graph using static analysis to extract the major features from mobile apps and generate a feature vector. Several features, such as ad view size, which the attacker can leverage to specify locations of the falsely clicked ad within the ad view, were considered to investigate malicious code pieces that an attacker can infuse into apps in order to produce false clicks with no user intervention. Compared with earlier tools, the proposed tool was capable of detecting 170 abnormal click behaviors from 166 suspect apps in less time. Chari et al. [72] conducted a study to evaluate many ML methods, including LR and recurrent NNs, and several boosting techniques, such as LightGBM, AdaBoost, and GB. In terms of feature engineering, new features were added, such as the frequency feature, which relates to the IP feature by calculating each unique IP in the dataset and its frequency, as well as adding the click ID, which is the identifier that reserves a certain number for each click. The time feature was divided into hour and day, and the IP feature was paired with all the other features in a combination of one/two attributes. With an accuracy of 98.69%, LR performs admirably; the findings demonstrated that using the LR algorithm to detect click fraud is quite successful and will attract the interest of advertisers and publishers.

3.3. Models Used to Classify Ad-Related Web Pages

There are studies that have also explored the use of AI models to classify pages with ads or not for fraud detection purposes. For example, Crussell et al. [73] conducted the first click fraud study on Android app ads, examining two basic behaviors: displaying ads while an app is running in the background and clicking on ads without any real interactions from the user. Moreover, they introduced a tool called MAdFraud that can work with multiple applications at the same time and directly recognize clicks and impressions through three stages: building Hypertext Transfer Protocol (HTTP) request trees, defining ad request pages using ML, and detecting clicks using heuristics in HTTP request trees. Several useful features for training the RF model were extracted in the first stage, such as request length, publisher ID, and timestamp. The results showed that there were 21 click fraud apps detected, with 35 clicks occurring in the background and 24 clicks occurring in the foreground. Also, in a study conducted by Iqbal et al. [74], a method called Fight Click Fraud (FCFraud) was proposed to provide a solution to click fraud and to safeguard users' devices from unintentionally becoming part of a fraudulent click network after being infected by malware. On the user's side, this solution could be integrated into the anti-malware software of the operating system (OS). In general, FCFraud recognizes background activities, scans and analyzes HTTP packets, and uses ML classifiers, such as NB, SVM, K-nearest neighbors (KNN), C4.5, and RF, to automatically identify ad HTTP requests. Following these, it applies a variety of heuristics to avoid false ad clicks.

3.4. Click Farm

Moreover, some studies have focused on a type of click fraud known as click farming—in which a large number of cheap laborers is hired to click on paid ad links.

Jiang et al. [75] used PU learning, a positive-unlabeled learning method, to gather reliable negative examples from an unlabeled set. They conducted weighted logistic regression to examine the contribution of extracted features in analyzing click farming on one of the largest Chinese shopping sites and to investigate whether click farming occurred in stores hosted on this platform. Among many classes, including SVM, RF, and NN, and then combining them through an ensembling method, the latter achieved the best accuracy at 97.4%.

3.5. Develop Feature Selection (FS) Methods

Furthermore, some authors have developed feature selection (FS) methods that have been validated primarily on datasets related to click fraud.

Taneja et al. [76] presented a novel method for selecting features using the recursive feature elimination method, classifying legitimate or illegitimate clicks related to mobile web browsing and dealing with data imbalance using the Hellinger distance decision tree (HDDT) classifier. Statistical methods were also used to engineer the features, such as click time, from which 12 features were extracted using statistical measures, such as maximum, average, and skewness. Experiments were carried out to compare the model with robust ML models, including J48, LogiBoost, REP tree, and RF; HDDT performed well, with an accuracy of 64.07%. In their further study of FS approaches, Thejas et al. [77] revealed that FS is beneficial for enhancing algorithm performance and lowering computing overhead; however, it does not demonstrate a meaningful benefit in some classifiers, such as RF classifiers. As a result, two FS techniques were designed: metric-ranked and accuracy-ranked feature inclusion. These two strategies are fundamentally hybrids of the wrapper and filter methods, with each feature assigned a value and then ranked, followed by the selection of features. The performance of the two techniques was validated using 12 datasets, three of which were related to click fraud. The proposed techniques do not require knowing the number of features in advance; they add features incrementally and do not require a specific, minimum, or maximum set of features. And lastly, Thejas et al. [78] designed the Kalman-SMOTE (KSMOTE) algorithm by combining the strong performance of the synthetic minority oversampling technique (SMOTE), an oversampling technique in which synthetic samples are produced for the minority class, along with a Kalman filter.

The approach removes noisy samples in variable proportions based on the user-specified number of iterations, thus maintaining class balance. The RF model was then applied to evaluate the proposed algorithm’s performance on three imbalanced ad click datasets. The KSMOTE outperformed earlier algorithms in most datasets, demonstrating a significant gain in performance.

As a summary of all previous reviewed articles, Table A1 in Appendix A displays the dataset, used features, strengths, limitations, outcomes, and evaluation metrics. Note: Studies are listed in chronological order (2012–2022).

4. Common Datasets in the Field of Click Fraud Detection Using AI Techniques

In this section, we go over the most relevant datasets mentioned in the literature review to detect and prevent ad click fraud using AI techniques. Private/non-open-source datasets, as well as those related to other fraud types, such as ad or impression fraud, are excluded. We discuss the descriptions, raw features, and some of the issues with the datasets, as identified in the literature review, as well as how these issues are addressed.

4.1. FDMA 2012 BuzzCity Dataset

4.1.1. Dataset Description

The FDMA BuzzCity dataset [79] was introduced in 2012. This dataset has been used in several experiments [26,28–32,35,37,39,43,59,62,66,73] and is divided into two portions—publishers and clicks—The publisher dataset contains data relevant to the publisher profile, whereas the click dataset provides click records associated with each publisher. All field descriptions for both datasets are shown in Section 4.1.2. This dataset was originally offered as part of a competition with the goal of developing an effective model/technology to detect fake publishers and understand publishers’ lack of credibility patterns based on publishers’ profiles and clickers’ click behavior. Each publisher has three statuses: OK indicates that the publisher is benign, fraud denotes that the publisher is illegitimate (intentionally generates high-cost clicks with no real interest in the ads by using automated software or click farms), and observation indicates that the publisher’s status has not been verified because the publisher is either new or has too many clicks but is not yet declared fraudulent. There are three sets of data available in FDMA 2012 BuzzCity: a training set (to develop predictive models), a validation set (to select models), and a test set (to test model generalizability). During a period of three days, the click dataset captures data related to clicks, while each publisher dataset records publishers who received at least one click. Table 1 presents the statistics of the dataset.

Table 1. Statistics of the FDMA 2012 BuzzCity dataset.

Dataset	Time Period	No. of Clicks	No. of Publishers			
			Fraud	Observation	OK	Total
Train	9–11 Feb 2012	3,173,834	72 (2.34%)	80 (2.60%)	2929 (95.07%)	3081
Validation	23–25 Feb 2012	2,689,005	85 (2.77%)	84 (2.74%)	2895 (94.48%)	3064
Test	8–10 Mar 2012	2,598,815	82 (2.73%)	71 (2.37%)	2847 (94.90%)	3000

4.1.2. Raw Features

As stated previously, the FDMA 2012 BuzzCity dataset has two sub-datasets. Because the data are obtained through mobile devices, certain features of similar data on desktop computer networks are not available. The publisher dataset contains information about the publisher, such as ID, address, bank account, and status, as listed in Table 2. The click dataset, on the other hand, contains information on the source of the click, such as ID and IP, as well as other attributes that describe the click behavior, such as device and click time, as shown in Table 3. For the sake of privacy protection, most of these data have been anonymized. The studies in the literature review did not use these features in their raw form to train models for detecting click fraud but instead went through a series of

processes with the purpose of engineering features using statistical approaches, a collection of two/three features, or other methods. For example, consider studies [28,30], which calculated the average, variance, maximum, and entropy of the click time feature at various time intervals, as well as other raw features. Another study [39] separated the click time feature into several time features, such as day, month, and period of day. Section 3 will provide further details on the new features derived using feature engineering.

Table 2. Features in the publisher dataset.

Attribute	Description
publisherid	Unique identifier of a publisher.
bankaccount	Bank account associated with a publisher (anonymized; may be missing/unknown).
address	Mailing address of a publisher (anonymized; may be missing/unknown)
status	Label of a publisher, which falls into three categories: <ul style="list-style-type: none"> • OK: Publishers whom BuzzCity deems as having healthy traffic (or those who slipped their detection mechanisms). • Observation: Publishers who may have just started their traffic or their traf- fic statistics deviates from system wide average. BuzzCity does not have any conclusive stand with these publishers yet. • Fraud: Publishers who are deemed as fraudulent with clear proof. BuzzCity suspends their accounts and their earnings will not be paid.

Table 3. Features in the click dataset.

Attribute	Description
id	Unique identifier of a particular click.
numericip	Public IP address of a clicker/visitor.
deviceua	Phone model/agent used by a clicker/visitor.
publisherid	Unique identifier of a publisher.
campaignid	Unique identifier of a given advertisement campaign.
usercountry	Country from which the clicker/visitor is.
clicktime	Timestamp of a given click (in yyyy-mm-dd format).
referredurl	URL where ad banners are clicked (anonymized; may be missing/unknown). Publisher’s channel type, which consists of: <ul style="list-style-type: none"> • ad: Adult sites. • co: Community. • es: Entertainment and lifestyle. • gd: Glamour and dating. • in: Information • mc: Mobile content. • pp: Premium portal. • se: Search, portal, services.
channel	

4.2. TalkingData AdTracking Dataset

4.2.1. Dataset Description

The TalkingData AdTracking dataset was put up on Kaggle in 2017 as a competition by the Chinese company TalkingData, which processes three billion clicks every day, 90% of which are possibly illegitimate. [80]. Its method of detecting and preventing click fraud is to monitor and analyze users’ click journeys across their portfolios; IP addresses that generate many clicks but never install apps are flagged. A blacklist of IP addresses and devices is then created based on this information. The goal of the competition was to develop the best model for predicting whether a user would proceed to install an app after clicking on an ad and to distinguish fraudulent clicks from benign ones. The data provided were extensive, comprising a total of 203,694,359 real-time ad click records captured on a mobile platform, with an overall size of roughly 7 GB over four days. Table 4 illustrates the statistics of the TalkingData dataset.

Table 4. Statistics of the TalkingData dataset.

Dataset	Time Period	No. of Clicks	No. of Publishers	
			Fraudulent	Non-Fraudulent
Train	6 Nov 2017	184,903,890	509,235.8975	203,185,123.103
Test	9 Nov 2017	18,790,469	(0.25%)	(99.75%)

4.2.2. Raw Features

The TalkingData dataset is generated from mobile phones and is widely used by studies in recent years to build and train click fraud detection models and approaches. It has millions of clicks distributed among eight features. These features are described in detail in Table 5; seven of these features are considered independent features, whereas one is deemed dependent (Is_attributed feature. It is the class that will be predicted; a value of 0 indicates a legitimate/non-fraudulent click, whereas a value of 1 indicates a fraudulent click). Some studies used raw features as they are, whereas others added primitive feature engineering. For instance, several studies have focused on extracting temporal features, such as minutes and seconds, from the click time feature [42,47,48,53,58,75] and on grouping IP features along other attributes in a combination of one/two attributes [42,48,69]. Section 3 provides the details of the new features derived using feature engineering.

Table 5. Features in the TalkingData dataset.

Attribute	Description
ip	Ip address of click.
app	App id for marketing.
device	Device type id of user mobile phone.
os	OS version id of user mobile phone.
channel	Channel id of mobile ad publisher.
click_time	Ad timestamp of click (UTC).
attributed_time	If user download the app after clicking an ad, this is the time of the app download.
is_attributed	the target that is to be predicted, indicating the app was downloaded.

4.3. Avazu Click-Through Rate Prediction Dataset

4.3.1. Dataset Description

The Avazu dataset was also presented on the Kaggle platform in 2014 [81]. As part of a competition hosted jointly by Avazu and Kaggle to determine the best strategy for predicting the CTR, which is a vital measure for analyzing ad performance. The dataset includes around 40 million records captured over 11 days, with 10 days serving as the training set and 1 day serving as the test set. Click prediction systems are critical, and they commonly use sponsored searches to rank ad links. The goal of CTR prediction is to estimate the likelihood that advertisements on a website will be clicked. By predicting the CTR, an advertising agency selects the potential visitors who are most likely to engage with the advertisements. Table 6 illustrates the statistics of the dataset.

Table 6. Statistics of the Avazu click-through rate prediction dataset.

Dataset	No. of Clicks	Class of Clicks	
		Non-Clicked	Clicked
Train	40,000,000	40,140,000 (0.25%)	4,460,000 (10%)
Test	4,600,000		

4.3.2. Raw Features

The dataset includes 21 distinct features, some of which describe the ad, such as the ad ID and position, as well as features describing the source of the click, such as device

type and connection type. The target feature is the click class, which is binary: 0 means that an ad was not clicked, and 1 indicates that the visitor clicked an ad. About eight out of the 21 features in the dataset are anonymous, as illustrated in Table 7. These are categorical fields that contain specific data about users’ and advertisers’ profiles and are hashed to a unique value to enable investigators to construct vectors [82]. These anonymous fields and their meanings were not publicly revealed or investigated in the studies we examined in the literature review; however, there were some efforts in other experiments in which it was inferred that C14 is the ad ID, C17 is the ad group ID, and C21 is the ad sponsor ID on the basis of interpreting the hierarchy of unknown features [83]. All the studies that applied this dataset in the literature review used all the information provided in the dataset to detect click fraud, along with separating the click time feature into month, day, and hour and determining the frequency of clicks in 10 h [47,74,75].

Table 7. Features in the Avazu click-through rate prediction dataset.

Attribute	Definition
ID	The unique identifier for all details that corresponds to one occurrence of an advertisement. This is a continuous variable.
Hour	The hour, in YYMMDDHH format. We could break this down and add additional features during the cleaning process. This is a continuous variable.
Banner_pos	The position in the screen where the advertisement was displayed. This shows the prominent place for an advertisement to get the attention of the user. This is a categorical integer
Site_id	The identifier to unique identify a site in which the advertisement was displayed. This is a hashed value.
Site_domain	The domain information of the website in which the advertisement was displayed.
Site_category	This is a categorical variable representing the field to which the website belongs to. This can be used 27 to understand if any site category has more visitor attraction during any particular time.
App_id	The identifier to unique identify a mobile application in which the advertisement was displayed. This is a hashed value.
App_domain	The domain information of the application in which the advertisement was displayed.
App_category	This is a categorical variable representing the field to which the application belongs to. This can be used to understand if any app category has more visitor attraction during any particular time. This is similar to the site category and can be compared relatively to check if app has more clicks over the website.
Device_id	The unique identifier that marks the device from which the click was captured. This is a hashed continuous variable and can be repeated in the data set.
Device_ip	The ipv4 address of the device from which the click was received. Hashed to a different value for privacy reasons to avoid trace back to the device.
Device_model	The model of the device. We choose not to use this value.
Device_type	The type of the device, is a categorical variable and has around 7 categories.
Device_conn_type	This is a hashed value about the connection type. We do not use this value for forming the vector.
C1	An anonymous variable. It has influence over the prediction.
C14–C21	Anonymous categorical variables that might have information about the advertisers’ profile and the users’ profile like the age, gender, etc.
Click	The target variable, 0 means an advertisement was not clicked and 1 means the ad was clicked. This is a categorical variable, binary typed.

4.4. Challenges in the Common Datasets in the Field of Click Fraud Detection

Most datasets related to click fraud detection suffer from imbalance, with the majority of the negative class (i.e., fraudulent clicks) being few and the benign class (i.e., legitimate clicks) compensating for most data records, causing the prediction model to be skewed toward the majority. Previous studies used various resampling procedures, such as up/down sampling and the SMOTE. In up-sampling, minority samples are reproduced until they are equal to those from the majority class. Under-sampling, on the other hand, excludes from the majority class at random until the class distribution is balanced. This issue was addressed in the FDMA 2012 BuzzCity datasets by implementing several workarounds. Under-sampling, for example, was used to overcome the problem of dataset imbalance in study [33], in which certain majority-class records were chosen for use along with all minority-class records. In another study [42] to address the same issue, the process was

carefully aimed at minimizing information loss caused by undersampling or overfitting because of oversampling. As a result, the following approach was used: oversampling positive samples and undersampling negative samples. Furthermore, the QDPSKNN method [62] was applied to address dataset imbalance, along with reducing capacity needs and improving implementation performance. It divides the data into four quadrants and then undersamples to balance class distribution. The authors in [34] also illustrated that to deal with imbalanced class distribution, in nonlinear classifications involving combinations of variable types and noisy or missing patterns, tree-based ensemble classifiers and backward feature exclusion can produce promising results.

