Egyptian Informatics Journal 24 (2023) 341–350

Contents lists available at ScienceDirect

Egyptian Informatics Journal

journal homepage: www.sciencedirect.com

Click fraud detection for online advertising using machine learning

Malak Aljabri [a,*], Rami Mustafa A. Mohammad [b]

[a] Department of Computer Science, College of Computers and Information Systems, Umm Al-Qura University, Makkah 21955, Saudi Arabia
[b] SAUDI ARAMCO Cybersecurity Chair, Department of Computer Information Systems, College of Computer Science and Information Technology, Imam Abdulrahman Bin Faisal University, P.O. Box 1982, Dammam 31441, Saudi Arabia

ARTICLE INFO

Article history:
Received 19 August 2022
Revised 23 December 2022
Accepted 7 May 2023
Available online 15 May 2023

Keywords:
Click fraud
Pay-per-click
Fraud
Machine learning
Online-advertising
Web-Journy
Bot detection

ABSTRACT

Advertising corporations have moved their focus to online and in-App advertisements in response to the expansion of digital technologies and social media. Online advertising represents the primary revenue source for advertising networks that serve as the middlemen between advertisers and advertisement publishers. The advertising networks pay the publisher of the advertisement based on the number of clicks through to advertisers, following the pay-per-click (PPC) payment scheme. However, there is a growing security issue with this payment approach known as Click Fraud. Click fraud is the illegal process of clicking on pay-per-click advertisements to increase publishers' revenue or deplete advertisers' budgets. Artificial intelligence techniques have been increasingly employed to solve complicated challenges in different research areas, including cybersecurity, to achieve unexpected outcomes. In this paper, several Machine Learning models were constructed to establish whether the user is a human or a bot and conducted a comparative performance analysis using a set of evaluation metrics. We used a real captured dataset detailing Internet users' behavior while navigating websites. We extracted a set of features related to users' behaviors, including the number of webpages viewed during the browsing session, the duration of the browsing journey, and the actions performed. The empirical results revealed that all the considered models obtained good results, where the random forest algorithm surpassed all other algorithms in all evaluation metrics.

1. Introduction

The introduction of the Internet in the 20th century, followed by continuous developments in networking and communication technologies, as well as the worldwide spread of cell phones, provided a noteworthy transition in Internet-based businesses, including the advertising industry, which witnessed unprecedented rapid growth in the form of the digital advertising sector [1]. Online digital advertising expanded incredibly fast, dominating the advertising business sector, achieving revenues expected to total 646 billion US dollars (and 495 billion US dollars for the mobile internet advertising sub-sector) by 2024 [2]. One of the most notable features of the internet is that it permits the online advertising industry to obtain accurate data about users' details and interests, which are shared on social networks or can be tracked via web cookies. This supports the customization and targeting of users via online advertisements in real-time. The most dominant platforms for online advertising are narrative advertising, social media advertising, content, and email marketing, and paying for ads to appear during searches by search engines [3–5]. The worldwide adoption of smart mobile phones and tablets, and recently the internet of things (IoT) devices has altered the initial objective of online advertising, which is now increasingly targeting these devices and integrating advertisements within the contents of Apps, thereby increasing the alignment of ads displayed with the end user interests, behaviors, and purchase history [6–8]. The basic strategy behind online advertising is that publishers (web contents makers and service providers) freely offer their business to end users in the form of a website or App and are paid when publishing the advertisement based on specific agreements with advertisers who purchase dedicated slots on websites or Apps to display content directly to end users who then visit the advertised page for more details about the product or to make a purchase. In addition to the publisher, advertiser, and end-users, a fourth party is involved in the digital advertising system, the advertising network. Advertising networks include internet giants such as Google, Facebook, etc., and act as middlemen between advertisers and publishers. They are responsible for receiving advertisements requests and

[*] Corresponding author.
E-mail addresses: mssjabri@uqu.edu.sa (M. Aljabri), rmmohammad@iau.edu.sa (R.M.A. Mohammad).

https://doi.org/10.1016/j.eij.2023.05.006
1110-8665/

related payments based on agreed billing schemes from advertisers, as well as accepting in-App or website publishing requests from publishers and issuing the payments accordingly. Many advertising network companies exist today, with the primary aim of coordinating with publishers and advertisers in exchange for a share of the publisher's revenue. There are various online advertising billing schemes based on advertising displays, actions, and sales resulting from advertising.

The most commonly used scheme is the pay-per-click (PPC) model, which takes payment based on the number of clicks on advertisements or banners, as each click redirects the user to content belonging to the advertising company [9]. However, this model often opens the door for fraudulent activity, known as "click fraud", which is an illegal practice whereby the advertisements are clicked on with malicious or vindictive intent to increase publishers' revenues or drain the advertisers' marketing budget without any interest in the advertisements. Click fraud affects all forms of paid online advertising, including paid advertisements, links or searches, promotions displayed in Apps, etc. An investigation by the University of Baltimore found that click fraud cost advertisers more than $35 billion in 2020. Moreover, this figure is expected to continue rising significantly [10].

Indeed, click fraud often results from two primary sources:

1. Publishers generate illegal clicks with malicious intent to earn extra money from advertisers without any interest in the clicked advertisements, thereby damaging the trust between the advertisers and the advertisement's publishers, and negatively affecting the growth of the online advertising market. In this case, the main reason behind the significant increase in click fraud activities is that fraudster publishers can quickly create a business generating considerable income with hardly any extra effort.
2. Competitor advertisers, whereby dishonest advertisers perform fraudulent clicks against a competitor to exhaust their advertising budget.

There are various forms of click fraud ranging from simply manual clicking to automated high-volume clicks on pay-per-click advertisements, with different techniques for making fraudulent clicks [10]. Among all these techniques, botnets generating automatic clicks and click-farms where real humans generate high clicks represent the most severe threats. Bots are sophisticated software Apps developed for various malicious and criminal activities such as stealing critical information and personal identities, network-related attacks such as denial of services, and phishing attacks, affecting systems such as malware, viruses, spamming, and advertisements-click fraud [11,12]. In the case of click fraud, botnets represent vast networks of computers infected with malware bot programmed to follow orders without the owner's awareness to automatically click on advertisements imitating legitimate user behavior, resulting in many unrelated clicks from various IP addresses [13]. Advertisers expect advertising networks to catch fraudulent clicks from the advertisement clickstreams as part of their business model.