On the other hand, the TalkingData dataset also suffers from a massive imbalance, with legitimate clicks representing only 0.25% ($Is_attributed$ feature = 0) of all clicks in the entire dataset, while fraudulent clicks ($Is_attributed$ feature = 1) represent 99.75%. The problem with the model being trained on an imbalanced dataset is that it will be skewed exclusively toward the majority class. This presents an issue when we are concerned with the prediction of the minority class. Several efforts have been made in earlier studies to overcome this challenge when training models. For example, a 15% random selection of unique IPs was used, followed by an 8% stratified sample from the rest to decrease the data size in [61]. To address the imbalance, the SMOTE [84] with neighbors = 5 was used, and the positive class oversampled by 11%. In [59], however, undersampling was used to balance skewed datasets by preserving all data in the minority class and decreasing the volume of the majority class. The authors suggested that the test set should be more comprehensive, so the original training set was divided into two new training and test sets, and then additional samples were drawn from the new test set by selecting the $is_attributed$ feature = zero and one click in a 1:1 ratio, resulting in a dataset that contains 50% legitimate and 50% fraudulent clicks. Furthermore, they trained all the different ML models using variable samples of training data; the goal of changing the training sample sizes was to enhance precision while lowering overfitting. In another effective way to address this issue [66], the entire dataset was divided into six classes based on the IP and app ID features as follows: class 1: 25,974 rows with a unique IP count < 20 and an app ID frequency < 70%; class 2: 27,174 rows with a unique IP count in the range \Rightarrow 20 and <1000 and app ID frequency < 70%; class 3: 112,790 rows with a unique IP count \Rightarrow 1000 and an app ID frequency < 70%; class 4: 784,964 rows with a unique IP count < 20 and an app ID frequency \Rightarrow 70%; class 5: 19,914,810 rows with a unique IP count in the range \Rightarrow 20 and <1000 times and an app ID frequency \Rightarrow 70%; and class 6: 164,038,178 rows with a unique IP count \Rightarrow 1000 and an app ID frequency \Rightarrow 70%. The classifiers were tested separately on each of these classes. In another study [56], the model was trained using only the first class.

And according to the statistics in Table 7, there is a significant disparity in the Avazu click-through rate prediction dataset, with the percentage of those who viewed the ad but did not click on it being 90%, and the percentage of those who viewed the ad but clicked on it being 10%. All the studies in our literature review used one million samples from the dataset [74,75], and applied undersampling technique to balance the dataset [50].

5. Common Features in Click Fraud Detection Using AI Techniques

In this section, we review the most frequently used features in AI-based click fraud detection techniques. We explain which features were considered in previous studies and their meanings, as well as which features were deemed essential for detecting ad click fraud.

A click request is transmitted to the Ad Manager ad server whenever a user clicks on any ad. When the ad server receives the request, it counts the clicks and directs the user to the landing page. As we all know, these clicks can come from real people who are interested in advertising or from a bot. Furthermore, advertising campaigns are managed either by the provider (e.g., Google Ads) or by a third party (e.g., marketing and advertising agencies), and one of the tasks in managing advertising campaigns is to prevent and block bot clicks as much as possible using various effective techniques, such as filtering or restricting click activity. For instance, a bot is identified as an IP/host that has made clicks greater than

the THRESHOLD TOTAL clicks during that same day or greater than the THRESHOLD HOURLY clicks at any hour of a given day. The thresholds THRESHOLD TOTAL and THRESHOLD HOURLY can be changed [85]. To effectively manage advertising campaigns, the provider or third party stores a variety of click-related features, which can be used to monitor and respond to advertising performance, calculate advertiser costs, prevent unauthorized clicks, or make other decisions. Some features can be captured from the browser and JavaScript. This is because JavaScript is implemented by at least 98% of web browsers [86], and most online advertising platforms rely on JavaScript functionality, such as recording mouse movements, keystroke events, or IP addresses, in addition to several temporal-spatial features [70].

5.1. Descriptions of the Most Commonly Used Features in Ad Click Fraud Detection

As illustrated in Table A1, prior ad click fraud detection efforts have mainly focused on accurate and careful data pre-processing and feature generation, as these are crucial phases that considerably enhance the overall accuracy of click fraud detection models. Malware, bots, and competitors frequently simulate their malicious activities by using a variety of methods, such as producing very sparse click sequences, changing IP addresses, and generating clicks from multiple machines in various locations. As a result, studies have been carried out to investigate click sources and user behavior during and after clicking on an ad. Most of these works have examined temporal characteristics; for example, in [29], an essential factor, click profile, or the time delay between consecutive clicks, was explored. Two types of click profiles were recorded—long and short. Both types calculated the number of clicks generated from the same IP address per day, which was measured at less than 5 s, between 5 s and 10 s, between 10 s and 20 s, between 20 s and 30 s, and so on over an interval >300 s. The difference of the two types is that the long profile should be at least 10 clicks from the same IP, whereas the short profile should not be fewer than five clicks from the same IP.

Furthermore, click timestamp was split into smaller segments, such as month, day, hour, minute, and second [31,36,42,45–47,56,58,69,74,75], to investigate click behavior in the smallest possible detail. Bot clicks frequently have patterns after an ad is clicked because their behavior when interacting with the landing page is minimal or almost non-existent, as demonstrated in [36]; for a click to be considered legitimate, one of the following criteria must be met: 30 s of dwell time, 15 mouse events, and 1 click; or 30 s of dwell time, 10 mouse events, 1 scroll event, and 1 click; and so on. Other experiments, such as [27,28,31,32,35,38,40,43,46,60,62,66,73], have attempted to construct more precise features with regard to click time and other features, such as the total number of clicks or devices used, by identifying each attribute individually and then modeling the click pattern according to that attribute through the generation of several parameters depending on the attribute. These parameters are determined using statistical measurements, such as maximum, mean, skew, and variance; for example, some features are Var5mins, Var10mins, Var1hr, Var2hrs, Var6hrs, or AvgClicksPer5Min, MaxClicksPer5Min, Varper5minclick, and MaxClicksperMin.

According to recent data [87], throughout the weekdays of an advertisement campaign in 2022, there was a major increase in bots' activities, with bots peaking around noon (Pacific Daylight Time) and uniformly growing from 0 to 800 per hour before returning back to 0 per hour after the noon peak during the week. On weekends, bot numbers decrease practically to none. The period of day may indicate a higher likelihood of illegitimate clicks. In another set of recent statistics [88], the most active period of day for bots was 1–2 pm worldwide in Eastern Standard Time, and the least active period was 9–10 pm. As a result, several studies [27,32,36,40,43,56,62] have focused on these two critical aspects when detecting click fraud. They split a day into numerous segments, such as morning, afternoon, and midnight. They also indicate whether it is weekday or weekend. There are few published studies that have considered user behavior [33,50,56,67]; for example, in some experiments, features such as the number of scrolling events, the number of mouse movements, keystroke data, and the types of actions that a visitor made after clicking are

used to detect click fraud. Some studies have relied on IP reputation as well, reasoning that any malicious action from an IP address indicates that all activities from that IP address are likely to be illegitimate. As a result, IP blacklists were constructed, and bots running from blacklisted IP addresses were prevented. In other research, features were introduced as groups of two or three rather than individually [48,69].

Table A2 in the Appendix B shows a list of all the features used in previous studies (“used features” in Table A1), which are classified as follows: temporal, spatial, user behavior, used medium, IP address, number of clicks, provider or advertiser, and other features.

To ensure that there was no dispersion, we standardized the names of features across all studies that were included in our literature review. In spite of this, we ensured that the names were precisely standardized, and that the features’ names were re-emphasized to reflect their meanings and accurate descriptions.

5.2. Discussion of the Features Used

In this sub-section, we explore and illustrate the most essential features that were considered significant when training models and building systems that can detect click fraud at a high level of certainty. As seen in Tables 1 and 7, most studies did not use features in their raw form but rather processed and engineered them in various ways to improve the accuracy of malicious click detection while avoiding the overfitting or underfitting of models. As a result, a properly engineered and selected feature set should be able to capture characteristics or trends relevant to fraudulent clicks and be robust to the changing patterns of behavior used by malicious parties. These malicious parties who launch fraudulent clicks may follow standard procedures by launching a series of chain clicks at periodic intervals. Alternatively, the process may be changeable and may mask its activities using various techniques, such as generating many clicks in a very spread and random way or generating clicks from different devices or countries. It is critical that the model detect and capture both forms of click fraud attacks.

It is worth noting that ad websites infected with click fraud rank significantly differently on the websites rank checker such as [89], with fraudulent parties targeting both large and small domains. Even well-known and reputable domains may be attacked [28]. Websites offering mobile-related content or adult content, for example, receive a high volume of illegitimate clicks, particularly late at night or early in the morning. Next are sites with entertainment and lifestyle content, which tend to have an increase in illegitimate clicks in the afternoons and evenings [35]. Definitely, other features have been obtained and engineered, but we now discuss the key aspects of the features used to identify illegitimate clicks.

5.2.1. Temporal Features

To analyze click patterns and behaviors, previous studies have investigated and evaluated temporal features to capture both short- and long-term click behaviors. This is because malicious parties frequently attempt to act logically and make consistent clicks across highly varying periods of time. Furthermore, several statistical approaches, such as calculating variance, skewness, and standard deviation, have been applied to the temporal features of the number of clicks at specific intervals of time. For instance, calculating the click variance at each time interval provides knowledge of a party’s click pattern, whereas click skewness determines the divergence of the number of clicks from a party’s average clicks at a given period [31]. Moreover, temporal features are constructed in a variety of ways, including breaking down the timestamp into further information, such as the month, day, minute, and second when the click occurred. Day has also been divided into different time intervals in several studies; for example, a day is divided into four 6 h categories: night (12 a.m.–5:59 a.m.), morning (6 a.m.–11:59 a.m.), afternoon (12 p.m.–5:59 p.m.), and evening (12 p.m.–5:59 p.m.) (6 p.m.–11:59 p.m.). Splitting a day into numerous parts helps determine the temporal patterns of clicks [30,32]. In addition, 1 h is also divided into four 15 min periods: first (0–14), second (15–29), third (30–44), and last (45–59). The third 15 min

period, for example, is the number of clicks between the 30th and 44th minute divided by the total number of clicks [34].

5.2.2. Spatial Features

The location of the clicker is associated with click fraud [29], particularly manual click fraud [42]. As a result, the geographic features of the click source, such as country, city, or area from which the click originated, are considered. Countries that generate a large number of clicks are investigated and evaluated as groupings or clusters, whereas clicks from exceptionally small countries are studied and analyzed individually. For example, distinct countries were analyzed, including the percentage of clicks generated by each city, in [35], and the number of clicks from the same country and the variance and skewness of these clicks were examined in [31]. The proportion of clicks from each country across various periods was also studied in [32].

5.2.3. Clicker Behavior Features

Before, websites simply banned all bots because of clear and poor bot-like behavior patterns, but as technology evolved, bot behavior became more complicated, demanding in-depth research and monitoring. Otherwise, organizations risk banning valid customers and good bots, and, worst, they risk bots taking over customer accounts and destroying the brand name. There are several indicators of bot behavior that every organization should be aware of. Speed is a useful indicator because robots are designed to outpace humans. It can also be a feature of bot behavior, in addition to unusual IP addresses or traffic from unfamiliar countries. Nevertheless, the complexity of bot behavior is developing and evolving on a daily basis, so it is essential to evaluate a typical user journey and consider what an unusual journey may look like [90].

As many bots attempt to mimic human behavior on advertisers' post-click sites, there has been little attention to the study of post-click behavior. Few studies have examined human-like behavior to determine whether the origin of clicks is real or a bot [33,50,56,67]. Mouse movements, keystroke data, surfing duration, number of pages viewed after clicking, and fields filled out are among the behavioral characteristics that have been investigated. It is vital to highlight the distinct differences between the activities and behaviors of a real human and a bot on an advertiser's site following the ad click. For example, in [36], a threshold was set for a number of events to distinguish between legitimate and illegitimate clicks. To be classified as legitimate, an ad click must satisfy at least one of the following criteria: (1) 30 s for surfing, 15 mouse events, and one click; (2) 30 s for surfing, 10 mouse events, one scroll event, and one click; and (3) 30 s for surfing, 10 mouse events, and two page visits. According to a recent study [91], bots are categorized into eight groups based on their behavior after the ad click. One example is sit-in bots, which are the most common. They leave a page after clicking and going to the landing page and after spending a certain amount of time on it; they do not do any actions on the advertiser's site. Another is long-distance lovers bots, which use VPNs or other impersonation techniques to appear to be users from particular locations. The third is scrapers, which produce 5000 clicks for each advertiser's site. These bots are designed to gather data and content from the site and spread it to other websites.

5.2.4. Medium and IP Features

To make things more complicated, bots are increasingly using the same technology as real humans. They use browsers with fingerprints; that are highly similar to human browsers and can, for example, leverage mobile phone farms to exploit real hardware rather than simulated hardware [92].

Several aspects of the medium, including the equipment or device used to produce the click, have been investigated to address this issue. This comprises a variety of factors, such as platform type (e.g., mobile phone or laptop), browser, OS, agent, and ISP. According to recent statistics, the Mozilla Firefox and Microsoft Edge browsers had the lowest percentage

of visits to bots' fraudulent advertisements in 2022, with 13.5% and 17.9%, respectively. Yandex and Opera, on the other hand, had the highest rates of ad fraud, with 29.4% and 26.3%, respectively. The Windows OS generated the most bot and ad fraud traffic. Desktops had the highest rate of bots and ad fraud in the second quarter of 2022, which was at 22.2%. Mobile ad bot traffic was 13.3%, approximately nine percentage points lower than desktop ad bot traffic [93]. Therefore, medium-related characteristics were analyzed using data obtained by JavaScript, such as browser type, language used, OS, and screen resolution, or data were collected in alternative ways for the purpose of identifying and examining malicious party patterns and preventing false clicks. Furthermore, IP-related features were obtained and engineered because the same IP can produce many sequential clicks or clicks at specific intervals. Bots may traverse millions of clean residential IP addresses. Usually, each IP address makes only one or two click requests before the bot switches on to another [94]. Some calculations, such as the variance of the IP clicks factor, were also done to assess IP patterns properly because it is suspicious that a single user, perhaps a click bot, performs a constant number of clicks from many IP addresses. Similarly, a clicker with a bad IP address, such as a scam or one on the IP blacklist, is extremely likely to create false clicks [36].

5.2.5. Ad-Related Features

Some studies have considered features associated with the clicked ad, such as information about the ad publisher, advertiser, ad campaign [26,28,30–32,41,48,65,73], as well as information about ad placement on the website, such as ad size on the screen and the distance between the ad and other elements of the web page [35,50,65]. Furthermore, it has been pointed out that investigating publishers' reputation is critical because some clicks originate from shady websites. Although historical features (e.g., the number of previous clicks at different periods, click rate of a specific city, and web pages in history) have been shown to have a higher descriptive power for click behavior than the contextual features of an advertisement (e.g., the location of the ad on screen and the number of ads in one web page) [37], this does not negate the ability of contextual features to clearly describe malicious fraud patterns and support the performance of click fraud detection models [38]. Furthermore, contextual features are critical in dealing with the issue of cold start. Contextual aspects are essential for predicting click identity for new users and advertisements.

5.3. Insights Derived

The primary findings of previous studies demonstrate that after several experiments with different models used to classify ad clicks as legitimate or illegitimate, some features offered more evident hints and clues to identify ad click fraud than others did, and these were considered vital indicators of fraud detection. These insights are explained in the following.

Notably, most previous studies did not identify the most significant critical features after experiments were conducted. Some did not even include FS approaches for identifying feature importance and choosing the best possible feature subset, or perhaps they did not declare this explicitly in their studies.

- Fraud detection relies heavily on features gathered from fine-grained time-series analysis [26,30,31,37,38,40,54,69,73]. The use of basic statistical calculations of temporal characteristics may preserve strong predictive ability across time while reducing the likelihood of overfitting the training data. The interval of the day (e.g., night and morning) is the most influential factor in [39], followed by the type of day. Time of day appears to be the least influential attribute;
- An adequate analysis of the spatio-temporal aspects of click traffic is required for effective fraud detection [29,30];
- Fraudulent clicks are more frequent in some countries (or finer-grained geographical locations) than in others. Geographical features, such as clicks from high-risk countries, might be used as fraud indicators [34];

- Clicks from the same device or IP address that are duplicated may be invalid clicks [48,69]. Similarly, a clicker with a bad IP address, which is on the IP blacklist, is extremely likely to create false clicks [36];
- The clicker's behavior on the advertised site (i.e., the user's browsing behavior data) is a critical aspect in verifying the clicker's identity [33,38,50]. Unfortunately, only a few previous studies have focused on investigating and monitoring clicker behavior on an advertiser's site; it is undoubtedly a significant signal that distinguishes fraudulent clicks from genuine ones;
- Another useful method for identifying fraudulent clicks is to observe mouse events. When visiting a website on a nonmobile platform, a human user must perform a mouse action at least once. The lack of mouse action identifies this click as invalid. However, this may not be the case for mobile platform users. Therefore, some methods were developed exclusively to test mouse event behavior on nonmobile platforms [36];
- One basic method for detecting fraudulent clicks is to check whether the computer supports JavaScript [36];
- It is necessary to track the number of ad clicks every day, as well as the frequency of each click, in order to make a reliable conclusion [60]. Duplicated clicks are most likely illegitimate clicks [34];
- Clicks that come from suspicious sites are highly likely to be fraudulent clicks [36];
- Historical features explain aggregated user behavior over time, and they are far more reliable than contextual features; however, contextual aspects are essential for predicting click identity for new users and advertisements [37];
- Ad placement control features, such as the acceptable number of ads on the screen, the visibility of the ad on the screen so that it is not hidden behind another element, the size of the ad displayed on the screen, and the distance between the ad and other clickable elements, such as buttons, are key aspects of click fraud detection [38].

6. Discussion

In this section, some of the points raised in the review of previous studies are presented, which can serve as recommendations or insights for ongoing and evolving attempts to detect click fraud effectively and robustly in future models or systems.

- There is a limited number of publicly available datasets, which may restrict the potential for wider and more in-depth investigations of PPC and click fraud. As a result, there is a need for fresh and diverse public datasets;
- All public datasets suffer heavily from negative class imbalance, which causes models to overclassify the greater class(es) because of the increased likelihood of their appearance; as a result, the model is biased toward the majority group. Dealing with this imbalance would require extensive processing;
- Clickers' behavior after visiting an advertiser's website has not been sufficiently explored, which is certainly an important factor in distinguishing human clicks from bot clicks. Most studies have not investigated behavioral aspects, as these are either not accessible in the (mainly public) datasets or are neglected and not of interest;
- Checking the legality of an IP address and whether it is on an IP blacklist may be valuable, but given the intelligence and sophistication of bots, it should not be heavily relied on, especially that bot creators use residential IP addresses to mask their bots as normal traffic. Bots can also spoof the IP addresses of other legitimate users;
- Integrating features as a combination of two or three features may be a promising step toward the more effective detection of click fraud and must be used with caution because it has been used in very few studies;
- According to Lyu et al. [95], current feature interaction models and techniques are broadly categorized as follows: (1) naive methods, which do not model feature interactions and use only raw features; (2) memorized methods, which explicitly present feature interactions as novel features and allocate trainable embeddings for them; and (3) factorized methods, in which raw features are transformed into latent vectors and

feature interactions are implicitly modeled across factorization functions. Most prior efforts to detect ad click fraud used the first two types of approaches; thus, including the third type and modeling feature interactions using factorization functions may yield excellent outcomes for click classification;

- Using numerous DL models may lead to superior classification and prediction performance and effectiveness. These can also be used to directly extract features, such as deep NNs, and an efficient feature set;
- Finally, detecting click fraud is a never-ending cat-and-mouse game, as bot creators are always seeking new methods to defeat bot detection systems. The process of building systems to identify and prevent click fraud should continue because bots are constantly evolving over time. Advanced solutions to prevent and deal with them are required.

7. Conclusions

Modern marketplaces are threatened by the widespread occurrence of click fraud. This happens when people click on ads to generate revenue for the application's publisher, not out of interest in the ads. Click fraud drains advertising expenses and threatens the future of the online advertising marketplace by diverting cash from legal partners (publishers). It exploits the PPC billing scheme to create illegitimate ad click requests; in 2021, click fraud accounted for \$42 billion in advertisers' losses [96,97]. Several successful attempts have been undertaken to detect and prevent this type of fraud. Among the most popular approaches is AI, namely, ML and DL, which have been extensively explored in this study. The focus has been on examining the features used previously for click identification (benign or fraudulent) and then determining the most critical ones that are considered strong evidence of click fraud. Nonetheless, as bots and technology evolve, click fraud detection and prevention systems must also evolve to keep up with the complexity of bots and deal with this problem rapidly and effectively.

Author Contributions: Conceptualization, M.A. and R.A.A.; methodology, M.A. and R.A.A.; validation, M.A. and R.A.A.; formal analysis, M.A. and R.A.A.; investigation, M.A. and R.A.A.; writing—original draft preparation, R.A.A.; writing—review and editing, M.A. and R.A.A.; visualization, R.A.A.; supervision, M.A.; project administration, M.A.; funding acquisition, R.A.A. All authors have read and agreed to the published version of the manuscript.

Funding: We thank SAUDI ARAMCO Cybersecurity Chair at Imam Abdulrahman Bin Faisal University (IAU) for supporting and funding this work.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Table A1. Literature review summery table. Note: Studies are listed in chronological order (2012–2022).

Ref.	Algorithms	Dataset	Used Features	Used Metrics (For the Best Preforming Model)
[35]	GBM + RF.	FDMA2012 BuzzCity Dataset.	% Clicks in periods, % Referred in periods, % Agent in periods, First 15 min present, Second 15 min present Third 15 min present, Last 15 min present, Avg and Std spiky ReAgCnIpCi, Avg and Std spiky ReAgCnIpCi in: (Morning, Afternoon, Night), Avg spiky referred URL, Avg and Std spiky Cid, Avg and Std spiky Cid, Avg spiky referred URL in: (Morning, Afternoon, Evening, Night), Avg and Std spiky agent, Avg and Std spiky IP, Avg and Std spiky ReAgCnIp, Avg and Std spiky Ag in: (Morning, Afternoon, Evening, Night), Cat number, Avg and Std, #Clicks, Avg and Std null Referrer in periods, Avg and Std #Clicks in 1 h, %Clicks certain Ctry, Unique agent, Unique, IP, #Clicks, Avg and Std null agent, Avg and Std null Referrer, Publisher ID, Unique Cid, Unique Ctry, and Unique Referrer.	Average Precision (AP) = 59.38%
[28]	DT.	Private -Crawled Dataset. 24,801,406 Ad-Paths.	Node roles, Node domain regs, Node frequency, and Node-pair frequency.	For “click-fraud pages”: False Detection (FD) = 14.61%, False Positive (FP) = 13. And for “click-fraud domain-paths” FD = 3.65%, FP = 125.
[29]	RF.	FDMA2012 BuzzCity Dataset.	#Clicks, #Clicks same IP, Unique IP, Category, Agent, Cid, Cntr, Long Click profile, Short click profile, and URL profile.	Accuracy: validation-set = 49.99%. and the test-set = 42.01%.
[30]	RF, REP tree, BN, Decision Table and NB.	Private Dataset.	#Advertisers, #Clicks, #Referrers, #Ips, #Clicks in night, #Clicks in morning, #Clicks in afternoon, #Clicks in evening, Click ratio on advertiser, Click/IP ratio, Click/cookie ratio, Var #Clicks same IP, Mean_Period_Click, Std_Period_Click, Popular area, Most_hour_click, Avg_Len_UA, and Avg_Lev_Referrer.	RF.Precision = 98.5%, and RF.Recall = 98.5%.
[31]	J48, RepTree, and RF.	FDMA2012 BuzzCity Dataset.	No_of_clicks_per_1min_Avg, No_of_clicks_per_1min_Max, No_of_clicks_per_1min_Var, No_of_clicks_per_1min_Skew, No_of_clicks_per_5min_Avg, No_of_clicks_per_5min_Max, No_of_clicks_per_5min_Var, No_of_clicks_per_5min_Skew, No_of_clicks_per_1hr_Avg, No_of_clicks_per_1hr_Max, No_of_clicks_per_1hr_Var, No_of_clicks_per_1hr_Skew, No_of_clicks_per_3hrs_Avg, No_of_clicks_per_3hrs_Max, No_of_clicks_per_3hrs_Var, No_of_clicks_per_3hrs_Skew,	Ensemble model. Accuracy = 59.39%.

Table A1. Cont.

Ref.	Algorithms	Dataset	Used Features	Used Metrics (For the Best Performing Model)
			No_of_clicks_per_6hrs_Avg, No_of_clicks_per_6hrs_Max, No_of_clicks_per_6hrs_Var, No_of_clicks_per_6hrs_Skew, Max #Clicks same IP, #Clicks unique IP, Click ratio on IP, Entropy #Clicks same IP, Var #Clicks same IP, Avg #Clicks same agent, Max #Clicks same Agent, Var #Clicks same agent, Skew #Clicks same agent, Max #Clicks same Ctry, Var #Clicks same Ctry, Skew #Clicks same Ctry, Max #Clicks same Cid, Var #Clicks same Cid, Skew #Clicks same Cid, and Channel.	
[32]	LR, and ERT.	FDMA2012 BuzzCity dataset.	#Clicks same IP ratio per 1 min, #Clicks ad ratio per 1 min, #Clicks agent ratio per 1 min, #Clicks ctry per 1 min, #Clicks Referrer ratio per 1 min, Channel, Publisher account, Publisher address, Click per (ad, device, ctry, and referrer) ratio, Gap interval btw clicks, Click fraction from top 20 ctry, Click fraction from non-top 20 ctry, Click fraction from UN/NA ctry, Click fraction from UN/NA referrer, and Click fraction from UN/NA Agent.	ERT.Average Precision (AP) = 55.64%.
[33]	SVM, LR, BN, MLP, C4.5, RepTree, and RF.	FDMA2012 BuzzCity Dataset.	Same as used features in [31], in addition to: Click ratio on IP, #Cid, Cid Entropy, Cid/Click Ratio, #Ctry, #Referrer, Ctry/Click Ratio, Ctry Entropy, Channel_Prior, Referrer/Click Ratio, and Non-Referrer/Click Ratio.	C4.5.Precision = 98.12%, Recall = 98.52, and F1-score = 98.32%.
[34]	Single: FT tree, REP tree, Bayes network, RPROP. Ensemble: LAD tree, NB tree, RF, Random subspace, Rotation Forest, Tree ensemble. Ensemble of ensemble: Blending.	FDMA2012 BuzzCity Dataset.	Unique Referrer, Unique Cid, Unique Ctry, #Clicks, Count IP in hour, Count IP&Agent in sec, Count IP&Agent in day, Count sub-IP in sec, Count sub-IP in min, Count sub-IP in hour, Count sub-IP in day, Count sub2-IP in day, Avg IP&Agent in min, Avg IP&Agent in day, Avg Cid in min, Agent-1, and Agent-2.	Ensembles of ensemble models.AP = 52.3%.
[36]	C4.5.	Real-World Collected Dataset: 9.9 thousand clicks.	Mouse clicks count, Mouse clicks on other pages, Mouse clicks on links, #Mouse scroll, #Mouse scroll on other pages, #Mouse moves, #Mouse moves on other pages, Page views, Visit duration, Execution efficiency, Browser Type, Legitimacy of IP, Publisher’s reputation, and Unique Ctry.	FP = 79%, False Negative Rate (FNR) = 5.6%, and Accuracy = 99.1%.

Table A1. Cont.

Ref.	Algorithms	Dataset	Used Features	Used Metrics (For the Best Performing Model)
[73]	RF.	508,005 HTTP requests.	# low entropy params # high entropy params Stddev(# param values to # apps) # unique params Avg(# params) Avg(# param values to # apps) # likely enum params Avg(depth in tree) # likely non-enum params Avg(tree height) # num-enum params % children with redirection Avg(response size) Stddev(# body urls) Stddev(depth in tree) Stddev(request size), Has parent, # enum params, status codes, the length of the requests, the length of the replies, timestamp, and publisher ID.	TPR = 71.8%, FPR = 00.1%, and Accuracy = 85.9%.
[37]	DT, and LR.	Facebook Ads Impression Dataset.	Confidential Features.	DT + LR. Normalized Entropy (NE) = 95.65%.
[38]	C4.5.	Apple iOS Apps 2012 Dataset: 13,267 game Apps.	App ID, Developer ID, Price, Category ID, App popularity, Release Date, Current Version, #Apps, Avg App Rating, Avg Number of App Versions, Avg pos_reviews, %Free apps, Number of Ad-controls, Visibility of Ad-controls, Size of Ad-control, Misplace of Ad-control on tiny screen, and User interest.	Precision = 87.6%, Recall = 71.3%, and F1-score = 76.8%.
[76]	RFE, and HDDT.	FDMA2012 BuzzCity Dataset.	Avg #Clicks per: (1 min, 5 min, 10 min, 1 hr, 2 h, and 6 h), Var #Clicks per: (1 min, 5 min, 10 min, 1 h, 2 h, and 6 h), Var #Clicks same agent, Var #Clicks same IP, Click ratio on: (IP, agent, cid, ctry, and referrer), Var #Clicks same Cid, and Var #Clicks same Ctry.	HDDT.Accuracy = 64.07%.
[39]	Decision Trees (DT), Naive Bayes (NB), and Support Vector Machines (SVM)	FDMA2012 user-click dataset.	Clicktime: Day, Hour of the day, Period of day, Type of day, and Month.	DT.Precision = 29.24%
[74]	NB, SVM, K-Nearest Neighbors (KNN), C4.5, and Random Forest (RF)	Websites dataset: 87 ad requests. Click dataset: 54 clicks.	To distinguish between an ad-related URL and a non-ad-related URL: 1st dataset: URL structure characteristics, additional URLs contained in the URL, and other page characteristics destination IP, content type, content length, status code, and location URL The number of query parameters, and their average length. 2nd dataset: NA.	RF.Avg. Accuracy = 99.61%, RF.Precision = 98.05%, RF.FP Rate= 0.27%
[40]	RF	FDMA2012 BuzzCity dataset.	Gap interval btw clicks in 1 h, #Clicks same IP in sec, Unique IP, #Clicks, and Click ratio on ctry.	Avg. Precision= 36.20%

Table A1. Cont.