Nevertheless, with the recent evolution of Artificial Intelligence (AI) technologies and the broad applicability of such tools in cybersecurity, advertising networks' defense systems have evolved to detect click fraud. In response, attackers have been actively improving their click fraud techniques to imitate legitimate users' behaviors and mislead detection systems, thereby increasing the need for more robust, up-to-date detecting techniques. The goal of this research is to build different Machine Learning (ML) models (Decision Tree (DT), Support Vector Machine (SVM), Naïve Bayes (NB), Ripper, PART, Neural Network (NN), and Random Forest (RF)) to recognize whether a visitor to a website is a human or an automated bot user. Specifically, we worked on a real captured dataset related to Internet users' behavior while navigating websites. We pre-processed the dataset and extracted essential features. We then applied these models to the dataset and conducted a comparative performance analysis between the different models. This article is motivated by a necessity to establish if users can be trusted based on information collected about them. A level of trust could then be employed when making automated decisions about whether the user is a human or if additional authentications are needed to confirm that they are not a bot.

The remainder of this paper is structured as follows. Section 2 presents and discusses related studies. In Section 3, the proposed research methodology is demonstrated. The evaluation results are discussed in Section 4. Finally, we conclude and discuss future work in Section 5.

2. Literature review

Exploiting intelligent techniques such as machine learning (ML) and deep learning (DL) to detect different kinds of attacks is becoming an active area of research [14–17]. Detecting click fraud using ML and DL represent the main goal for many research articles in the current literature. Several studies have been conducted to detect click fraud by analyzing real-world click fraud datasets incorporating advertising campaign access details. Individual ML classifiers are applied in most of the related work. Researchers followed this approach to study the effect of building a single ML and/or DL model and perform comparative performance analysis. Mouawi et al. [18] followed this approach to identify fraudulent publishers. They developed different ML and DL models for detecting publishers with a high click fraud rate in the mobile advertising domain and comparing their results' accuracy. Specifically, they applied SVM, K-Nearest Neighbor (KNN), and Artificial Neural Network (ANN). The suggested model integrated click details and user details acquired from advertising networks and advertisers to identify click behavior. They used multiple simulated advertisements traffic datasets with 500,000 advertising requests and 1000 different publishers each. Five different features were extracted based on the information collected from the advertisers, which showed the users' behavior after re-direction to the advertisements, such as the duration of time spent examining the advertisement's contents, and from the advertising network related to the number of clicks. Extracted features per publisher were the percentage of suspicious clicks and the duration of clicks, the total number of clicks, the percentage of unique IP addresses from the total number of clicks, the percentage of App downloads, and changes to the number of advertisement clicks between time iterations. KNN produced the highest accuracy results of 98%. In [9], researchers applied SVM, RF, NB, and DT algorithm on an open-source dataset for fraud detection in mobile advertising (FDMA 2012) [19]. The original dataset was severely imbalanced, necessitating balancing techniques; thus, the positive samples were oversampled, whereas the negative samples were under-sampled. The results demonstrated that the RF algorithms outperformed others and achieved a prediction accuracy of 91%.

Another area of research focused on applying ensemble techniques to minimize various performance concerns, such as noise, bias, and variance. Dash et al. [20] followed this direction when determining the accuracy of a set of single and ensemble ML algorithms for detecting click fraud in an online context. They investigated the patterns of clicks using a dataset with 32,119 clicks collected over a few days. The key goal was to evaluate a user's click journey across their portfolio and identify click-spam from legitimate click streams by determining the IP addresses that generate many clicks but never result in App installation. The results revealed that Gradient Tree Boosting (GTB) algorithm achieved the highest accuracy of 97.2% Minastireanu et al. [21] followed a similar approach, but with a different model. They applied LightGBM, which is a newly developed decision tree-based gradient boosting algorithm that is fast, distributed, and high-performing. The dataset used was open-source and largely imbalanced. The dataset handled 200 million clicks over four days from the TalkingData company, and was provided by Kaggle [22]. For each click record, the dataset includes features conveying information related to the click's IP address, marketing App id, users mobile phones' OS version, users mobile phones' device type, publishers' channel id, and click time. Features engineering and selection aided the researchers with improving the fraud detection performance. In terms of performance and memory consumption, the suggested approach surpassed existing methods with an accuracy of 98%. Similarly, Sisodia et al. [23] proposed a model based on the ensemble classifier GTB to address issues related to identifying the behaviors of publishers from raw user click data. The model was applied to the FDMA 2012 dataset [19], and the results demonstrated that the GTB algorithm outperformed other ML models in terms of average precision, recall, and f-score.

Another research direction investigates various approaches for improving the performance of ML or DL models by developing hybrid or multi-layer models for identifying click fraud. G.S. et al. [24] developed a hybrid DL model consisting of a Semi-supervised Generative Adversarial Network (GAN), Auto Encoder and Neural Network for detecting and discarding fake clicks in real-time. The dataset used was the TalkingData dataset [22]. It consists of approximately 200 million records for real-time mobile advertisement clicks and has seven different features. Researchers handled the imbalance by dividing the datasets into different classes with different characteristics based on the IP address of clicks and mobile App id, then applying under-sampling by taking a similar number of instances from each class. The model achieved good accuracy of 89.7%. Following a similar approach of applying a hybrid model to a larger dataset, Thejas et al. [25] proposed a CFXGB (Cascaded Forest and XGBoost) as a combination of two models: feature transformer and classifier. They used three open-source datasets the TalkingData dataset [22], the Avazu dataset [26], and the Kad dataset [27]. Compared to the other models and simply utilizing cascaded forest as a classifier, this hybrid model excelled at comparative analysis and significantly improved performance. With a focus on peer-to-peer botnets, Khan et al. [28] developed a four-layer adaptive technique based on classifying traffic into P2P botnets or non-P2P. They aimed to reduce the amount of network traffic at the first layer. The network traffic collected was then further classified into non-P2P and P2P by the second layer. Then the number of features was reduced at the third layer to enhance classification accuracy. Finally, the classification was conducted at the fourth layer utilizing a decision tree classification algorithm. The developed model was applied to a combination of two open-source datasets: CTU-13 Dataset [29] of botnet traffic, consisting of thirteen scenarios using different botnet samples, and ISOT [30], which contain both malicious and benign traffic. They applied 10-fold cross-validation and achieved an accuracy of 98.7%.

Other researchers have proposed techniques to identify click fraud and address related issues. One such research is by S. Jain et al. [31], which proposed a method for identifying click fraud by analyzing details related to the identity of website visitors. These details included the user's IP address, the type of operating system, browser details, and time pattern when studying if access typically occurred at specific times. Weights were then assigned to all these features. Consequently, clicks were ranked and compared against specific threshold values, so user access will be suspended if the related clicks are found to be suspicious.

In this research, we followed the first approach and applied a set of ML models to determine whether a visitor to a website is a human or an automated bot user.