Ref.	Algorithms	Dataset	Used Features	Used Metrics (For the Best Performing Model)
[43]	GBDT.	Private Dataset: 0.13 billion impression records, 8.5 million click records and about 1.85 million conversion records.	Clicks, Unique IP, Unique Ad-project, Unique Ad-position, Unique agent, Click ratio on: (IP, cookie Unique Ad-project, Unique Ad-position, and Unique agent.) , #Clicks in periods (evening and night), Var Period Click, Mean (var) clicks, cookie, agent, creative, project, and position in hrs, Mean (var) clicks, cookie, agent, creative, project, and position in mins, Mean (var) clicks, cookie, agent, creative, project, and position in sec, Mean (var) of intervals, Agent ID, and Channel.	In user classifier: Precision = 85.02%, Recall = 98.31%, and F1-score = 91.18%. In traffic classifier: Precision = 96.53%, Recall = 94.89%, and F1-score = 95.70%.
[44]	LR, DT, GBDT, and XGBOOST.	AdMaster Dataset. Train-set: 4,800,000. Test-set: 1,200,000	Click time, UAgent frequency, IP_Cookie frequency, IP frequency, Publisher ID, Campaign ID, Publisher frequency, Cookie frequency, and Born Cookie. (Note: There is insufficient description on the features involved in training the model.)	DT.Accuracy = 94.8%, DT.F1-score = 98.83%, and DT.Run-Time = 4621.50 s.
[63]	KNN, SVM, and ANN.	Private Dataset: 3247 instances. Train-set= 30%, and Test-set= 70%.	Var #Clicks per time intervals, #Clicks, #Clicks unique IP, Click ratio on IP, Suspicious click ratio, and App down ratio.	KBB.Accuracy = 98.26%, and F1-score = 99%.
[64]	AE.	TalkingData Dataset.	Ip, App Id, Agent, OS, Channel Id, click_time, Attributed time, and Is attributed.	In balanced dataset: Accuracy = 68%, Precision = 62.3%, Recall = 98%, and NPV (Negative predictive value) = 94%. In Un-balanced dataset: Accuracy = 91%, Precision = 100%, Recall = 92%, and NPV = 5%.
[65]	Cost sensitive neural BPNN (CSBPNN) + ABC.	FDMA2012 BuzzCity dataset.	#Clicks, Second 15 min present, Avg spiky ReAgCnIpCi, Avg spiky ReAgCnIpCi in: (Afternoon, Night), Avg spiky referred URL, Avg spiky referred URL in night, Avg spiky agent, Avg spiky IP, Std #Clicks in min, Std #Clicks in hr, Ctry present, and Status.	Accuracy = 99.14, Precision = 88.27%, Recall = 91.87%, and F1 score = 90.03%.
[53]	K-NN + TF-IDF algorithm.	Private online shop dataset. 30,000 Instances. Train-set: 80%, Validation-set: 10%, and Test-set: 10%.	#Mouse scroll, Mouse scrolling, #Mouse moves, Mouse movements, Page views, Pages in history, #Visits, Keystroke data, ISP, HTTP referrer, Screen resolution, Screen orientation, Language, IP, and Fingerprint	Accuracy = 97.11%.
[54]	SVM, K-NN, AdaBoost, DT, and Bagging.	Waseet Dataset: 500 clicks.	Gap interval in: (1 min, 1 s > 1 min, >1 s), Agent, Session ID, IP, #Clicks, Is clicked (Note: There is insufficient description on the features involved in training the model.)	KNN.Precision = 97.6%, KNN.Recall = 97.6%, KNN.TPR = 97.6%, and KNN.FPR = 0.034%.

Table A1. Cont.

Ref.	Algorithms	Dataset	Used Features	Used Metrics (For the Best Performing Model)
[41]	NB, DT, RF.	Private Dataset: 98570 CloudBot records and 164786 human records.	#Peer nodes, #Packets, #Bytes, #Flows, Max, Min, Med, Mean, Var, and STD of flows in packets, Max, Min, Med, Mean, Var, and STD of bytes in flow, Max, Min, Med, Mean, Var, and STD of bytes in flow duration, Max, Min, Med, Mean, Var, and STD of application layer protocol fields, Max segment size, TCP window size, TCP window scale, NOP, Time to Live, and Port.	RF.Precision = 90.8%, RF.TP = 29162, and RF.FP = 2954.
[55]	SVM.	Click dataset, and User Dataset.	IP, Click time, Click frequency, real evaluation score, click stream density, Gap interval btw clicks, Surfing time, total time spent (after click), students' evaluation information, and #Clicks. (Note: There is insufficient description on the features involved in training the model.)	Accuracy = 97.8%, Time complexity = 10%, and Arrival Rate = above 95%.
[56]	LR.	Public Kaggle AdTrackin TalkingData Dataset.	IP, App Id, OS, Agent, Channel Id. and Click time: minutes and seconds.	Normality Test (NT) = 0.0567, and Variance Test (VT) = 0.0598
[66]	ANN, AE, GAN, LR, SVM, RF, and Multinomial NB.	Public Kaggle AdTrackin TalkingData Dataset.	IP, Agent, Channel Id, App Id, OS, Click time, and Is attributed.	LR.Accuracy = 99.91%, LR.Recall = 99%, and LR.Precision = 99%.
[45]	LightGBM	Public Kaggle AdTrackin TalkingData Dataset.	OS, App Id, Agent, Channel Id, Day, Count: (IP, Hour, Day), (IP, App Id), ((IP, App Id, OS). Group by Count unique: IP Channel Id, (IP, Day Hour, IP App Id, (IP, App Id) OS, IP Agent, App Channel, (IP, Agent, OS) Channel Id. Group by Variance: ((IP, App Id, OS) hour, ((IP, App Id, Channel Id) Day, ((IP, App Id, Channel Id) Hour, and Next_click: (click_time, Channel)	Accuracy = 98%.
[67]	Sequential Minimal Optimization, Bagging, RF, Logistic Model Tree, and CNN.	Private- signals dataset: 50,000 click, constitutes 4,957,200 labelled time-domain signals.	Time-domain features 1-norm, Infinity norm, Fresenius Norm, Max, Min, RMS, RMSE, Mean, PNum, TNum, SMA, Skew, Kurt, ATP, and ATT. Frequency-domain features, including Amplitude spectrum, Power spectrum, Formant of the signal spectrum, etc. (Note: There is insufficient description on the Frequency-domain involved in training the model.)	CNN.Accuracy = 84.21%.
[75]	SVM, RF, NN.	Taobao platform dataset: 1100 instances.	Main good features: Store markdown, Sales, #Successful transactions, Collection, and Collection conversion rate. On-line shop feature: Deposit, Operating duration, Reputation level, Avg reputation score, Number of fans, Favourable rating, Number of reviews, Rate of reviews with pictures, Rate of additional reviews, and Collection conversion rate of.	Ensemble model.Accuracy = 97.4%, Ensemble model.Precision = 99.6%, Ensemble model.Recall = 90.1%, Ensemble model.F1-score = 94.5%, and Ensemble model.AUC = 98.8%.

Table A1. Cont.

Ref.	Algorithms	Dataset	Used Features	Used Metrics (For the Best Performing Model)
[46]	GTB.	FDMA2012 BuzzCity Dataset.	Avg spiky ReAgCnIpCi in: (Morning, Afternoon, Night), Std spiky ReAgCnIpCi in: (Morning, Afternoon, Night), #Clicks, Avg and Std #Clicks in min, Avg and Std #Clicks URL is null, %Clicks (Morning, Afternoon, Evening, Night), Avg and Std same URL_Ag_Ctry_Cid in 1 min, #Period_clicks/#clicks, %Brand_clicks/#clicks, and %Ctry_clicks/#clicks.	AP = 60.5%, Recall = 57.8%, and F1-score = 59.1%.
[68]	DNN.	Private Mobile-ad Dataset.	User ID, app ID, ad ID, geographical attributes: encrypted IP and city, action type, action time, device ID, device system models, and screen size. +MORE extracted features based on Window T. (Note: There is insufficient description on the features involved in training the model.)	Avg (For the whole dataset): Precision = 63%, and AUC = 95.65%
[69]	CSCNN.	FDMA2012 BuzzCity dataset.	Unique agent in periods T2, Avg and Std of Unique agent in periods T1, Agents' ratio in T2, Std same agent T2, Max same agent T2, Agent Entropy T2, Unique IP in periods T2, Avg and Std of Unique IP in periods T1, IPs' ratio in T2, Std same IP T2, Max same IP T2, IP Entropy T2, #Clicks in T2, Unique Cid in periods T2, Avg and Std of Unique Cid in periods T1, Cids' ratio in T2, Std same Cid T2, Max same Cid T2, Cid Entropy T2, Unique Ctry in periods T2, Avg and Std of Unique Ctry in periods T1, Ctry ratio in T2, Std same Ctry T2, Max same Ctry T2, Ctry Entropy T2, Unique Referred periods T2, Avg and Std of Unique Referred in periods T1, Referred ratio in T2, Std same Referred T2, Max same Referred T2, Referred Entropy T2, Unique IP&Agent periods T2, Avg and Std of Unique IP&Agent in periods T1, IP&Agent ratio in T2, Std same IP&Agent T2, Max same IP&Agent T2, and IP&agent Entropy T2.	Precision = 89%, Recall = 93%, F1-score = 92%, G-mean = 96.5%, and AUC = 93%.
[47]	SVM, KNN, DT, RF, and GBDT	University- dataset: 32119 instances, 8713 fraud clicks, 23406 benign clicks.	Referred URL, Server IP, IP, Agent, and Click time	GBDT.Accuracy = 97.20%, False Negative Rate (FNR) = 15.20%, True Negative Rate (TNR) = 93.10%, False Positive Rate (FPR)= 76.90%, and True Positive Rate (TPR) = 96.80%.
[48]	XGBoost	Private Dataset. Train: 75%, Test: 25%.	Day, Hour, IP, App Id, OS, and Agent.	Accuracy = 91%, Precision = 87%, Recall= 96%, and F1-score = 91%.

Table A1. Cont.

Ref.	Algorithms	Dataset	Used Features	Used Metrics (For the Best Performing Model)
[49]	MLP, NB, KNN, ML), Linear-SVC, and Heuristics	Public Kaggle AdTrackin TalkingData Dataset.	For MLP: App Id, Agent, OS, App Id, Channel, Hour, Day, and click count. For other models: App Id, Agent, OS, App Id, Channel, Hour, Day, Avg hourly click CvR, Avg App CvR, Avg OS CvR, Avg Agent CvR, Avg Channel CvR, #Clicks same IP, #Clicks unique IP, IP only one, Avg IP CvR, #Click IP&Hour&Day, #Click IP&App, #Click IP&App&OS, and Var click day by IP&App&Channel.	MLP.Precision = 95.43%, Training Time per Click = 284 ms, and Prediction Time per Click = 9.63 ms.
[42]	RF.	FDMA2012 BuzzCity dataset.	Same as Ref. [35]	Accuracy = 93%, acc+ = 93.24%, acc- = 92.15%, and G_means = 90.37%
[77]	RF.	1st Dataset: Public Kaggle AdTrackin TalkingData Dataset. 2nd Dataset: Avazu Click-Through Rate Dataset. 3rd Dataset: Criteo Dataset: 756,554.	1st Dataset: IP, App, Device, OS, Channel, Click day, Click hour, Click min, Click sec, Click ID, attributed_time, and is_attributed. 2nd Dataset: Hour, Site_Id, Site_cat, Site_domain, App, App_Id, App_cat, App_domain, device_Id, Device_Ip, Devic_model, Device_type, Device_conn_type, hour_of_click: day, hour, and month, Click_frequency_in_10hrs, C14-C21 (anonymized categorical variables). (Note: There is insufficient description on the 3rd dataset's features involved in training the model.)	ARFI: For 1st Dataset: Accuracy = 95.121%, Precision = 95.150%, Recall = 95.120%, F1-score = 95.010% and AUC = 91.401%. For 2nd Dataset: Accuracy = 83.328%, Precision = 79.260%, Recall = 83.330%, F1-score = 78.130% and AUC = 54.040%. For 3rd Dataset: Accuracy = 70.270%, Precision = 67.670%, Recall = 70.270%, F1-score = 67.210% and AUC = 59.381%. MRFI: For 1st Dataset: Accuracy = 95.121%, Precision = 95.150%, Recall = 95.120%, F1-score = 95.010% and AUC = 91.401%. For 2nd Dataset: Accuracy = 83.328%, Precision = 79.260%, Recall = 83.330%, F1-score = 78.130% and AUC = 54.040%. For 3rd Dataset: Accuracy = 70.270%, Precision = 67.670%, Recall = 70.270%, F1-score = 67.210% and AUC = 59.381%.
[70]	FCNN.	For Click monitoring = 300,000 logs, and for Lead monitoring = 263,000 logs.	For click monitoring: Date, Click time, Web Id, OS, Browser Type, ISP, Language, IP, #Mouse scroll, Mouse scrolling, Mouse movements, #Page views, and Keystroke data. For Lead monitoring: (In addition to the same "click monitoring" features): #Mouse moves, #Pages in history, #Pasted word, #Text_field clicked, and Filled form.	Accuracy = 99.5%, Precision = 99.5%, Recall = 99.5%, and F1-score = 99.5%.
[71]	VAEs.	20,000 Mobile apps.	#Axis APIs, #View-Size APIs, #Rand Num APIs, DDG size, and Rand condition.	RunTime: 18.42 s.

Table A1. Cont.

Ref.	Algorithms	Dataset	Used Features	Used Metrics (For the Best Performing Model)
[50]	CF + XGBoost.	Three data sets: Public Kaggle AdTrackin TalkingData Dataset. Avazu Dataset, and Kad Dataset: 1000 samples.	1st dataset: IP, App Id, OS, channel Id, Day, Hour, Minute, Second, and Unique IP in different time intervals. 2nd dataset: Site_Id, Site_cat, Site_domain, App, App_Id, App_cat, App_domain, Agent, Agent ID, IP, Conn type, Day, Hour, Month, Click frequency: (in 10 h.) 3rd dataset: Day, Hour, Second, Month, Age, Income, Daily Internet Usage, City, Cntry, Gender, and Is clicked.	For the 1st dataset: Precision = 96.45%, Recall = 96%, F1 = 96%, and AUC = 93.87%. 2nd dataset: Precision = 93%, Recall = 93%, F1 = 93%, and AUC = 87.37%. 3rd dataset= Precision = 95%, Recall = 94%, F1 = 94%, and AUC = 93.43%.
[57]	LR.	Public Kaggle AdTrackin TalkingData Dataset.	IP, and Click time.	NA.
[72]	LR, RNN, LightGBM, AdaBoost, XGBoost, and GB.	Public Kaggle AdTrackin TalkingData Dataset.	Ip, App Id, Agent, OS, Channel Id, Click frequency, Hour, Day, Attributed time, Click ID, IP/App Id, IP/Agent, IP/OS, IP/Channel Id, IP/Hour, IP/Day, and IP/Attributed time.	LR.Accuracy: 98.69%, Precison: 99%, Recall: 99%, and F1-score: 99%.
[51]	XGBoost, Gradient Boosting, and AdaBoost.	Two public Kaggle datasets: AdTrackin TalkingData and Advertising.	1st dataset: Ip, App Id, Agent, OS, Channel Id, Click time, Attributed time, and the IP feature was combined with all other features in combination of one/two attributes: IP/App Id, IP/Agent, IP/OS, IP/Channel Id, IP/Click time, and IP/Attributed time. 2nd dataset: Daily time, Age, Area, Ad Id, City, Ctry, Click time, and Is clicked.	XGBoost.Accuracy for 1st dataset: 96%, and for 2nd dataset: 93%.
[58]	Multimodal and Contrastive Learning.	Private Dataset from Alibaba. Train-set: 2.54 million clicks, and Test-set: 0.75 million clicks.	NA.	Precision = 98.7%, Recall = 85.4%, F1-score = 91.6%, and AUC = 93.3%.
[59]	HMSM.	Private Dataset.	Click time, Event Id, Event Timestamp, Event count, Max event count, Unique event, Unique event groups, Surfing time, and Day type. (Note: There is insufficient description on the features involved in training the model.)	Accuracy = 94%, precision = 99 %, recall of 95 %, specificity of 91%, and F-score of 97%.
[78]	RF.	1st Dataset: Public Kaggle AdTrackin TalkingData Dataset. 2nd Dataset: Avazu Click-Through Rate Dataset. 3rd Dataset: Criteo Dataset: 1,000,000.	1st Dataset: IP, App Id, Agent, OS, Channel Id, Day, Hour, Minute, Second, Click ID, Attributed time, and Is attributed. 2nd Dataset Site_Id, Site_cat, Site_domain, App, App_Id, App_cat, App_domain, Agent, Agent ID, IP, Conn type, Day, Hour, Month, Click frequency: (in 10 h), and C14-C21 (anonymized categorical variables). (Note: There is insufficient description on the 3rd dataset's features involved in training the model.)	For 1st Dataset: Accuracy = 91.04%, AUC = 88.35%, Recall = 83.02%, Precision = 81.27%, F1-score = 82.14%. For 2nd Dataset: Accuracy = 77.99%, AUC = 56.48%, Recall = 23.98%, Precision = 30.69%, F1-score = 26.92%. For 3rd Dataset: Accuracy = 64.63%, AUC = 62.79%, Recall = 57.83%, Precision = 45.19%, F1-score = 50.74%.