3. Methodology

Our experiments were conducted in collaboration with Beacon [32]. Beacon is an internationally recognized software and support company located in London, UK. It provides solutions to support the optimization of internet marketing and substantially improve advertising Return-on-Investment. Beacon has developed a monitoring tool that captures crucial data related to an Internet user's behavior while navigating websites. The information collected is explored and analyzed via intelligent ML approaches. Fig. 1 depicts how a user interacts with the webpages. This research study focuses on those webpages that contain advertisements within them, in the sense that the current article aims to predict whether a user is a human or a bot based on information collected about their behavior while traversing a webpage and directly before clicking on an embedded advertisement(s) using Beacon's software. A web surfer may be a human or a bot browsing webpages using a browser. All web surfers are tracked using a cookie designed by Beacon to track them across various webpages uniquely. A webpage view starts when a surfer requests a particular website from the HTTP server. Many webpage visits happening close to each other triggered by the same surfer are viewed as part of the same journey.

Actions include the activities a surfer performs during a webpage visit; for instance, clicking a button, downloading something,

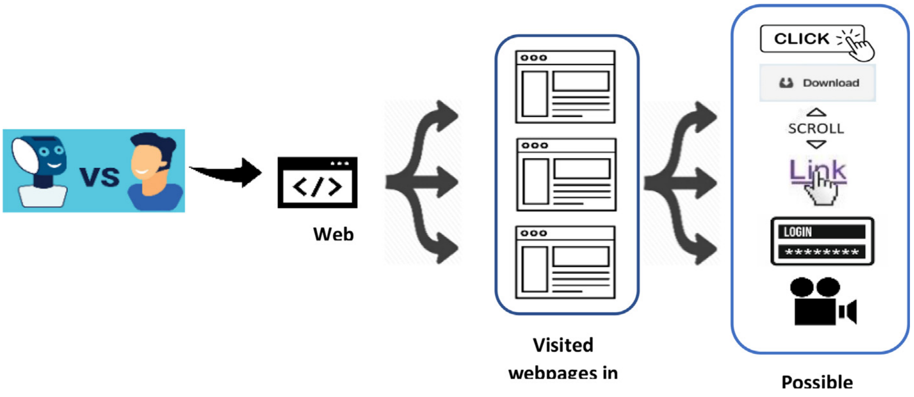


Fig. 1. How a user interacts with webpages.

scrolling up or down, clicking a link, completing a form, watching a video, etc. In this research, we followed the methodology depicted in Fig. 2 and explained in detail the following sub-sections. Specifically, in the pre-processing step, we performed under-sampling to balance the dataset, extracted the related features, and studied their correlation. We then trained our dataset using a 10-fold cross-validation technique and built ML models. Finally, we conducted a comparative performance evaluation using a pre-determined set of metrics.

3.1. Dataset description

The datasets employed in this study were all obtained from Beacon; collected via the Beacon tracking tool. These datasets include information about surfer journeys and are provided as JSON files, which are later reformatted in a form that can be uploaded to WEKA [33] for further experiments. WEKA is a well-known open-source Java-based tool for building intelligent models. WEKA includes several Data Mining and Machine Learning algorithms, which can be directly applied to a dataset and are commonly saved as a CVS file (Comma-Separated Values) or an ARFF file (Attribute-Relation File Format). Three categories of journeys were collected, as shown in Table 1. The fourth column in the table provides a complete description of each dataset.

This research study focuses on the human dataset and the bot dataset. It is noteworthy that all personal information was removed from the datasets before they were sent to the researchers. Moreover, as advised by Beacon, the Google Bot is assumed to be a "Good Bot". This was the main reason for not considering Google Bot in this research.

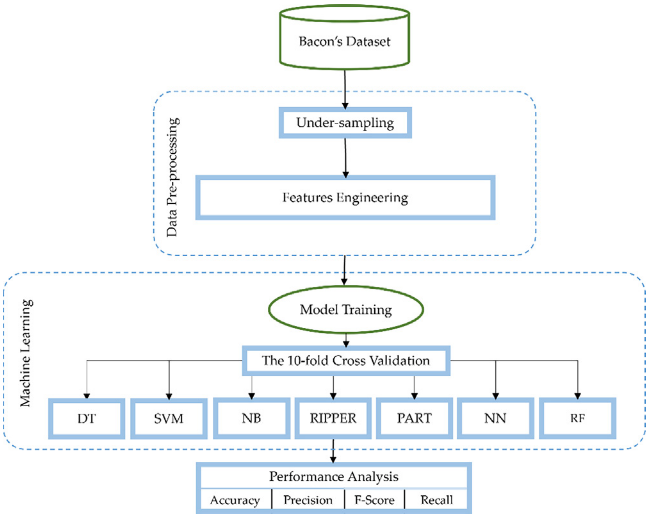


Fig. 2. Research methodology.

3.2. Pre-processing

3.2.1. Under-sampling:

Notably, the provided datasets are unbalanced because the human dataset is almost 28 times the size of the Bot dataset. Unbalanced datasets are considered one of the common challenges when creating any intelligent model. Several methods were proposed in the literature to address this problem. Among others, under-sampling is the most commonly used technique [34]. This research used the under-sampling method by randomly selecting a subset of examples from the human dataset. Hence, 86 random samples were chosen from the human dataset and merged with the bot dataset resulting in an almost balanced training dataset.

3.2.2. Features engineering

This section discusses the set of features extracted from the dataset and how they were pre-processed.

1. User Agent Feature: All HTTP requests create a header field known as the User-Agent to describe the browser they are using. The field usually includes information about the name and version of the browser. Additionally, it provides information about the operating system the browser is running on. Tables 2 and 3 show the top 3 user agents in both the Human and Bot datasets respectively.

However, the information offered by the user-agent feature only cannot be employed to reliably determine whether a surfer is a Bot or a Human. Several pilot experiments using various ML algorithms revealed that information provided by the user-agent delivered a weak contribution when classifying a journey as a human or a bot attempt. Therefore, this feature was not considered when building all intelligent models. Nevertheless, the reason behind listing this feature among the possible features is that it might include information about "Good Bot" such as Google Bot, which has been removed from the datasets as mentioned earlier.

2. Number of Webpages Viewed: This feature is included because bots spend less time on the webpage while visiting more webpages and completing more actions. Therefore, the next three features considered in this study, in addition to the number of webpages viewed, are the time spent on each journey, the number of actions carried out during each journey, and the type of actions completed in the whole journey. Table 4 shows statistics on the number of webpages that surfed in both the Bot and the Human datasets.

The table shows that the average number of pages surfed by Bots is three times higher than the number for Humans. This confirms our assumption that Bots usually visit more webpages in a single journey than Humans. That was established in the results shown in Table 4, which shows the maximum number of webpages viewed by Bots was almost 20 times more than that viewed by Humans, whereas the maximum number of webpages viewed by Bots is 175 webpages, and the Human dataset showed the maximum number of webpages viewed is just nine webpages in a single

Table 1
Categories of journeys collected in this study.

Journey Type	Number of Journeys	Description
Bot	83	It contains traffic manufactured by Beacon's in-house created Bot, so this dataset is confirmed as bot data.
Google Bot	–	This dataset includes information about Google Bot journeys. Google Bot is employed for searching the web. It works by using a website crawling tool created by Google that allows them to scan, find, add, and index new webpages. Google Bot is likely to visit websites submitted to the index occasionally to keep its index updated.
Human	2334	This dataset is taken from Beacon's real-time system. This dataset is assumed to contain information about human journeys. All Google Bot journeys were removed from this dataset. However, as advised by Beacon, although this dataset mainly contains human journeys, this dataset is not guaranteed to exclude all Google Bots data.

Table 2
Top 3 user agents shown in the Bot dataset.

User-Agent
Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/77.0.3835.0 Safari/537.36
Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) HeadlessChrome/77.0.3835.0 Safari/537.36
Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/75.0.3770.142 Safari/537.36

Table 3
Top 3 user agents shown in the Human dataset.

User-Agent
Mozilla/5.0 (Macintosh; Intel Mac OS X 10_13_6) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/75.0.3770.80 Safari/537.36
Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/75.0.3770.100 Safari/537.36
Mozilla/5.0 (iPhone; CPU iPhone OS 12_1_3 like Mac OS X) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/12.0 Mobile/15E148 Safari/604.1

Table 4
Statistics on the number of webpages surfed in both Bot and Human datasets.

Category	Max	Min	Mean	Median	Standard Deviation
Human	9	1	1.872093	1	1.599974
Bot	175	1	6.048193	1	27.01395

journey. However, the results showed that both datasets (Bots and Humans) visited at least one webpage in a single journey. The two datasets also produced the same results when considering the median.

Nevertheless, the standard deviation for Bots is almost 17 times higher than the standard deviation for Humans. This result suggests a more significant variation in the number of webpages Bots visit compared to humans. Therefore, it could be concluded that Bots continue to browse a website as long as the webpages contain at least one advertisement. On the other hand, more advertisements on webpages do not motivate a Human to continue surfing the internet. Figs. 2 and 3 show the percentage of webpages viewed by Bots and Humans, respectively.

The results depicted in Figs. 3 and 4 showed that both Bots and Humans typically visit one webpage in a single journey. For instance, Bots made 26% of extra surfs that involved a single webpage. To elaborate further, 92% of Bot surfs involved a single webpage. Yet, the Human dataset showed that 66% of the web surfs involved a single webpage. Nevertheless, Humans rarely surf more than ten webpages in a single journey, and 0% of surfs involve more than ten webpages. On the other hand, the Bot dataset showed that 6% of the surfs involved more than ten webpages. This confirms the assumption that the more advertisements included in a webpage, the more webpages a Bot will surf.

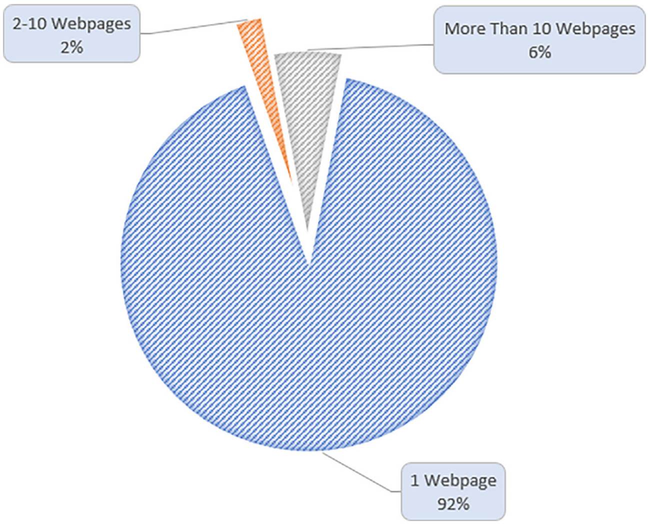


Fig. 3. Percentage of pages visited by Bot.

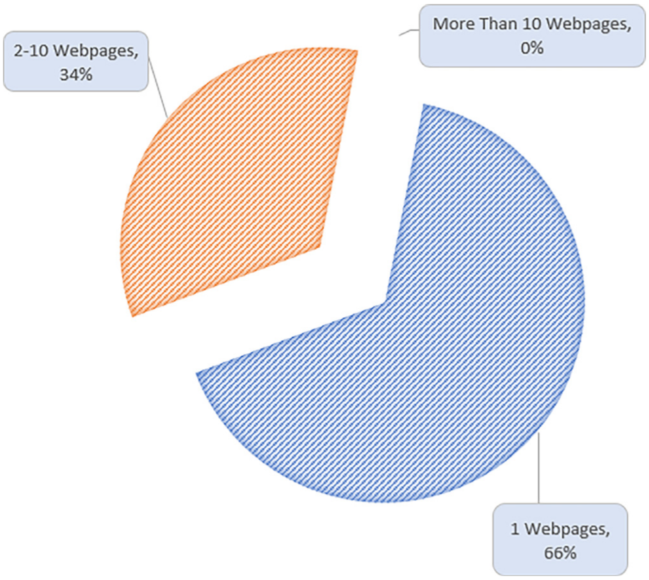


Fig. 4. Percentage of pages visited by Human.

3. Journey Duration: Bots tend to end the journey as soon as they cannot detect any advertisements on a webpage or when no more advertisements on any webpage belong to the same website. Therefore, Bots typically spend less time on a single journey than humans. This can be justified because Humans usually spend more time establishing whether the webpage they have landed on is the one they were looking for. In contrast, a Bot will simply detect if there is any advertisement included on a webpage and click on that only before moving to the next webpage. Fig. 5 shows that there were more Bot journeys of one minute or less compared to Human journeys at a ratio of almost 28%. This figure confirms that Bots spend less time on a journey than humans. To elaborate further, Fig. 5 shows that the only case where the number of Bot journeys was greater than the number of Human journeys was only in a

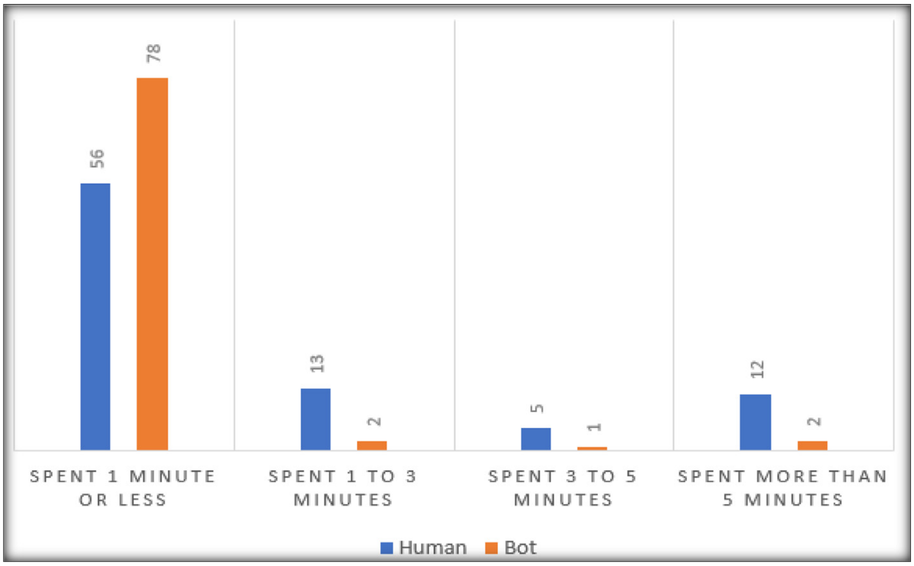


Fig. 5. Time spent in Bot and Human Journeys.

time interval between 0 and 1 min. Meanwhile, at other time intervals, i.e. (one to three minutes), (three to five minutes), and (more than five minutes), the human journeys lasted longer with ratios of 85%, 80%, and 83% respectively.