Table A1. *Cont.*

Ref.	Algorithms	Dataset	Used Features	Used Metrics (For the Best Performing Model)
[60]	LR and GNB.	Public Kaggle AdTrackin TalkingData Dataset.	IP, Agent, Channel Id, App Id, Attributedtime, OS, Click time, Is attribute, #Clicks daily, and Click frequency.	GNB.Accuracy = 99.76%, GNB.Recall = 99%, and GNB. F1-Score = 100%.
[61]	Catboost.	Private Dataset “mobile.de”: Train-set: 200,000 records. Test-set: 240,000 records. And a part of the public Kaggle AdTrackin TalkingData Dataset.	1st dataset: (Confidential 27 features). 2nd dataset: IP, App Id, Agent, OS, Channel Id, click time, Day, and hour.	On 1st dataset: F1-Score = 71.27%, Precision = 73.35%, Recall = 68.95%, and AUC = 98.09%. On 2nd dataset: F1-Score = 98.88%, Precision = 99.02%, Recall = 98.73%, and AUC = 99.94%.
[62]	QDPSKNN.	FDMA2012 BuzzCity Dataset (click dataset only)	Same as Ref. [35]	1st dataset: Precision = 75.1%, Recall = 70.1%, F1-Score = 72.5%, and G-mean = 73.3%.
[52]	Deep Convolutional Neural Network (DCNN), SVM, RF, DT, AdaBoost, and GBT.	FDMA2012 BuzzCity Dataset.	Same as Ref. [35]	GTB.Precision = 79.8%, GTB.Recall = 70.8%, GTB.F1-score = 72.4%, and GTB.AUC = 92.4%.

Appendix B

Table A2. A brief description of all features used in previous studies (included in the literature review).

Features	Descriptions
Temporal Features	
Click time	Timestamp of ad click in Coordinated Universal Time
Hour	Hour of day when the click was made
Minute	Minute of hour when the click was made
Second	Second of minute when the click was made
Period	Period of day when the click was made: midnight, early morning, morning, midday, evening, and night
#Clicks in periods	Total number of clicks in the evening (18:00–24:00) and at night (00:00–06:00)
% Clicks in periods	Percentage of clicks belonging to each publisher in the morning (during the hours 06:00:00–12:00:00 of the day), afternoon (during the hours 12:00:00–17:00:00 of the day), evening (during the hours 17:00:01–20:00:00 of the day), and night (during the hours 20:00–24:00:00 and 00:00:00–06:00:00 of the day)
%Referredin periods	Percentage of distinct referred URLs belonging to each publisher in the morning (during the hours 06:00:00–12:00:00 of the day), afternoon (during the hours 12:00:00–17:00:00 of the day), evening (during the hours 17:00:01–20:00:00 of the day), and night (during the hours 20:00–24:00:00 and 00:00:00–06:00:00 of the day)
%Agentin periods	Percentage of distinct agents belonging to each publisher in the morning (during the hours 06:00:00–12:00:00 of the day), afternoon (during the hours 12:00:00–17:00:00 of the day), evening (during the hours 17:00:01–20:00:00 of the day), and night (during the hours 20:00–24:00:00 and 00:00:00–06:00:00 of the day)
Month	Month of the year when the click was made
Day	Day of the week when the click was made
Day type	Weekday or weekend
Date	Date of the day when the click was made
Long click profile	#Clicks per day generated from the same IP address were measured in intervals lower than 5 s, from 5 s to 10 s, from 10 s to 20 s, from 20 s to 30 s, and so on up to 300 s; must be at least 10 clicks in the same IP
Short click profile	#Clicks per day generated from the same IP address were measured in intervals lower than 5 s, from 5 s to 10 s, from 10 s to 20 s, from 20 s to 30 s, and so on up to 300 s; must be at least five clicks in the same IP
URL profile	#Clicks per day generated from the same IP address were measured in intervals lower than 5 s, from 5 s to 10 s, from 10 s to 20 s, from 20 s to 30 s, and so on up to 300 s; must be at least five clicks in the same IP
Avg #Clicks per 1 min, 5 min, 10 min, 1 h, 2 h, 3 h, and 6 h	Average number of clicks for each partner during time intervals of 1 min, 5 min, 10 min, 1 h, 2 h, 3 h, and 6 h
Max #Clicks per 1 min, 5 min, 1 h, 3 h, and 6 h	Max number of clicks for each partner during time intervals of 1 min, 5 min, 10 min, 1 h, 2 h, 3 h, and 6 h
Var #Clicks per time intervals (1 min, 5 min, 10 min, 1 h, 2 h, 3 h, and 6 h)	Variance number of clicks for each partner during time intervals of 1 min, 5 min, 10 min, 1 h, 2 h, 3 h, and 6 h
Skew #Clicks per 1 min, 5 min, 1 h, 3 h, and 6 h	Skewness number of clicks for each partner during time intervals of 1 min, 5 min, 10 min, 1 h, 2 h, 3 h, and 6 h
#Clicks per 1 min	Number of clicks for each partner per 1 min
#Visitors per 1 min	Number of visitors for each partner per 1 min
#Ads per 1 min	Number of ads for each partner per 1 min

Table A2. Cont.

Features	Descriptions
#Agent per 1 min	Number of agents for each partner per 1 min
#Ctry per 1 min	Number of countries for each click per 1 min
#Referrer per 1 min	Number of referrer URLs for each partner per 1 min
#Clicks same IP ratio per 1 min	Percentage of the number of clicks from the same device IP per 1 min
#Clicks ad ratio per 1 min	Percentage of clicks from the ad banner per 1 min
#Clicks agent ratio per 1 min	Percentage of clicks from the device agent per 1 min
#Clicks ctry per 1 min	Percentage of clicks from the country per 1 min
#Clicks referrer ratio per 1 min	Percentage of clicks from the referrer URL per 1 min
Gap interval btw clicks	Time difference between two successive clicks
Gap interval btw clicks in 1 h, over 1 min, 1 s > 1 min	Time difference between two successive clicks in 1 h, less than 1 s, from 1 s to 1 min, and over 1 min)
Avg, STD, count IP in s	Average, standard deviation, and total IP visits per second
Avg, STD, count IP in min	Average, standard deviation, and total IP visits per minute
Avg, STD, count IP in h	Average, standard deviation, and total IP visits per hour
Count IP in hour	Total number of visits (>2) by IP each hour
Count IP&agent in s	Total number of visits (>2) by IP + agent per second
Count IP&agent in day	Total number of visits (>2) by IP + agent per day
Count sub-IP in s	Total number of visits (>2) by subnetwork (divided by 1,000,000) per second
Count sub-IP in min	Total number of visits (>2) by subnetwork (divided by 1,000,000) per minute
Count sub-IP in h	Total number of visits (>2) by subnetwork (divided by 1,000,000) per hour
Count sub-IP in day	Total number of visits (>2) by subnetwork (divided by 1,000,000) per day
Count sub2-IP in day	Total number of visits (>2) by subnetwork (divided by 1000) per day
Avg IP&agent in min	Average visit by IP + agent per minute
Avg IP&agent in day	Average visit by IP + agent per day
Avg Cid in min	Average visit by campaign ID per minute
#Clicks same IP in s	Number of clicks from the same IP address occurring every 10 s, every 20 s, every 30 s, and so on up to the interval [590 s, 600 s]
Mean (var) clicks, cookie, agent, creative, project, and position in h	Mean (variance) of clicks, cookie, agent, creative, project, and position in hours
Mean (var) clicks, cookie, agent, creative, project, and position in min	Mean (variance) of clicks, cookie, agent, creative, project, and position in minutes
Mean (var) clicks, cookie, agent, creative, project, and position in s	Mean (variance) of clicks, cookie, agent, creative, project, and position in seconds
Mean (var) of intervals	Mean (variance) of click intervals
Attributed time	Time of the first action performed by the user after clicking (e.g., download app)
First 15 min percent	Number of clicks from the 1st to the 14th minute divided by the number of clicks
Second 15 min percent	Number of clicks from the 15th to the 29th minute divided by the number of clicks
Third 15 min percent	Number of clicks from the 30th to the 44th minute divided by the number of clicks

Table A2. Cont.

Features	Descriptions
Last 15 min percent	Number of clicks from the 45th to the 59th minute divided by the number of clicks
Avg and STD spiky ReAgCnIpCi	Average and standard deviation of the number of the same referred URL, agent, country, IP, and Cid being duplicated in 1 min
Avg and STD spiky ReAgCnIpCi in morning, afternoon, and night	Average and standard deviation of the number of the same referred URL, agent, country, IP, and Cid being duplicated in 1 min in the morning (during the hours 06:00:00–12:00:00 of the day), afternoon (during the hours 12:00:00–17:00:00 of the day), evening (during the hours 17:00:01–20:00:00 of the day), and night (during the hours 20:00–24:00:00 and 00:00:00–06:00:00 of the day)
Avg and STD spiky referred URL	Average and standard deviation of the number of times the same referred URL is duplicated in 1 min
Avg and STD spiky Cid	Average and standard deviation of the number of times the same campaign ID is duplicated in 1 min
Avg and STD spiky Cid	Average and standard deviation of the number of times the same country is duplicated in 1 min
Avg spiky referred URL in morning, afternoon, evening, and night	Average and standard deviation of the number of the same agent being duplicated in the morning (during the hours 06:00:00–12:00:00 of the day), afternoon (during the hours 12:00:00–17:00:00 of the day), evening (during the hours 17:00:01–20:00:00 of the day), and night (during the hours 20:00–24:00:00 and 00:00:00–06:00:00 of the day)
Avg and STD spiky agent	Average and standard deviation of the number of times the same agent is duplicated in 1 min
Avg and STD spiky IP	Average and standard deviation of the number of times the same IP is duplicated in 1 min
Click frequency	Click frequency at different time intervals
Avg and STD spiky ReAgCnIp	Average and standard deviation of the number of the same referred URL, agent, country, and IP being duplicated in 1 min
Avg and std spiky ReAgCn	Average and standard deviation of the number of the same referred URL, agent, and country being duplicated in 1 min
Avg and STD spiky Ag in morning, afternoon, evening, and night	Average and standard deviation of the number of times the same agent being duplicated in 1 min in the morning (during the hours 06:00:00–12:00:00 of the day), afternoon (during the hours 12:00:00–17:00:00 of the day), evening (during the hours 17:00:01–20:00:00 of the day), and night (during the hours 20:00–24:00:00 and 00:00:00–06:00:00 of the day)
Cat number	Total number of clicks for a publisher in each category (which includes ad: adult sites, co: community, es: entertainment and lifestyle, gd: glamour and dating, in: information, mc: mobile content, pp: premium portal, and se: search and portals and services)
Avg and STD #Clicks	Average and standard deviation of the total clicks that occurred every minute for a particular publisher
Avg and STD null referrer in periods	Average and standard deviation of the clicks with null URL referrer, in the morning (during the hours 06:00:00–12:00:00 of the day), afternoon (during the hours 12:00:00–17:00:00 of the day), evening (during the hours 17:00:01–20:00:00 of the day), and night (during the hours 20:00–24:00:00 and 00:00:00–06:00:00 of the day)
Avg and STD#Clicks in 1 h	Average and standard deviation of a publisher's total clicks acquired in 1 h
Avg hourly click CvR	Average conversion rate of all clicks in 1 h.
#Clicks at T2	Total number of clicks made in the past by the same publisher across T2; T2: first 15 min (0–14), second 15 min (15–29), third 15 min (30–44), last 15 min (45–59), night (0:00–5:59), morning (6:00–11:59), afternoon (12–17:59), evening (18:00–23:59), and 3 days.
Avg clicks at night, morning, afternoon, and evening	Average number of clicks during the night (24:00–5:59), morning (6:00–11:59), afternoon (12:00–17:59), and evening.
%Clicks at night, morning, afternoon, and evening	Percentage of the number of clicks during the night (24:00–5:59), morning (6:00–11:59), afternoon (12:00–17:59), and evening (18:00–23:59).
Mean period click	Mean of the number of clicks in four periods: night (24:00–5:59), morning (6:00–11:59), afternoon (12:00–17:59), and evening (12:00–17:59) (18:00–23:59).

Table A2. Cont.

Features	Descriptions
STD period click	Standard deviation of clicks in four periods: night (24:00–5:59), morning (6:00–11:59), afternoon (12:00–17:59), and evening (12:00–17:59) (18:00–23:59).
Var period click	Variance of clicks in four periods: night (24:00–5:59), morning (6:00–11:59), afternoon (12:00–17:59), and evening (12:00–17:59) (18:00–23:59).
Most hour click	Maximum number of clicks in 1 h for each publisher.
Avg and STD same URL_Ag_Ctry_Cid in 1 min	Average and standard deviation of clicks generated by each publisher in 1 min when the same field (referred URL, agent, Ctry, IP, and Cid) is duplicated.
#Period_clicks/#clicks	Total number of clicks produced by each publisher between the hours of (0–14), (15–29), (30–44), and (45–59) divided by the total number of clicks.
Spatial Features	
Ctry	Countries from which the clicks received by each publisher were made
#Ctry	Total number of countries from which every publisher received clicks
%Ctry	Percentage of countries from which every publisher received clicks
Unique Ctry	Distinct countries.
%Clicks certain Ctry	Percentage of clicks coming from different countries (AZ, ID, IN, US, NG, TR, RU, TH, SG, UK, and others) out of the total clicks for a particular publisher
Popular area	The most common area where clicks are generated.
Ctry/click ratio	Average click ratio for each country.
Ctry entropy	Entropy of the click distribution in each country.
Max #Clicks same Ctry	Total number of clicks originating from the same country; the maximum of this is then obtained.
Var #Clicks same Ctry	Total number of clicks originating from the same country; the variance of this is then obtained.
Skew #Clicks same Ctry	Total number of clicks originating from the same country; the skewness of this is then obtained.
Click fraction from top 20 ctry	Fraction of the number of clicks from the top 20 countries (top 20 click-producing countries)
Click fraction from non-top 20 ctry	Fraction of the number of clicks from the other/non-top 20 countries (least 20 click-producing countries)
Click fraction from UN/NA ctry	Fraction of the number of clicks from unknown/empty countries
City	Cities from which the clicks received by each publisher were made.
Area	City district in which the user resides.
Unique Ctry at period T2	Number of different countries associated with the publisher at T2; T2: first 15 min (0–14), second 15 min (15–29), third 15 min (30–44), last 15 min (45–59), night (0:00–5:59), morning (6:00–11:59), afternoon (12–17:59), evening (18:00–23:59), and 3 days.
Avg and STD of unique Ctry at period T1	Average and standard deviation of the number of different countries for publishers at T1; T1: per minute, 5 min, 15 min, hour, 3 h, 6 h, and 1 day.
Ctry ratio at T2	Ratio of different countries to clicks at T2; T2: first 15 min (0–14), second 15 min (15–29), third 15 min (30–44), last 15 min (45–59), night (0:00–5:59), morning (6:00–11:59), afternoon (12–17:59), evening (18:00–23:59), and 3 days.
STD same Ctry T2	Standard deviation of the number of clicks for different countries at T2; T2: first 15 min (0–14), second 15 min (15–29), third 15 min (30–44), last 15 min (45–59), night (0:00–5:59), morning (6:00–11:59), afternoon (12–17:59), evening (18:00–23:59), and 3 days.
Max same Ctry T1	Maximum number of clicks for different countries at T1; T1: per minute, 5 min, 15 min, hour, 3 h, 6 h, and 1 day.
Ctry entropy T2	Number of clicks entropy corresponding to different countries at T2; T2: first 15 min (0–14), second 15 min (15–29), third 15 min (30–44), last 15 min (45–59), night (0:00–5:59), morning (6:00–11:59), afternoon (12–17:59), evening (18:00–23:59), and 3 days.

Table A2. Cont.

Features	Descriptions
#Click from differ Ctry	Number of clicks from different countries.
User Behavior Features	
Event ID	Unique identifier of the action taken by the user after clicking on the ad, such as watching a video or making a purchase
Event timestamp	The time the user's action occurred after clicking on the ad
Event count	Number of times the user has undertaken a particular action
Max event count	Sum of the number of times the user has undertaken a particular action; its maximum is then obtained.
Unique event	Number of times the user has undertaken a previously unknown and unique given action
Unique event groups	Number of times the user has undertaken a previously unknown and unique given action within a certain event group (e.g., a group of mouse events and a group of text events)
Surfing time	Duration of time the user has spent surfing the advertiser's website in seconds
Mouse clicks count	Total number of clicks made on the advertised website
Mouse clicks on other pages	Total number of clicks done on pages other than the landing page.
Mouse clicks on links	Total number of clicks on hyperlinks
#Mouse scroll	Total number of scroll events
Mouse scrolling	Monitoring page scrolling, such as scroll-up and scroll-down
#Mouse scroll on other pages	Total number of scroll events done on pages other than the landing page
#Mouse moves	Total number of mouse movement events, such as the number of left, center, and right mouse button clicks
#Mouse moves on other pages	Total number of scroll movements done on pages other than the landing page
Mouse movements	Monitoring the data of mouse movements, such as mouse right clicks, mouse double clicks, mouse movements, and area of the moved mouse
#Page views	Total number of pages viewed by the user
#Pages inlanguages	Total number of pages in the browser history
#Visits	Total number of visits to the ad web pages
Visit duration	Amount of time the user spends on the site
Execution efficiency	Client's JavaScript code execution time for a given task
User interest	Determine whether the user interest ratio is below a predefined threshold $\text{User Interest} = \frac{\text{no. of times Ads are visited by users} > 1 \text{ sec}}{\text{no. of clicks on Ads by users}}$
Keystroke data	Monitoring the keys struck on a keyboard, such as the number of pressed keys and the number of switches between text fields on the page using the tab key
#Pasted word	Number of pasted words from the clipboard
#Text_field clicked	Number of text fields clicked
Filled form	Monitor form fill events, such as the number of controls in the form that must be filled, the number of fields replaced with new values in the controls, the number of texts changed while editing, the number of fields filled with no keys pushed, and the number of fields filled quickly
Medium Features	
Avg Len UA	Average user agent length
Avg Len referrer	Average referrer URL length
Agent	Phone/device models and types used by clickers
Agent ID	Unique identifier of a given device agent to distinguish between phone and computer platforms

Table A2. Cont.

Features	Descriptions
Unique agent	Distinct device agents
Unique agent at period T2	In T2, the number of distinct devices related to the publisher at T2: first 15 min (0–14), second 15 min (15–9), third 15 min (30–44), last 15 min (45–9), night (0:00–5:59), morning (6:00–11:59), afternoon (12–17:59), evening (18:00–23:59), and days
Avg and STD of unique agent at period T1	Average and standard deviation of the number of different devices for publishers at T1: per minute, 5 min, 15 min, 1 h, 3 h, 6 h, and 1 day
Agent’s ratio at T2	Ratio of different devices to clicks at T2; T2: first 15 min (0–14), second 15 min (15–29), third 15 min (30–44), last 15 min (45–59), night (0:00–5:59), morning (6:00–11:59), afternoon (12–17:59), evening (18:00–23:59), and 3 days
STD same agent T2	Standard deviation of the number of clicks for different devices at T2; T2: first 15 min (0–14), second 15 min (15–29), third 15 min (30–44), last 15 min (45–59), night (0:00–5:59), morning (6:00–11:59), afternoon (12–17:59), evening (18:00–23:59), and 3 days
Max same agent T2	Maximum number of clicks for different devices at T1; T1: per minute, 5 min, 15 min, 1 h, 3 h, 6 h, and 1 day.
Agent entropy T2	Number of clicks entropy corresponding to different devices at T2; T2: first 15 min (0–14), second 15 min (15–29), third 15 min (30–44), last 15 min (45–59), night (0:00–5:59), morning (6:00–11:59), afternoon (12–17:59), evening (18:00–23:59), and 3 days
Agent entropy	Entropy of the click distribution on agent attribute
Avg #Clicks same agent	Number of clicks made by the same device agent; the average of this is then obtained.
Max #Clicks same agent	Number of clicks made by the same device agent; the maximum of this is then obtained.
Var #Clicks same agent	Number of clicks made by the same device agent; the variance of this is then obtained.
Skew #Clicks same agent	Number of clicks made by the same device agent; the skewness of this is then obtained.
Click fraction from UN/NA agent	Fraction of the number of clicks from the unknown/empty device model/agent used by the user to click
Agent-1	Statistics for click data sorted by click time and agent as follows: Clicks are arranged by click time, followed by agents. Each click log row is then compared with the following click log row. If the device is the same, the current row is preserved; otherwise, it is eliminated. The filter row portion is then computed over the row total for each publisher.
Agent-2	Statistics for click data sorted by agent only, calculated in the same way as Agent-1 (above feature)
Browser type	Names of the browsers used by the ad clickers
OS	OS version ID of the user’s phone/computer
Web ID	A unique identifier and a mechanism for Internet service providers and members to recognize who they are connecting with
App ID	Unique identifier of a given application that contains an advertisement
App price	Price of the application that contains an advertisement
Category ID	A unique identifier of the category to which the application belongs
App popularity	Popularity of the application among users, from the number of times it has been downloaded and its ratings
Release date	Application launch date
Current version	Current version number of the application
ISP	Internet service provider for the user’s connection
Conn type	Denotes whether the connected individual is a generator, distributor, or a user, as well as whether the connection connects with the grid, a local network, or an embedded network
HTTP referrer	The heading HTTP referrer
Screen resolution	Screen resolution of the user agent
Screen orientation	Screen orientation (vertical or horizontal) of the user agent

Table A2. Cont.

Features	Descriptions
Language	Information on the language of the agent, browser, or app
Session ID	A unique identifier for the session by the user
%Brand	Percentage of clicks coming from each agent brand (e.g., MAUI, Nokia, generic, Apple, Blackberry, Samsung, Sony, LG, and other brands) out of the total clicks for a particular publisher
UAgent frequency	User agent frequency at different time intervals
IP_Cookie frequency	Reveals the number of cookies in each IP address at different time intervals A cookie is a piece of data from a website that is stored within a web browser and can be accessed by the website later.
Max segment size	A Transmission Control Protocol (TCP) header option field parameter.
TCP window size	In each instance, determines the window size and frequency of appearance in synchronized packets during the TCP handshake
TCP window scale	A TCP header option field parameter typically used in conjunction with window size to assist in determining the OS type
NOP	The no-option (NOP) number is one byte in size and is used to fill the TCP options field in order to boost the packet length by a fourfold factor. Each operating system has its own set of TCP NOP settings.
Time to live	A timer value that tells the recipient how long to hold before discarding and expiring the packet A timer value that instructs the recipient how long to keep the packet until dismissing and expiring it
Port	Examine whether the click generator (human or bot) has a proclivity to use specific ports, such as TCP or User Datagram Protocol (UDP).
Avg App CvR	Average conversion rate of the app ID
Avg OS CvR	Average conversion rate of the OS ID
Avg agent CvR	Average conversion rate of the agent ID
Avg channel CvR	Average conversion rate of the channel ID
Unique IP&agent periods T2	Number of different IPs + agents associated with the publisher at T2; T2: first 15 min (0–14), second 15 min (15–29), third 15 min (30–44), last 15 min (45–59), night (0:00–5:59), morning (6:00–11:59), afternoon (12–17:59), evening (18:00–23:59), and 3 days
Avg and STD of unique IP&agent at period T1	Average and standard deviation of the number of different IPs + agents for publishers at T1; T1: per minute, 5 min, 15 min, 1 h, 3 h, 6 h, and 1 day
IP&agent ratio at T2	Ratio of different IPs + agents to clicks at T2; T2: first 15 min (0–14), second 15 min (15–29), third 15 min (30–44), last 15 min (45–59), night (0:00–5:59), morning (6:00–11:59), afternoon (12–17:59), evening (18:00–23:59), and 3 days
STD same IP&agent T2	Standard deviation of the number of clicks for different IPs + agents at T2; T2: first 15 min (0–14), second 15 min (15–29), third 15 min (30–44), last 15 min (45–59), night (0:00–5:59), morning (6:00–11:59), afternoon (12–17:59), evening (18:00–23:59), and 3 days
Max same IP&agent T1	Maximum number of clicks for different IPs + agents at T1; T1: per minute, 5 min, 15 min, 1 h, 3 h, 6 h, and 1 day
IP&agent entropy T2	Number of clicks entropy corresponding to different IPs + agents at T2; T2: first 15 min (0–14), second 15 min (15–29), third 15 min (30–44), last 15 min (45–59), night (0:00–5:59), morning (6:00–11:59), afternoon (12–17:59), evening (18:00–23:59), and 3 days
%Brand_clicks/#clicks	Percentage of clicks received by each brand divided by the total number of clicks produced by each publisher.
Avg and STD null agent	Average and standard deviation of the total clicks with null as agent that occurred every minute for a particular publisher
IP Features	
IP	IP addresses' clicks

Table A2. Cont.

Features	Descriptions
Server IP	IP address of the ad server
Unique IP	Distinct IP addresses' clicks per publisher
Unique IP at different time intervals	Distinct IP addresses' clicks per publisher in 1 h and 10 h
Unique IP parts	Distinct parts of the IP addresses
Legitimacy of IP	Whether the originating IP address is on a blacklist
Unique IP at period T2	Number of different IP addresses associated with the publisher at T2; T2: first_15 min (0–14), second_15 min (15–29), third_15 min (30–44), last_15 min (45–59), night (0:00–5:59), morning (6:00–11:59), afternoon (12–17:59), evening (18:00–23:59), and 3 days
Avg and STD of unique IP at period T1	Average and standard deviation of the number of different IP addresses for publishers at T1; T1: per minute, 5 min, 15 min, 1 h, 3 h, 6 h, and 1 day
IPs' ratios at T2	Ratio of different IP addresses to clicks at T2; T2: first 15 min (0–14), second 15 min (15–29), third 15 min (30–44), last 15 min (45–59), night (0:00–5:59), morning (6:00–11:59), afternoon (12–17:59), evening (18:00–23:59), and 3 days
STD same IP T2	Standard deviation of the number of clicks for different IP addresses at T2: first 15 min (0–14), second 15 min (15–29), third 15 min (30–44), last 15 min (45–59), night (0:00–5:59), morning (6:00–11:59), afternoon (12–17:59), evening (18:00–23:59), and 3 days
Max same IP T2	Maximum number of clicks for various IP addresses at T1; T1: per minute, 5 min, 15 min, 1 h, 3 h, 6 h, and 1 day
IP entropy T2	Number of clicks entropy corresponding to different IP addresses at T2; T2: first 15 min (0–14), second 15 min (15–29), third 15 min (30–44), last 15 min (45–59), night (0:00–5:59), morning (6:00–11:59), afternoon (12–17:59), evening (18:00–23:59), and 3 days
IP frequency	IP frequency at different time intervals
Number of Clicks Features	
#Clicks	Total number of clicks, count of rows in the click log.
#Clicks daily	Total number of clicks, count of rows in the click log on a daily basis
Click ID	A unique identifier for the ad click by the user
Avg and STD #Clicks in min	Average and standard deviation of the total clicks that occurred every minute for a particular publisher.
Avg and STD #Clicks in h	Average and standard deviation of the total clicks that occurred every minute for a particular publisher
Avg and STD #Clicks URL is null	Average and standard deviation of the total clicks produced by each publisher with null as a referrer that occurred every minute during the night, morning, afternoon, and evening
#Clicks same IP	Number of clicks originating from the same computer IP address
#Clicks unique IP	Number of clicks originating from different/unique computer IP addresses
IP only one	Whether an IP address has only one ad click
Avg IP CvR	Average conversion rate of an IP having #clicks
Max #Clicks same IP	Number of clicks originating from the same computer IP address; the maximum of this is then obtained.
Entropy #Clicks same IP	Number of clicks originating from the same computer IP address; the entropy of this is then obtained.
Var #Clicks same IP	Number of clicks originating from the same computer IP address; the variance of this is then obtained.

Table A2. Cont.

Features	Descriptions
Click ratio on advertiser, IP, cookie, agent, cid, ctry, referrer, Unique Ad-project, Unique Ad-position, and Unique agent	Click ratio for each #Clicks/#Advertisers, #Clicks/#IP, #Clicks/#cookie, #Clicks/#agent, #Clicks/#Cid, #Clicks/#Country, #Clicks/#Referrer, #Clicks/# Unique Ad-project, #Clicks/# Unique Ad-position, and #Clicks/# Unique Ad-agent
Click per ad, device, ctry, and referrer ratio	Percentage of the number of clicks per ad, device, and country
Suspicious click ratio	Percentage of the duration of suspicious clicks per publisher
#Clicks in channel	Total number of clicks produced by each publisher across all channels
%Ctry_clicks/#clicks	Percentage of clicks from every country divided by the total number of clicks produced by each publisher
Avg and std null referrer	Average and standard deviation of the total clicks with null as the referrer URL that happened for a particular publisher in every minute
#Click IP&Hour&Day	Total number of clicks grouped by IP address, day, and hour of the click
#Click IP&App	Total number of clicks grouped by IP address and app ID of the click
#Click IP&App&OS	Total number of clicks grouped by IP address, OS, and app ID of the click
Var click day by IP&App&Channel	Variance in click days for an IP, app, and channel
Provider/Advertiser/Ad Features	
Publisher ID	Unique identifier of the particular publisher of an advertisement campaign
Campaign ID	Unique identifier of a given advertisement campaign
Ad ID	Unique identifier of a given advertisement for all commercial assets
#Cid	Total number of campaign IDs associated with the clicks received by each publisher
Cid entropy	Entropy of the click distribution on campaign IDs
Cid/click ratio	Average percentage of the number of clicks for each campaign
Unique Cid	Distinct number of campaign IDs
Max #Clicks same Cid	Number of clicks received by the same campaign ID; the maximum of this is then obtained.
Var #Clicks same Cid	Number of clicks received by the same campaign ID; the variance of this is then obtained.
Skew #Clicks same Cid	Number of clicks received by the same campaign ID; the skewness of this is then obtained.
Publisher address	Mailing address of the publisher
Publisher account	Bank account associated with the publisher
Publisher's reputation	Whether the click occurs from an untrustworthy website
Visibility of ad controls	Determine whether the ad is clearly visible on the screen and not hidden behind a button or is off-screen
Size of ad control	Determine whether an ad is too small on the screen, making it difficult for users to read or locate it
Misplace of Ad-control on tiny screen	Examine to verify that the distance between an ad control and a clickable non-ad item is less than a certain threshold
Unique ad project	Number of unique ad projects
Unique ad position	Number of unique ad positions
Publisher frequency	Publisher ID frequency at different time intervals
Other Features	
Channel ID	Unique identifier of a given ad publisher

Table A2. Cont.

Features	Descriptions
Channel	Channel type of the publisher, which includes ad: adult sites, co: community, es: entertainment and lifestyle, gd: glamour and dating, in: information, mc: mobile content, pp: premium portal, and se: search and portals and services.
Referred URL	The URL referrer's web page that contains the ad content
#Referrer	Total number of URL referrers from which every publisher received clicks
Unique referrer	Distinct URL referrers
Channel_Prior	Prior fraud probability in each category
Referrer/click ratio	Average click ratio for each referrer URL
Non-referrer/click ratio	Total number of non-referred URL clicks with respect to the total clicks received for each publisher
Click fraction from UN/NA referrer	Fraction of the number of clicks from unknown/empty referrer URLs
Developer ID	Unique identifier of a given developer who developed the app that contained the advertisement
#Apps	Number of apps developed by the same developer
Avg app rating	Average rating of the app by users
Avg app versions	Average number of versions of the application
Avg pos_reviews	Average rate of positive reviews and helpfulness of the application
%Free apps	Percentage of free apps developed by the same developer
No. of ad controls	Recognize the number of ads displayed on the screen; it must be one ad per phone screen and three ads per tablet screen at most.
App down ratio	Ratio of app downloads per publisher
Is attributed	Target class of prediction, indicating whether the action was done after clicking
Is clicked	Target class of prediction, indicating whether the ad was clicked
Status	Status that describes the click's legitimacy (benign/fraud)
Fingerprint	An identification based on the JavaScript language parameters accessible in the browser
Store markdown	Difference between a product's initial and current prices
#Successful transactions	Amount of goods exchanged after the seller has delivered the goods and the buyer has validated receipt of the goods
Deposit	Margin offered to the Taobao platform by the online retailer
Dynamic score	A careful examination of the store based on three criteria: commodity quality, service attitude, and logistical service
Number of additional reviews	Number of reviews written by purchasers after they have used a product or service for a length of time
Collection conversion rate	Ratio of the number of successful transactions of major goods in a store to the number of collections throughout time
Avg reputation score	A shop's reputation score during the period of its operation
Rate of reviews with pictures	Number of reviews with photos divided by the total number of reviews for the store
Rate of additional reviews	Number of added reviews divided by all of the store's reviews
Collection conversion rate of secondary goods	Ratio of secondary products sales at a store to the number of collections over time
Operating duration	Length of time the store has been operating since its formation
Reputation level	A store's reputation level over the period of its operation
Favorable rating	When visitors have a positive opinion or reaction to something, it means that they agree with it and appreciate it.