4. Visit has Ended: This feature indicates that no other webpage views have been tracked within a single session. The cookie that tracks users' activities automatically assigns the value saved in this feature. For instance, a cookie continues monitoring the user's behavior until there is a reason to end the session automatically, and therefore a value of 1 that indicates that the visit has ended is assigned to this feature. Typical reasons for automatically ending a session include, but are not limited to, the following:

- The browser has been closed completely.
- A specific period of time has elapsed without any actions.

These two reasons could be applied to both Bots and Humans. Therefore, in isolation, one might assume that this feature is not a conclusive attribute to determine whether a user is a Human or a Bot. This is a plausible assumption. However, in this research, it is believed that if this feature is combined with other features, it could assist effectively in distinguishing Humans from Bots. Precisely, suppose this feature is combined with the journey duration feature. In that case, this combination could determine whether the user is a Human or a Bot; as if the cookie detected that the session has a start time t_0 and an end time t_1, where the difference between t_1 and t_0 ≈0 s, then this could indicate that the user is a Bot who started the session and as soon as it detected an advertisement clicked on it and closed the browser tab to move to the next webpage. In the Bot dataset, there were 26 cases, constituting 31% of visits, ending after the individual spent almost 0 s on the webpage. Whereas, in the Human dataset, there were only 6 cases where the visit ended after nearly 0 s, which constitutes 7% only.

5. Actions Performed and Number of Actions: Two extra features are also considered in this research: "*The Actions Performed*" and *"The Number of Actions Taken*". A beacon tracking tool can detect a user's actions while surfing a webpage. The tool can detect several actions as shown in Table 5.

These give some clues about what the surfer is doing and whether he is a Human or a Bot. Table 6 provides some statistics obtained from Bot and Human datasets.

Table 5
Actions Detected by Beacon Tool.

No.	Action
1	Click
2	Scroll Down
3	Scroll up
4	Hit a Button
5	Download
6	Hit Top
7	Hit Bottom
8	Click Internal Link
9	Click External Link
10	Video Buffering
11	Mouse Move

Table 6 illustrates that Humans take more actions than Bots. Once they land on a webpage, they start taking further actions to understand whether the website they are visiting is the one they are looking for. However, as soon as a Bot lands on a webpage, it detects if it includes an advertisement and whether the ad requires clicking on to be counted as a visit or if it just requires the surfer to open the webpage to count it as a visit. Hence, typically the Bot will be performing fewer actions. For instance, the average number of actions performed by Humans is 47 times the average number of actions performed by Bots. In addition, the standard deviation of Bot indicates only a few variations in the number of actions taken in each journey. However, the standard deviation for Humans confirms that the number of actions taken varies from one journey to another. Looking at the maximum number of actions a human takes, one can obviously comprehend that it is almost 32 times that Bots take. Digging deeper into the dataset, it has been found that 77% of Bots performed no actions. In contrast, only 26% of Humans visited webpages without completing any actions.

3.3. Model training

Validation involves evaluating whether the mathematical models assessing relations amongst features sufficiently explain the dataset. Normally, an error estimation for the produced classification model is created as soon as the training phase is completed. However, the main concern with such a strategy is that it will not clarify how well the produced classification models might work on the unseen dataset. On the other hand, cross-validation

Table 6
Statistics about Actions Taken by Bots and Humans.

Category	Average	Standard Deviation	Minimum Number of Webpages	Maximum Number of Webpages
Bot	0.28	0.65	0	4
Human	13.06	20.33	0	127

might be one of the most effective techniques helping to recognize this about the produced models. Cross-Validation is a technique for evaluating an ML model by dividing the training dataset into *P* equal partitions, where *P-1* is used for training the model, and the remaining partition is left for validating the produced model. Following several previous studies [35–37], in this study, *P* is set to 10, as shown in Fig. 6. This process is repeated P times, where each partition is used only once for validation and nine other times for training. The error rates produced from each round of validation are averaged to get a true insight into the quality of the produced models. One advantage of cross-validation is that it reduces the chance of creating an overfitted model. Hence, the produced models attain the generalization capability that is considered a good indication of a resilient model.

3.4. Machine learning models

Several ML algorithms were considered in this study, including Decision Tree (DT), Support Vector Machine (SVM), Naïve Bayes (NB), Ripper, PART, Neural Network (NN), and Random Forest (RF). The rationale behind considering these algorithms is that they apply different techniques for building intelligent models. In addition, these algorithms are frequently used for creating supervised classification models where the class value is explicitly included in the training dataset. Moreover, they are commonly used for building classification models in several domains [38]. More information about how these algorithms work is available in [34,33]. Thanks to WEKA for offering several Metalearner functions to facilitate the parameter optimization process to create intelligent models that provide the best performance using cross-validation. This study uses the CVParameterSelection function to optimize the parameters for all the considered intelligent algorithms. The CVParameterSelection is selected because it can be used to optimize the parameters for both the classification algorithms and regression algorithms considered in this study.

3.5. Evaluation metrics

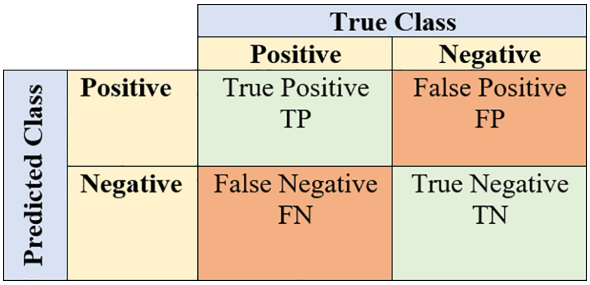
Four evaluation metrics were used in this study to assess the intelligent models' overall performance. The rationale behind the selection of these features is that they are commonly used in the literature [39,40,36].