Table A2. Cont.

Features	Descriptions
Site ID	A unique identification number assigned by the ministry to a site listed in the site registry
Site cat	The category to which the site belongs, such as e-commerce
App cat	The category to which the app belongs, such as entertainment
App dom	A method used within the common language infrastructure to isolate running software applications so that they do not interfere with one another
Age	Age of the Internet user who clicked on the ad
Gender	Gender of the Internet user who clicked on the ad
Income	Income of the Internet user who clicked on the ad
Daily Internet usage	Daily Internet consumption of the user
Daily time	Duration of time a user spends on the Internet on a daily basis
Unique Cid at period T2	Number of different campaign IDs associated with the publisher at T2; T2: first 15 min (0–14), second 15 min (15–29), third 15 min (30–44), last 15 min (45–59), night (0:00–5:59), morning (6:00–11:59), afternoon (12–17:59), evening (18:00–23:59), and 3 days
Avg and STD of unique Cid at period T1	Average and standard deviation of the number of different campaign IDs for publishers at T1; T1: per min, 5 min, 15 min, 1 h, 3 h, 6 h, and 1 day
Cids' ratio at T2	Ratio of different campaign IDs to clicks at T2; T2: first 15 min (0–14), second 15 min (15–29), third 15 min (30–44), last 15 min (45–59), night (0:00–5:59), morning (6:00–11:59), afternoon (12–17:59), evening (18:00–23:59), and 3 days
STD same Cid T2	Standard deviation of the number of clicks for different campaign IDs at T2; T2: first 15 min (0–14), second 15 min (15–29), third 15 min (30–44), last 15 min (45–59), night (0:00–5:59), morning (6:00–11:59), afternoon (12–17:59), evening (18:00–23:59), and 3 days
Max same Cid T1	Maximum number of clicks for different campaign IDs at T1; T1: per minute, 5 min, 15 min, 1 h, 3 h, 6 h, and 1 day
Cid entropy T2	Number of clicks entropy corresponding to different campaign IDs at T2; T2: first 15 min (0–14), second 15 min (15–29), third 15 min (30–44), last 15 min (45–59), night (0:00–5:59), morning (6:00–11:59), afternoon (12–17:59), evening (18:00–23:59), and 3 days
Unique referred periods T2	Number of different referred URLs associated with the publisher at T2; T2: first 15 min (0–14), second 15 min (15–29), third 15 min (30–44), last 15 min (45–59), night (0:00–5:59), morning (6:00–11:59), afternoon (12–17:59), evening (18:00–23:59), and 3 days
Avg and std of unique referred at period T1	Average and standard deviation of the number of different referred URLs for publishers at T1; T1: per minute, 5 min, 15 min, hour, 3 h, 6 h, and 1 day
Referred ratio at T2	Ratio of different referred URLs to clicks at T2; T2: first 15 min (0–14), second 15 min (15–29), third 15 min (30–44), last 15 min (45–59), night (0:00–5:59), morning (6:00–11:59), afternoon (12–17:59), evening (18:00–23:59), and 3 days
STD same referred T2	Standard deviation of the number of clicks for different referred URLs at T2; T2: first 15 min (0–14), second 15 min (15–29), third 15 min (30–44), last 15 min (45–59), night (0:00–5:59), morning (6:00–11:59), afternoon (12–17:59), evening (18:00–23:59), and 3 days
Max same referred T1	Maximum number of clicks for different referred URLs at T1; T1: per minute, 5 min, 15 min, hour, 3 h, 6 h, and 1 day
Referred entropy T2	Number of clicks entropy corresponding to different referred URLs at T2; T2: first 15 min (0–14), second 15 min (15–29), third 15 min (30–44), last 15 min (45–59), night (0:00–5:59), morning (6:00–11:59), afternoon (12–17:59), evening (18:00–23:59), and 3 days
#Axis APIs	Number of APIs for determining the actual click coordinates generated by users
#View-size APIs	Number of APIs for determining the size of an ad view; acquiring the size of an ad view in order to position the coordinates of the false click within the ad view
#Rand num APIs	Number of APIs for generating random numbers
DDG size	Size of the data dependence graph (DDG); the larger the size of the DDG, the more the data have been processed, indicating fraudulent behavior

Table A2. Cont.

Features	Descriptions
Rand condition	Random numbers in condition expression
Node roles	Identifying exploited servers by nodes, especially those with URLs that do not follow well-known advertising URL patterns Each URL discovered during data crawling is referred to as a node.
Node domain regs	Determine node domain registration times, as most malware domains expire within a year after registration Each URL discovered during data crawling is referred to as a node.
Node frequency	The popularity and reliability of node domains are measured; recognize each node’s domain and calculate the number of distinct publishers linked to this domain every day. Each URL discovered during data crawling is referred to as a node.
Node pair frequency	Analyze the frequency of two adjacent nodes on ad paths Each URL discovered during data crawling is referred to as a node.
Cookie frequency	Cookie frequency at different time intervals A cookie is a piece of data from a website that is stored within a web browser and can be accessed by the website later.
Born cookie	The time the cookie was generated A cookie is a piece of data from a website that is stored within a web browser and can be accessed by the website later.
#Peer nodes	Number of peer nodes, derived from raw traffic information Peers are other nodes that operate in the same manner as the nodes in the network.
#Packets	Number of packets transmitted through the network
#Bytes	Sum of the number of bytes for all the packets transmitted through the network
#Flows	Number of flows made over the network
Max, min, med, mean, var, and STD of flows in packets	Maximum, minimum, median, mean, variance, and standard deviation of the number of packets in each flow
Max, min, med, mean, var, and STD of bytes in flow	Maximum, minimum, median, mean, variance, and standard deviation of the number of bytes in each flow
Max, min, med, mean, var, and STD of bytes in flow duration	Maximum, minimum, median, mean, variance, and standard deviation of the number of bytes in each flow duration
Max, min, med, mean, var, and STD of application layer protocol fields	Maximum, minimum, median, mean, variance, and standard deviation of the number of unique method types, number of unique hosts and URLs and their corresponding numbers of packets and bytes in HTTP requests, and number of unique server names appearing in the TLS client hello and their corresponding numbers of packets and bytes
1 norm	Maximum value of the absolute values for every column in the signals’ matrix Note: This value was determined using the signals provided by the mobile phone’s sensors.
Infinity norm	Maximum value of the absolute values for each row in the signals’ matrix Note: This value was determined using the signals provided by the mobile phone’s sensors.
Fresenius norm	Square root of the sum of squares of all the elements in the signals’ matrix Note: This value was determined using the signals provided by the mobile phone’s sensors.
Max, min, RMS, RMSE, mean, PNum, TNum, SMA, skew, kurt, ATP, and ATT	Maximum value, minimum value, root mean square value, root mean square error, average value of each tap, number of local peaks, number of local troughs, signal magnitude area, asymmetry of the curve, peakedness of the curve, average time to a peak, and average time to a trough of the time-domain signal in each dimension. Note: These values were determined using the signals provided by the mobile phone’s sensors.
Amplitude spectrum	A power spectrum’s square root, which is used to define broadband signals and disturbances. Note: This value was determined using the signals provided by the mobile phone’s sensors.
Power spectrum	Describes the distribution of power into frequency elements that make up the signal. Note: This value was determined using the signals provided by the mobile phone’s sensors.
Formant of the signal spectrum	Commonly used to describe the wide peak or the local maximum in a spectrum

References

1. Clickcease. The State of Click Fraud in SME Advertising. 2022. Available online: <https://www.clickcease.com/blog/wp-content/uploads/2020/09/SME-Click-Fraud-2020.pdf> (accessed on 1 August 2022).
2. Aljabri, M.; Aljameel, S.S.; Mohammad, R.M.A.; Almotiri, S.H.; Mirza, S.; Anis, F.M.; Aboulmour, M.; Alomari, D.M.; Alhamed, D.H.; Altamimi, H.S. Intelligent techniques for detecting network attacks: Review and research directions. *Sensors* **2021**, *21*, 7070. [[CrossRef](#)] [[PubMed](#)]
3. Aljabri, M.; Alahmadi, A.A.; Mohammad, R.M.A.; Aboulmour, M.; Alomari, D.M.; Almotiri, S.H. Classification of Firewall Log Data Using Multiclass Machine Learning Models. *Electronics* **2022**, *11*, 1851. [[CrossRef](#)]
4. Aljabri, M.; Altamimi, H.S.; Albelali, S.A.; Al-Harbi, M.; Alhuraib, H.T.; Alotaibi, N.K.; Alahmadi, A.A.; Alhaidari, F.; Mohammad, R.M.A.; Salah, K. Detecting Malicious URLs Using Machine Learning Techniques: Review and Research Directions. *IEEE Access* **2022**, *10*, 121395–121417. [[CrossRef](#)]
5. Aljabri, M.; Alhaidari, F.; Mohammad, R.M.A.; Mirza, S.; Alhamed, D.H.; Altamimi, H.S.; Chrouf, S.M.B. An Assessment of Lexical, Network, and Content-Based Features for Detecting Malicious URLs Using Machine Learning and Deep Learning Models. *Comput. Intell. Neurosci.* **2022**, *2022*, 3241216. [[CrossRef](#)] [[PubMed](#)]
6. Cheq. The Impact of Invalid Traffic on Marketing. 2022. Available online: https://cheq.ai/wp-content/uploads/2022/01/The-Impact-of-Invalid-Traffic-on-Marketing_REPORT_2-4.pdf (accessed on 3 August 2022).
7. Khan, A.G. Electronic Commerce: A Study on Benefits and Challenges in an Emerging Economy. *Glob. J. Manag. Bus. Res.* **2016**, *16*, 1–5.
8. Pantelimon, F.-V.; Georgescu, T.M.; Posedaru, B.-S. The Impact of Mobile e-Commerce on GDP: A Comparative Analysis between Romania and Germany and how Covid-19 Influences the e-Commerce Activity Worldwide. *Inform. Econ.* **2020**, *24*, 27–41. [[CrossRef](#)]
9. Semerádová, T.; Weinlich, P. *Impacts of Online Advertising on Business Performance*; IGI Global: Hershey, PA, USA, 2019; ISBN 1799816206.
10. Bala, M.; Verma, D. A critical review of digital marketing. *Int. J. Manag. IT Eng.* **2018**, *8*, 321–339.
11. Layton, R.A. Towards a theory of marketing systems. *Eur. J. Mark.* **2011**, *45*, 259–276. [[CrossRef](#)]
12. Zumstein, D.; Kotowski, W. Success Factors of E-Commerce-Drivers of the Conversion Rate and Basket Value. In Proceedings of the 18th International Conference e-Society, Sofia, Bulgaria, 2–4 April 2020; pp. 43–50.
13. Evans, D.S. The online advertising industry: Economics, evolution, and privacy. *J. Econ. Perspect.* **2009**, *23*, 37–60. [[CrossRef](#)]
14. Dean, J. *Big Data, Data Mining, and Machine Learning: Value Creation for Business Leaders and Practitioners*; John Wiley & Sons: Hoboken, NJ, USA, 2014; ISBN 1118618041.
15. Pooranian, Z.; Conti, M.; Haddadi, H.; Tafazolli, R. Online advertising security: Issues, taxonomy, and future directions. *IEEE Commun. Surv. Tutor.* **2021**, *23*, 2494–2524. [[CrossRef](#)]
16. Aksu, H.; Babun, L.; Conti, M.; Tolomei, G.; Uluagac, A.S. Advertising in the IoT Era: Vision and Challenges. *IEEE Commun. Mag.* **2018**, *56*, 138–144. [[CrossRef](#)]
17. Aslam, B.; Karjaluoto, H. Digital advertising around paid spaces, E-advertising industry’s revenue engine: A review and research agenda. *Telemat. Inform.* **2017**, *34*, 1650–1662. [[CrossRef](#)]
18. Ha, L. Online advertising research in advertising journals: A review. *J. Curr. Issues Res. Advert.* **2008**, *30*, 31–48. [[CrossRef](#)]
19. King, A.B. *Website Optimization*; O’Reilly Media, Inc.: Sebastopol, CA, USA, 2008; ISBN 0596515081.
20. Chen, G.; Xie, P.; Dong, J.; Wang, T. Understanding programmatic creative: The role of AI. *J. Advert.* **2019**, *48*, 347–355. [[CrossRef](#)]
21. Chaffey, D.; Smith, P.R. *eMarketing eXcellence: Planning and Optimizing Your Digital Marketing*; Routledge: London, UK, 2013; ISBN 0203082818.
22. Yagci, T. Blended Learning via Mobile Social Media & Implementation of “EDMODO” in Reading Classes. *Adv. Lang. Lit. Stud.* **2015**, *6*, 41–47.
23. Zhu, X.; Tao, H.; Wu, Z.; Cao, J.; Kalish, K.; Kayne, J. *Fraud Prevention in Online Digital Advertising*; Springer: Berlin/Heidelberg, Germany, 2017; ISBN 3319567934.
24. Narayanan, M.; Asur, S.; Nair, A.; Rao, S.; Kaushik, A.; Mehta, D.; Athalye, S.; Malhotra, A.; Almeida, A.; Lalwani, R. Social media and business. *Vikalpa* **2012**, *37*, 69–112. [[CrossRef](#)]
25. Rutz, O.J.; Bucklin, R.E.; Sonnier, G.P. A latent instrumental variables approach to modeling keyword conversion in paid search advertising. *J. Mark. Res.* **2012**, *49*, 306–319. [[CrossRef](#)]
26. Weideman, M.; Kritzinger, W. Parallel search engine optimisation and pay-per-click campaigns: A comparison of cost per acquisition. *S. Afr. J. Inf. Manag.* **2017**, *19*, 1–13.
27. Wood, A.K.; Ravel, A.M. Fool me once: Regulating fake news and other online advertising. *S. Cal. L. Rev.* **2017**, *91*, 1223.
28. Stone-Gross, B.; Stevens, R.; Zarras, A.; Kemmerer, R.; Kruegel, C.; Vigna, G. Understanding fraudulent activities in online ad exchanges. In Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference, Berlin, Germany, 2–4 November 2011; pp. 279–294.
29. Li, Z.; Zhang, K.; Xie, Y.; Yu, F.; Wang, X.F. Knowing your enemy: Understanding and detecting malicious Web advertising. In Proceedings of the 2012 ACM Conference on Computer and Communications Security-CCS, Raleigh, NC, USA, 16–18 October 2012; pp. 674–686. [[CrossRef](#)]

30. Berrar, D. Random forests for the detection of click fraud in online mobile advertising. In Proceedings of the 1st International Workshop on Fraud Detection in Mobile Advertising (FDMA), Singapore, 4 November 2012; pp. 1–10. Available online: http://berrar.com/resources/Berrar_FDMA2012.pdf (accessed on 17 August 2022).
31. Yan, J.H.; Jiang, W.R. Research on information technology with detecting the fraudulent clicks using classification method. *Adv. Mater. Res.* **2014**, *859*, 586–590. [CrossRef]
32. Perera, K.S.; Neupane, B.; Faisal, M.A.; Aung, Z.; Woon, W.L. A novel ensemble learning-based approach for click fraud detection in mobile advertising. *Lect. Notes Comput. Sci.* **2013**, *8284*, 370–382. [CrossRef]
33. Oentaryo, R.J.; Lim, E. Mining Fraudulent Patterns in Online Advertising. In Proceedings of the First International Network on Trust (FINT) Workshop 2013, Singapore, 21–23 November 2013; pp. 21–23.
34. Perera, B.K. A Class Imbalance Learning Approach to Fraud Detection in Online Advertising. *Citeseer* **2013**. Available online: <https://pdfs.semanticscholar.org/24ca/e6b0d1d192e6421905dc65fe8efa2d4343d9.pdf> (accessed on 22 August 2022).
35. Oentaryo, R.; Lim, E.P.; Finegold, M.; Lo, D.; Zhu, F.; Phua, C.; Cheu, E.Y.; Yap, G.E.; Sim, K.; Nguyen, M.N.; et al. Detecting click fraud in online advertising: A data mining approach. *J. Mach. Learn. Res.* **2014**, *15*, 99–140.
36. Phua, C.; Cheu, E.-Y.; Yap, G.-E.; Sim, K.; Nguyen, M.-N. Feature engineering for click fraud detection. In Proceedings of the 2012 Workshop on Fraud Detection in Mobile Advertising (FDMA), Singapore, 4 November 2012; Volume 2010, pp. 1–10. Available online: <http://palanteer.sis.smu.edu.sg/fdma2012/doc/FirstWinner-Starrystarrynight-Paper.pdf%5Cnpapers2://publication/uuid/9290A6CF-A861-4058-99F4-D39706B0619A> (accessed on 27 August 2022).
37. Xu, H.; Liu, D.; Koehl, A.; Wang, H.; Stavrou, A. Click fraud detection on the advertiser side. *Lect. Notes Comput. Sci.* **2014**, *8713*, 419–438. [CrossRef]
38. He, X.; Pan, J.; Jin, O.; Xu, T.; Liu, B.; Xu, T.; Shi, Y.; Atallah, A.; Herbrich, R.; Bowers, S.; et al. Practical lessons from predicting clicks on ads at Facebook. In Proceedings of the Eighth International Workshop on Data Mining for Online Advertising, New York, NY, USA, 24–27 August 2014. [CrossRef]
39. Vani, M.S.; Bhramaramba, R.; Vasumati, D.; Babu, O.Y. TUI based touch-spam detection in mobile applications to increase the security from advertisement networks. *Int. J. Adv. Comput. Commun. Control* **2014**, *2*, 17–22.
40. Beránek, L.; Nýdl, V.; Remeš, R. *Click Stream Data Analysis for Online Fraud Detection in E-Commerce*; Inproforum: České Budějovice, Czech Republic, 2016; pp. 175–180.
41. Berrar, D. Learning from automatically labeled data: Case study on click fraud prediction. *Knowl. Inf. Syst.* **2016**, *46*, 477–490. [CrossRef]
42. Guo, Y.; Shi, J.; Cao, Z.; Kang, C.; Xiong, G.; Li, Z. Machine learning based cloudbot detection using multi-layer traffic statistics. In Proceedings of the 2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS), Zhangjiajie, China, 10–12 August 2019; pp. 2428–2435. [CrossRef]
43. Li, Z.; Jia, W. The Study on Preventing Click Fraud in Internet Advertising. *J. Comput.* **2020**, *31*, 256–265. [CrossRef]
44. Wang, K.; Xu, G.; Wang, C.; He, X. A Hybrid Abnormal Advertising Traffic Detection Method. In Proceedings of the 2017 IEEE International Conference on Big Knowledge (ICBK), Hefei, China, 9–10 August 2017; pp. 236–241. [CrossRef]
45. Wang, J.; Wu, C.; Ji, S.; Gu, Q.; Li, Z. Fraud Detection via Coding Nominal Attributes. In Proceedings of the 2017 2nd International Conference on Multimedia Systems and Signal Processing, Taichung, Taiwan, 13–16 August 2017; pp. 42–45. [CrossRef]
46. MINASTIREANU, E.-A.; MESNITA, G. Light GBM Machine Learning Algorithm to Online Click Fraud Detection. *J. Inf. Assur. Cybersecur.* **2019**, *2019*, 1–12. [CrossRef]
47. Sisodia, D.; Sisodia, D.S. Gradient boosting learning for fraudulent publisher detection in online advertising. *Data Technol. Appl.* **2021**, *55*, 216–232. [CrossRef]
48. Dash, A.; Pal, S. Auto-Detection of Click-Frauds using Machine Learning Auto-Detection of Click-Frauds using Machine Learning. *Int. J. Eng. Sci. Comput.* **2020**, *10*, 27227–27235.
49. Viruthika, B.; Das, S.S.; Manishkumar, E.; Prabhu, D. Detection of advertisement click fraud using machine learning. *Int. J. Adv. Sci. Technol.* **2020**, *29*, 3238–3245. [CrossRef]
50. Srivastava, A. Real-Time Ad Click Fraud Detection. 2020, pp. 1–63. Available online: https://scholarworks.sjsu.edu/etd_projects/916 (accessed on 4 September 2022).
51. Thejas, G.S.; Dheeshjith, S.; Iyengar, S.S.; Sunitha, N.R.; Badrinath, P. A hybrid and effective learning approach for Click Fraud detection. *Mach. Learn. Appl.* **2021**, *3*, 100016. [CrossRef]
52. Gohil, N.P.; Meniya, A.D. *Click Ad Fraud Detection Using XGBoost Gradient Boosting Algorithm*; Springer: Cham, Switzerland, 2021; Volume 1235, ISBN 978-981-15-6647-9.
53. Sisodia, D.; Sisodia, D.S. Feature space transformation of user-clicks and deep transfer learning framework for fraudulent publisher detection in online advertising. *Appl. Soft Comput.* **2022**, *125*, 109142. [CrossRef]
54. Gabryel, M. Data Analysis Algorithm for Click Fraud Recognition. *Commun. Comput. Inf. Sci.* **2018**, *920*, 437–446. [CrossRef]
55. Almahmoud, S.; Hammo, B.; Al-Shboul, B. *Exploring Non-Human Traffic in Online Digital Advertisements: Analysis and Prediction*; Springer: Cham, Switzerland; Volume 3, ISBN 9783030283742.
56. Pan, L.; Mu, S.; Wang, Y. User click fraud detection method based on Top-Rank- k frequent pattern mining. *Int. J. Mod. Phys. B* **2019**, *33*, 1950150. [CrossRef]

57. Sadashiva, T.G. Click Fraud Detection in Online and In-App Advertisements: A Learning Based Approach. Ph.D. Thesis, Florida International University, Miami, FL, USA, 2019.
58. Borgi, M. *Advertisement Click Fraud Detection System: A Survey*; Springer: Singapore, 2021; Volume 10, ISBN 9789811696695.
59. Li, W.; Zhong, Q.; Zhao, Q.; Zhang, H.; Meng, X. Multimodal and Contrastive Learning for Click Fraud Detection. *arXiv* **2021**, arXiv:2105.03567.
60. Aberathne, I.; Walgampaya, C. Real time mobile ad investigator: An effective and novel approach for mobile click fraud detection. *Comput. Inform.* **2021**, *40*, 606–627. [[CrossRef](#)]
61. Mikkili, B.; Sodagudi, S. Advertisement Click Fraud Detection Using Machine Learning Algorithms. *Smart Innov. Syst. Technol.* **2022**, *282*, 353–362. [[CrossRef](#)]
62. Dekou, R.; Savo, S.; Kufeld, S.; Francesca, D.; Kawase, R. Machine Learning Methods for Detecting Fraud in Online Marketplaces. In Proceedings of the 2021 International Workshop on Privacy, Security, and Trust in Computational Intelligence, Gold Coast, QLD, Australia, 1–5 November 2021; Volume 3052.
63. Sisodia, D.; Sisodia, D.S. Quad division prototype selection-based k-nearest neighbor classifier for click fraud detection from highly skewed user click dataset. *Eng. Sci. Technol. Int. J.* **2022**, *28*, 101011. [[CrossRef](#)]
64. Mouawi, R.; Awad, M.; Chehab, A.; El Hajj, I.H.; Kayssi, A. Towards a Machine Learning Approach for Detecting Click Fraud in Mobile Advertising. In Proceedings of the 2018 International Conference on Innovations in Information Technology (IIT), Al Ain, United Arab Emirates, 18–19 November 2018; pp. 88–92. [[CrossRef](#)]
65. Renström, M.; Holmsten, T. Fraud Detection on Unlabeled Data with Unsupervised Machine Learning. 2018, p. 62. Available online: <http://kth.diva-portal.org/smash/get/diva2:1217521/FULLTEXT01.pdf> (accessed on 7 September 2022).
66. Zhang, X.; Liu, X.; Guo, H. A click fraud detection scheme based on cost sensitive BPNN and ABC in mobile advertising. In Proceedings of the 2018 IEEE 4th International Conference on Computer and Communications (ICCC), Chengdu, China, 7–10 December 2018; pp. 1360–1365. [[CrossRef](#)]
67. Thejas, G.S.; Boroojeni, K.G.; Chandna, K.; Bhatia, I.; Iyengar, S.S.; Sunitha, N.R. Deep learning-based model to fight against Ad click fraud. In Proceedings of the 2019 ACM Southeast Conference, Kennesaw, GA, USA, 18–20 April 2019; pp. 176–181. [[CrossRef](#)]
68. Shi, C.; Song, R.; Qi, X.; Song, Y.; Xiao, B.; Lu, S. ClickGuard: Exposing Hidden Click Fraud via Mobile Sensor Side-channel Analysis. In Proceedings of the ICC 2020—2020 IEEE International Conference on Communications (ICC), Dublin, Ireland, 7–11 June 2020. [[CrossRef](#)]
69. Hu, J.; Li, T.; Zhuang, Y.; Huang, S.; Dong, S. GFD: A Weighted Heterogeneous Graph Embedding Based Approach for Fraud Detection in Mobile Advertising. *Secur. Commun. Netw.* **2020**, *2020*, 1–12. [[CrossRef](#)]
70. Liu, X.; Zhang, X.; Miao, Q. *A Click Fraud Detection Scheme Based on Cost-Sensitive CNN and Feature Matrix*; Springer: Singapore, 2020; Volume 1210, ISBN 978-981-15-7529-7.
71. Gabryel, M.; Scherer, M.M.; Sułkowski, Ł.; Damaševičius, R. Decision Making Support System for Managing Advertisers by Ad Fraud Detection. *J. Artif. Intell. Soft Comput. Res.* **2021**, *11*, 331–339. [[CrossRef](#)]
72. Zhu, T.; Meng, Y.; Hu, H.; Zhang, X.; Xue, M.; Zhu, H. Dissecting Click Fraud Autonomy in the Wild. In Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security, Virtual, 15–19 November 2021; pp. 271–286. [[CrossRef](#)]
73. Harsha, C.; Aswale, S.; Pawar, V.N. Advertisement Click Fraud Detection Using Machine Learning Techniques. In Proceedings of the 2021 International Conference on Technological Advancements and Innovations (ICTAI), Tashkent, Uzbekistan, 10–12 November 2021; Volume 10, ISBN 9789811696695.
74. Crussell, J.; Stevens, R.; Chen, H. MAdFraud: Investigating ad fraud in Android applications. In Proceedings of the MobiSys 2014—12th Annual International Conference on Mobile Systems, Applications, and Services, Bretton Woods, NH, USA, 16–19 June 2014; pp. 123–134. [[CrossRef](#)]
75. Iqbal, S.; Zulkernine, M.; Jaafar, F.; Gu, Y. Protecting internet users from becoming victimized attackers of click-fraud. *J. Softw. Evol. Process* **2016**, *30*, e1871. [[CrossRef](#)]
76. Jiang, C.; Zhu, J.; Xu, Q. Dissecting click farming on the Taobao platform in China via PU learning and weighted logistic regression. *Electron. Commer. Res.* **2022**, *22*, 157–176. [[CrossRef](#)]
77. Taneja, M.; Garg, K.; Purwar, A.; Sharma, S. Prediction of click frauds in mobile advertising. In Proceedings of the 2015 Eighth International Conference on Contemporary Computing (IC3), Noida, India, 20–22 August 2015; pp. 162–166. [[CrossRef](#)]
78. Thejas, G.S.; Rameshwar Grag, S.S.; Iyengar, N.R.S. MRFI and ARFI: Hybrids of Filter and Wrapper Feature Selection Approaches. *IEEE Access* **2021**, *9*, 128687–128701. [[CrossRef](#)]
79. Thejas, G.S.; Hariprasad, Y.; Iyengar, S.S.; Sunitha, N.R.; Badrinath, P.; Chennupati, S. An extension of Synthetic Minority Oversampling Technique based on Kalman filter for imbalanced datasets. *Mach. Learn. Appl.* **2022**, *8*, 100267. [[CrossRef](#)]
80. Buzzcity Mobile Advertisement Dataset. 2014. Available online: <https://larc.smu.edu.sg/buzzcity-mobile-advertisement-dataset> (accessed on 14 September 2022).
81. TalkingData. TalkingData AdTracking Fraud Detection Challenge. 2017. Available online: <https://www.kaggle.com/c/talkingdata-adtracking-fraud-detection> (accessed on 14 September 2022).
82. Click-Through Rate Prediction. 2014. Available online: <https://www.kaggle.com/c/avazu-ctr-prediction> (accessed on 12 November 2022).

83. Ramanathan, M. An Ensemble Model for Click through Rate Prediction. 2019. Available online: https://scholarworks.sjsu.edu/etd_projects/697 (accessed on 14 September 2022).
84. Zhou, X.; Yan, P. Avazu Click—Through Rate Prediction Problem description. 2015. Available online: <http://techblog.youdao.com/wp-content/uploads/2015/03/Avazu-CTR-Prediction.pdf> (accessed on 15 September 2022).
85. Larsen, B.S. Synthetic Minority Over-sampling Technique (SMOTE). Retrieved August 2021, 13, 2021.
86. Sadeghpour, S.; Vljajic, N. Click fraud in digital advertising: A comprehensive survey. *Computers* **2021**, *10*, 164. [CrossRef]
87. About JavaScript. Available online: https://developer.mozilla.org/en-US/docs/Web/JavaScript/About_JavaScript (accessed on 16 September 2022).
88. Bot & Ad Fraud Statistics. Available online: <https://www.fraudlogix.com/bot-adfraud-statistics/> (accessed on 17 September 2022).
89. The Global PPC Click Fraud Report 2020–2021. 2021. Available online: <https://www.searchenginejournal.com/the-global-ppc-click-fraud-report-2020-21/391493/> (accessed on 17 September 2022).
90. Google PageRank Checker. Available online: <https://sitechecker.pro/page-rank/> (accessed on 17 September 2022).
91. Cresci, S. A Decade of Social Bot Detection. *Commun. ACM* **2020**, *63*, 72–83. [CrossRef]
92. Sites, E. Ad Fraud Bot Behavior on E-Commerce Sites. Available online: <https://cheq.ai/wp-content/uploads/2021/12/Ad-fraud-bot-behavior-on-e-commerce-sites.pdf> (accessed on 18 September 2022).
93. Qi, S.; AlKulaib, L.; Broniatowski, D.A. *Detecting and Characterizing Bot-like Behavior on Twitter*; Springer: Cham, Switzerland, 2018; Volume 10899, ISBN 9783319933719.
94. Ad Fraud % by Device Type. 2022. Available online: <https://www.fraudlogix.com/stats/ad-fraud-by-device-type/> (accessed on 12 November 2022).
95. Chu, Z.; Gianvecchio, S.; Wang, H. *Bot or Human? A Behavior-Based Online Bot Detection System*; Springer: Cham, Switzerland, 2018; ISBN 9783030048334.
96. Lyu, F.; Tang, X.; Guo, H.; Tang, R.; He, X.; Zhang, R.; Liu, X. Memorize, Factorize, or be Naive: Learning Optimal Feature Interaction Methods for CTR Prediction. In Proceedings of the 2022 IEEE 38th International Conference on Data Engineering (ICDE), Kuala Lumpur, Malaysia, 9–12 May 2022; pp. 1450–1462. [CrossRef]
97. Statista. Advertising Revenue of Google from 2001 to 2021. Available online: <https://www.statista.com/statistics/266249/advertising-revenue-of-google/> (accessed on 23 September 2022).

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.