1. Accuracy rate: This metric describes the ratio of correctly classified examples among the total number of examples holding the same class value.
2. Precision score: Precision describes how well a model can uniquely identify the members of a class versus the remaining possible categories. High precision means the number of false positives for a class is low.
3. Recall score: Recall describes a model's ability to identify all class examples from an unknown dataset. A high recall means the number of false negatives is low for a given class.
4. F1 score: F1 provides an overview of both precision and recall by taking the harmonic average of both. It is sometimes used in place of accuracy. Precision and Recall are inverse metrics and often juxtaposed, meaning that as you optimize for one, the other worsens. Sometimes it is more desirable to optimize for precision and sometimes for recall.

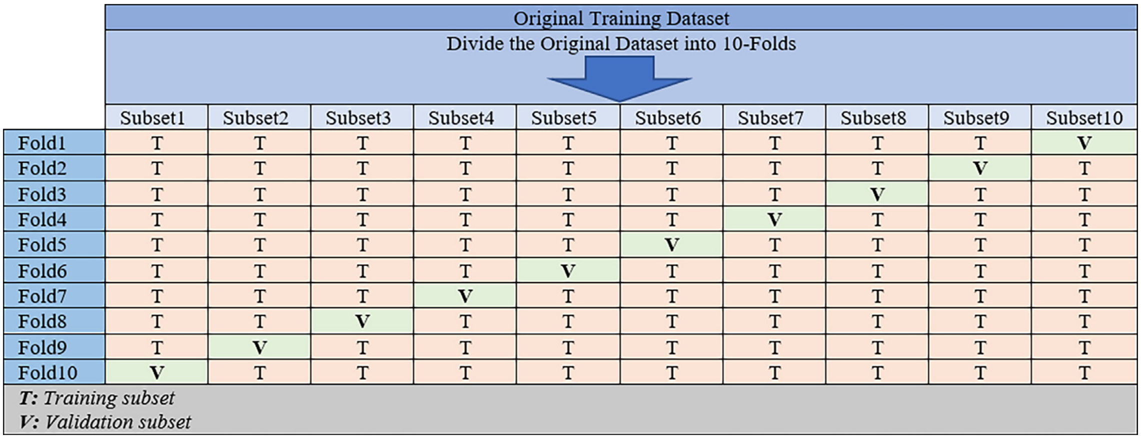
All these metrics can be extracted from the confusion matrix illustrated in Table 7.

TP, FP, FN, and TN have specific meanings in this research. For instance, TP denotes the set of examples correctly classified as Bots. TN represents the set of examples accurately categorized as Human. FP represents the set of examples that are classified as Bots

Table 7
Confusion Matrix.



		True Class	
		Positive	Negative
Predicted Class	Positive	True Positive TP	False Positive FP
	Negative	False Negative FN	True Negative TN



Original Training Dataset

Divide the Original Dataset into 10-Folds

	Subset1	Subset2	Subset3	Subset4	Subset5	Subset6	Subset7	Subset8	Subset9	Subset10
Fold1	T	T	T	T	T	T	T	T	T	V
Fold2	T	T	T	T	T	T	T	T	V	T
Fold3	T	T	T	T	T	T	T	V	T	T
Fold4	T	T	T	T	T	T	V	T	T	T
Fold5	T	T	T	T	T	V	T	T	T	T
Fold6	T	T	T	T	V	T	T	T	T	T
Fold7	T	T	T	V	T	T	T	T	T	T
Fold8	T	T	V	T	T	T	T	T	T	T
Fold9	T	V	T	T	T	T	T	T	T	T
Fold10	V	T	T	T	T	T	T	T	T	T

T: *Training subset*
V: *Validation subset*

Fig. 6. The 10-fold Cross Validation.

while in fact being Human. On the other hand, FN denotes the set of examples categorized as Human while being Bots.

4. Experimental results and discussion

Table 8 and Fig. 7 show the results of the intelligent algorithms adopted in this study.

The results depicted in Table 8 and Fig. 7 show that the intelligent algorithms produced relatively good classification performance, considering that Beacon (the source of the training dataset) stated that the Human dataset could not be 100% confirmed to be a Human dataset, which means that this dataset does not accurately represent Human behaviors. RF outperformed all other considered algorithms in terms of all the evaluation metrics adopted here. For instance, the accuracy rate obtained by the RF surpasses DT, SVM, NB, Ripper, PART, and NN with a ratio of 1.77%, 2.36%,5.32%, 4.14%, 2.36%, and 0.59% respectively. Again, RF produced the highest recall ratio at 87.00%, which means that the RF falsely classified Humans as Bots and vice versa.

Furthermore, it has been found that the RF has a recall ratio of 97.60% for detecting Bots, which means that RF falsely classified Bots as Humans 2.4% of times. On the other hand, RF falsely classified Humans as Bots 29.10% of times. These figures confirmed that spotting Bots is much more accurate than detecting Humans. Again, this proves that the Human dataset still suffers from having rogue Bots designed in a way that makes them difficult to detect. Surprisingly, all the other algorithms include the same situation as RF, in the sense that the recall ratio for detecting Bots is higher than that for detecting Humans. This can be justified because there might be a lot of users who do almost nothing during their journey and use well-known browsers that Bots tend to impersonate leaving very few discriminative data points that can be used to determine if they are Bots or Humans. NB produced the worst classification performance in terms of all evaluation metrics. This can be justified because NB assumes all input features are independent and gives every feature the same level of importance when building the classification model. However, this assumption is not always accurate because some features might be more important than others when building classification models and should garner more attention when creating the classification models.

Nevertheless, the low-performance rates obtained from NB confirm that some input features are highly correlated. This led us to test the correlation among the input features, as shown in Table 9. The results portrayed in Table 9 showed that the Journey Duration feature is highly correlated with the Number of Webpages Viewed and the Actions Performed feature.

This makes sense because the more time the user spends on a journey, the more webpages s/he might visit, and the more webpages s/he visits, the more kind of actions s/he might perform. However, the correlation ratio between Journey Duration and the Number of actions performed is very low, implying that although various actions might be made, each journey does not need to include at least one action. To elaborate, many journeys had no actions at all. In our dataset, it was found that the users (both Humans and Bots) took no actions in almost 57% of journeys, where nearly 43% belong to the Bot dataset and the remaining 14% to the Human dataset. This indicates that Humans are more likely to carry out actions than Bots in a single journey. Overall, intelligent techniques around ML are confirmed to be effective for classifying Human and Bot attempts. Moreover, extra features could be included to improve classification rates. The other set of features below will be considered in the near future to explore their effect on building classification models:

- Double Click: More than one click on the same advertisement within a short period of time.
- Fake Advertisements Click: This feature can be collected by showing a set of fake ads with the expectation that the user will not click on them. However, if the user clicks on them, that could indicate a possible Ad Bot.
- Multiple Advertisements Click: If more than one advertisement is clicked on within a short period of time, this could be a sign of a possible Advertisement Bot.

Table 8
Obtained results from the considered intelligent algorithms.

Algorithm	Accuracy	Precision	Recall	F1-Score
DT	82%	84%	82%	82%
SVM	82%	84%	82%	81%
NB	79%	82%	79%	78%
RIPPER	80%	82%	80%	80%
PART	82%	84%	82%	81%
NN	83%	86%	83%	83%
RF	84%	87%	84%	84%

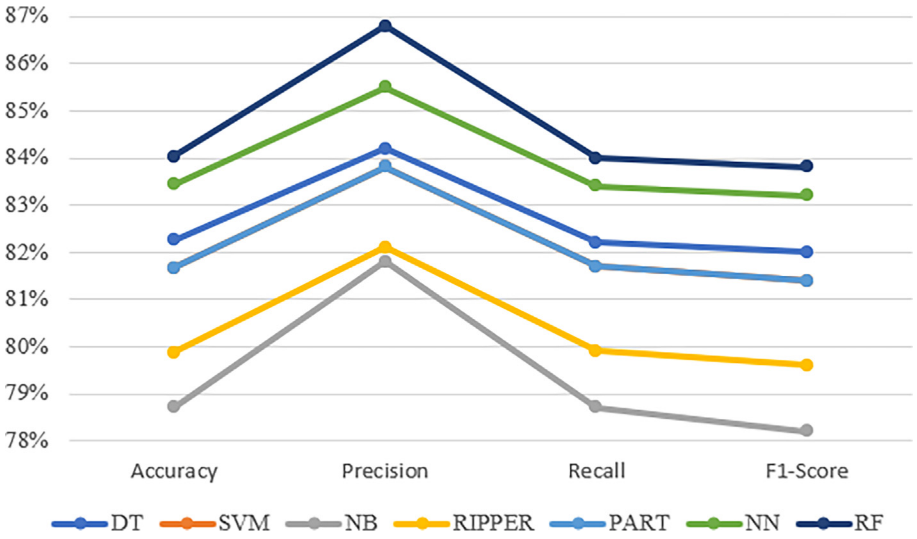


Fig. 7. Obtained results from the considered intelligent algorithms.

Table 9
Correlations among Input Features.

	Journey Duration	Visit has Ended	Number of Webpages Viewed	Actions Performed	Number of Actions
Journey Duration	1				
Visit has Ended	−0.12759	1			
Number of Webpages Viewed	0.465676	−0.01877	1		
Actions Performed	0.636574	−0.1167	0.28805	1	
Number of Actions	−0.08192	0.016949	−0.03835	−0.1396	1

5. Conclusion and future research work

With the rapid increase in the prevalence of electronic advertisements, online scammers have found a golden opportunity to launch their malevolent campaigns via the so-called ad bots. In the pay-per-click payment model, the more clicks an advertisement obtains, the more payments the publisher makes, and the more money the advertiser pays. Hence, scammers tend to profit more by deliberately inflating the number of clicks using ad bots. Ad bots are a significant problem for anyone working in the advertising industry, as they can reduce credibility and advertising campaigns' usefulness and quickly consume budgets. Mitigating ad bots remains an issue of ongoing attention for academic researchers as well as the professional community. In this article, several experiments were conducted in collaboration with Beacon, an internationally recognized software and support company located in London, UK. Beacon has created a tracking tool that records important data about Internet users' habits while navigating websites. The accumulated data is investigated and evaluated using various intelligent techniques. Several conclusive features were extracted from Beacon's raw dataset, including User-Agent, Number of Webpages Viewed, Journey Duration, Visit has Ended, and Action Performed and Number of Actions. The intelligent techniques considered in this article were DT, SVM, NB, Ripper, PART, NN, and RF. The empirical results showed that all the considered algorithms obtained good results, which reflects the effectiveness of the features engineered in this article. The RF surpassed all other considered algorithms. The experimental results showed that classifying bots is much more accurate than detecting humans. The NB obtained the lowest classification performance because it assumes that all features are independent, which is not true since the experimental results showed some highly correlated features. For instance, the Journey Duration correlates to the Number of Webpages Viewed and Actions Performed. In the near future, we will incorporate several other features for building the training dataset, such as Double Click, Fake Advertisements Click, and Multiple Advertisements Click. In addition, it would be interesting to explore several other intelligent approaches, such as Deep Learning and Ensemble techniques.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

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

Availability of data and materials

The dataset used in this study are not publicly available and cannot be provided.

Competing interests

The authors declare that they have no conflicts of interest to report regarding the present study.

Funding

This research was funded by the SAUDI ARAMCO Cybersecurity Chair at Imam Abdulrahman Bin Faisal University.

Authors' contributions

The authors declare equal contributions. The authors read and approved the final manuscript.

References

[1] Zhu X, Tao H, Wu Z, Cao J, Kalish K, Kayne J. Fraud Prevention in Online Digital Advertising. 1st ed. Springer; 2017.

[2] Statista Research Department, "Digital advertising spending worldwide from 2019 to 2024," 2021. [Online]. Available: https://www.statista.com/statistics/237974/online-advertising-spending-worldwide/.

[3] Zenetti G, Bijmolt T, Leeflang P, Klapper D. Search Engine Advertising Effectiveness in a Multimedia Campaign. Int J Electron Commer 2014;18:7–38. doi: https://doi.org/10.2753/JEC1086-4415180301.

[4] Li H, Yang Y. Optimal Keywords Grouping in Sponsored Search Advertising Under Uncertain Environments. Int J Electron Commer 2020;24(1):107–29. doi: https://doi.org/10.1080/10864415.2019.1683704.

[5] Pentina I, Guilloux V, Micu AC. Exploring Social Media Engagement Behaviors in the Context of Luxury Brands. J Advert 2018;47(1):55–69. doi: https://doi.org/10.1080/00913367.2017.1405756.

[6] J. Martins, C. Costa, T. Oliveira, R. Gonçalves, and F. Branco, "How smartphone advertising influences consumers' purchase intention," *J Bus Res*, vol. 94, no. August 2017, pp. 378–387, 2019, 10.1016/j.jbusres.2017.12.047.

[7] Smith KT. Mobile advertising to Digital Natives: preferences on content, style, personalization, and functionality. J Strateg Mark 2019;27(1):67–80. doi: https://doi.org/10.1080/0965254X.2017.1384043.

[8] Aksu H, Babun L, Conti M, Tolomei G, Uluagac AS. Advertising in the IoT Era: Vision and Challenges. IEEE Commun Mag 2018;56(11):138–44. doi: https://doi.org/10.1109/MCOM.2017.1700871.

[9] Li Z, Jia W. The Study on Preventing Click Fraud in Internet Advertising. J Comput 2020;31(3):256–65. doi: https://doi.org/10.3966/199115992020063103020.

[10] N. Gohil and A. D. Meniya, "A Survey on Online Advertising and Click fraud detection," in *2nd National Conference On Research Trends in Information and Communication Technology*, 2020, no. August, [Online]. Available: https://www.researchgate.net/publication/344105501

[11] Silva SSC, Silva RMP, Pinto RCG, Salles RM. Botnets: A survey. Comput Netw 2013;57(2):378–403. doi: https://doi.org/10.1016/j.comnet.2012.07.021.

[12] Shafee A. Botnets and their detection techniques. 2020 Int Symp Networks, Comput Commun ISNCC 2020, 2020,. doi: https://doi.org/10.1109/ISNCC49221.2020.9297307.

[13] J. Williams, "What Are the Types Of Click Fraud?," 2019. [Online]. Available: https://fruition.net/about/blog/types-click-fraud-detect/.

[14] M. Aljabri *et al.*, "Intelligent Techniques for Detecting Network Attacks: Review and Research Directions," *Sensors*, vol. 21, no. 21, 2021, 10.3390/s21217070.

[15] Aljabri M et al. 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. doi: https://doi.org/10.1155/2022/3241216.

[16] M. Aljabri, A. A. Alahmadi, R. M. A. Mohammad, M. Aboulnour, D. M. Alomari, and S. H. Almotiri, "Classification of Firewall Log Data Using Multiclass Machine Learning Models," *Electron.*, vol. 11, no. 12, 2022, 10.3390/electronics11121851.

[17] Aljabri M et al. Detecting Malicious URLs Using Machine Learning Techniques: Review and Research Directions. IEEE Access 2022;10(October):121395–417. doi: https://doi.org/10.1109/access.2022.3222307.

[18] R. Mouawi, M. Awad, A. Chehab, I. H. El Hajj, and A. Kayssi, "Towards a Machine Learning Approach for Detecting Click Fraud in Mobile Advertizing," *Proc. 2018 13th Int. Conf. Innov. Inf. Technol. IIT 2018*, pp. 88–92, 2019, 10.1109/INNOVATIONS.2018.8605973.

[19] R. Oentaryo *et al.*, *Detecting click fraud in online advertising: A data mining approach*, vol. 15. 2014.

[20] A. Dash, S. Pal, "Auto-Detection of Click-Frauds using Machine Learning Auto-Detection of Click-Frauds using Machine Learning," *Indones J Educ Sci*, vol. 10, no. September, 2020.

[21] E. Minastireanu, G. Mesnita, "Light GBM Machine Learning Algorithm to Online Click Fraud Detection Light GBM Machine Learning Algorithm to Online Click Fraud Detection," *J Inf Assur Cybersecurity*, no. April, 2019, 10.5171/2019.263928.

[22] Kaggle.com, "TalkingData AdTracking Fraud Detection Challenge," 2018. https://www.kaggle.com/c/talkingdata-adtracking-fraud-detection (accessed Apr. 08, 2022).

[23] Sisodia D, Sisodia DS. Gradient boosting learning for fraudulent publisher detection in online advertising. Data Technol Appl 2021;55(2):216–32. doi: https://doi.org/10.1108/DTA-04-2020-0093.

[24] G.S. Thejas, K.G. Boroojeni, K. Chandna, I. Bhatia, S.S. Iyengar, N.R. Sunitha, "Deep learning-based model to fight against Ad click fraud," *ACMSE 2019 - Proc. 2019 ACM Southeast Conf.*, no. May, pp. 176–181, 2019, 10.1145/3299815.3314453.

[25] G.S. Thejas, S. Dheeshjith, S.S. Iyengar, N.R. Sunitha, P. Badrinath, A hybrid and effective learning approach for Click Fraud detection, *Mach Learn Appl*, vol. 3, no. November 2020, p. 100016, 2021, 10.1016/j.mlwa.2020.100016.

[26] Kaggle.com, "Click-Through Rate Prediction," 2015. https://www.kaggle.com/c/avazu-ctr-prediction/data (accessed Aug. 04, 2022).

[27] Kaggle.com, "Advertising Dataset," 2017. https://www.kaggle.com/tbyrnes/advertising/data (accessed Apr. 08, 2022).

[28] R. U. Khan, X. Zhang, R. Kumar, A. Sharif, N. A. Golilarz, and M. Alazab, "An adaptive multi-layer botnet detection technique using machine learning classifiers," *Appl Sci*, vol. 9, no. 11, 2019, 10.3390/app9112375.

[29] García S, Grill M, Stiborek J, Zunino A. An empirical comparison of botnet detection methods. Comput Secur Sep. 2014;45:100–23. doi: https://doi.org/10.1016/J.COSE.2014.05.011.

[30] S. Saad *et al.*, "Detecting P2P botnets through network behavior analysis and machine learning," in *2011 Ninth Annual International Conference on Privacy, Security and Trust*, 2011, pp. 174–180, 10.1109/PST.2011.5971980.

[31] S. Jain, F. Jindal, A. Goyal, and S. Mudgal, "Identification of Click Fraud and Review of Existing Detection Algorithms," *Proc. 2nd Int. Conf. Smart Syst. Inven. Technol. ICSSIT 2019*, pp. 894–899, 2019, 10.1109/ICSSIT46314.2019.8987878.

[32] N. Bridges *et al.*, "Beacon." https://www.thisisbeacon.com/ (accessed Apr. 08, 2022).

[33] H. Mark, F. Eibe, H. Geoffrey, P. Bernhard, R. Peter, and W. I. H., "Waikato Environment for Knowledge Analysis," *University of Waikato*, 2011. http://www.cs.waikato.ac.nz/ml/weka/ (accessed Feb. 07, 2022).

[34] Witten I, Frank E, Hall M. Data Mining: Practical Machine Learning Tools and Techniques. 3rd ed. Morgan Kaufmann; 2011.

[35] Mohammad RMA. An Improved Multi-Class Classification Algorithm based on Association Classification Approach and its Application to Spam Emails. IAENG Int J Comput Sci 2020;47(2):187–98.

[36] Mohammad RMA, Alsmadi MK, Almarashdeh I, Alzaqebah M. An improved rule induction based denial of service attacks classification model. Comput Secur 2020;99(1):1–17.

[37] Mohammad RMA, Thabtah F, McCluskey L. Predicting phishing websites based on self-structuring neural network. Neural Comput Appl 2014;25(2):443–58. doi: https://doi.org/10.1007/s00521-013-1490-z.

[38] Mohammad RMA, Alqahtani M. A comparison of machine learning techniques for file system forensics analysis. J Inf Secur Appl 2019;46:53–61. doi: https://doi.org/10.1016/j.jisa.2019.02.009.

[39] Mohammad RMA. A lifelong spam emails classification model. Appl Comput Informatics 2020:1–20.

[40] Mohammad RMA, Alsmadi MK. Intrusion detection using Highest Wins feature selection algorithm. Neural Comput Appl 2021;33(16):9805–16.