ClickGuard: Exposing Hidden Click Fraud via Mobile Sensor Side-channel Analysis

Congcong Shi*†, Rui Song‡, Xinyu Qi‡, Yubo Song‡, Bin Xiao§ and Sanglu Lu*
*State Key Laboratory for Novel Software Technology, Nanjing University
†State Grid Key Laboratory of Information & Network Security, Global Energy Interconnection Research Institute co. Ltd.
‡School of Cyber Science and Engineering, Southeast University
§Department of Computing, The Hong Kong Polytechnic University

***Abstract*—Advertising income depends on the amount of clicks by users of websites and mobile applications. However, the emergence of click fraud greatly reduces the real benefits of the advertisement. Most existing researches focus on detecting click fraud by analyzing properties and patterns of click data streams, but attackers can construct data that looks legitimate by replaying former data streams.**

In this paper, we propose a novel system called ClickGuard to detect click fraud attacks. ClickGuard takes advantage of motion sensor signals from mobile devices, since the pattern of motion signals is completely different under real click events and fraud events. To prevent attackers from bypassing the system by faking the time-domain statistical characteristics of original signals, we introduce the MFCC algorithm in feature extraction phase. MFCC algorithm can extract frequency-domain features of original signals in specific frequency bands which are hardly constructed out of thin air. Classifiers are finally constructed using these features and several machine learning algorithms. Experiments show that ClickGuard can achieve the accuracy of 96.71% in general environment and 84.16% when attackers modify the time-domain statistical characteristics of raw data.

***Index Terms*—Click Fraud, Motion Sensor Signals, Side-channel, MFCC**

I. Introduction

With the development of the Internet, it is possible to share resources and information free of charge. In order to broaden the means of profit and make up for the absence of income by finding a new revenue, content creators' cash in traffic and clicks by showing advertisement to their visitors.

While mobile advertising is gaining popularity as a mean for publishers to monetize their free apps, new types of net attack have emerged. One of the main concerns in the in-app advertising industry is the popular attack known as click fraud, which is the act of clicking on an ad, not because of interest in this ad, but rather to generate illegal revenues for the application publisher [1]. Click fraud happens in pay-per-click ad networks where the ad network charges advertisers for every click on their ads. In a click fraud attack, an automated software clicks on an ad with a malicious intent and advertisers need to pay for those valueless clicks [2].

Thus, click fraud does not bring actual promotional economic benefits to advertisers but inflicts losses on tens of thousands of online advertisers in the order of hundreds of millions of dollars each year [3]. Click fraud has become one of the biggest threats to public advertising, which is responsible for around 30% of click traffic in ad networks [4], costing more than 6 billion dollars to advertisers worldwide [5].

Corresponding author: sanglu@nju.edu.cn

To prevent click fraud attacks, anti-cheating systems are introduced to detect fraud streams. These systems usually use IP detection, cookie authentication, client click verification scripts to detect click fraud and identify malicious attacks [6]. However, these traditional methods are inherently flawed and malicious attackers can bypass these anti-cheating systems. Attackers can launch attacks with the help of dynamic IP pools to deal with IP blacklist. Cookie authentication can also be bypassed since valid cookie can be constructed by malicious scripts. As for client-side detection methods, the users are required to install additional programs on their devices, which is not highly implementable.

Inference techniques which detect click fraud attacks by the statistical characteristics of data streams are introduced by researchers to improve the performance of anti-cheating systems [7], [8]. These systems isolate click fraud attacks by detecting data patterns of ad network click streams. With the help of machine learning algorithms, these methods can effectively identify click fraud attacks and provide protection to advertising networks. However, attackers can counter these defense systems by reuse or fabricate click streams. In this case, this type of defense systems tends to perform poorly.

In this paper, we propose a novel system called ClickGuard, which uses the side-channel of the motion sensors in mobile devices to prevent the click fraud attacks. Unlike systems mentioned above, ClickGuard relies on motion sensor readings of mobile devices, which are harder to replay and fake than click data streams. Sensor readings from accelerometer, gyroscope and orientation sensor are collected as the input. However, most feature extraction algorithms only consider time-domain characteristics, which are easy to be forged. In order to prevent attackers from forging sensor data, we introduce a frequency-domain feature extraction algorithm called MFCC, which can extract features from specific frequency bands of original signals. It is almost impossible for attackers to fabricate a segment of signal whose time-domain and frequency-domain characteristics are similar to those of normal signals. Therefore, ClickGuard can prevent attacks by fake motion sensor signals.

978-1-7281-5089-5/20/$31.00 ©2020 IEEE

The contributions of this paper are summarized as follows:

- We collect motion sensor readings generated when click events take place and build a sensor data set. 450 participants are invited perform click events on mobile devices. More than 4,957,200 motion sensor signals with label are collected from 50,000 keystroke events.
- We design ClickGuard, a novel inference system based on side-channel motion sensor signals. ClickGuard can isolate click events from valid users and malicious programs with the help of classifier trained by motion sensor signals.
- We introduce cepstral analysis technique and MFCC algorithm to improve the performance of ClickGuard. MFCC algorithm are used in feature extraction phase to build robust feature sets and prevent forging sensor signals from attackers.

The rest of the paper is organized as follows. Section II briefly introduces existing researches in this field. Section III reveals the structure and implementation of ClickGuard. Section IV discusses cepstral analysis techniques and the principle of MFCC algorithm. Section V evaluates the performance of ClickGuard in several aspects. Section VI concludes the whole paper.

II. Related Work

In the early days, conventional anti-cheating method adopted at home and abroad is analyzing the data itself, e.g. IP, cookies and click rate threshold [9]–[11]. Rajashree et al. propose a scheme for IP address assignment to smart devices and validation of source IP address in the received IP packets at the gateway device, which show superiority on IP packets detection of the IPv4 and IPv6 protocols [9]. Limitation of this method is that it assumes that the MAC address which is used as device identifier can be duplicated by unauthorized users. Sathitaseelan et al. aimed at the problem of preventing session hijacking, establish a detection system with modified one-time cookies, but the falsifying cookies data is still inevitable [10].

Some researchers employed a technique called threshold-based defense which demonstrated a focus on the volumes of click-fraud based on the bad-listed sources. Pan and his team originate an algorithm based on threshold orthogonal matching pursuit, which possess a better performance than classical match filter [11]. However, the shortcomings of such strategy are obvious that attackers were able to abandon their publisher accounts after receiving reputational hits from ad network defenses, so it is inclined to fail easily [12], [13].

Another promising technique which can be used to combat this form of fraud is to disrupt the value chain consisting of three parties: an advertiser, a publisher, and an ad network. One example of the successful use of this method was the campaign to shut down payment processors used for scareware [14]. In that case, the monetization scheme was the distribution of a fake anti-virus (fake AV), which linked to particular payment processors. But there is no centralized database to build a global picture of the business relationships between actors involved in such fraud case. Based on this, Faou et al. reconstruct a map of the actors by analyzing and aggregating the redirection chains gathered from observation of the network activity of machines infected this malware, describing the structure of this click-fraud ecosystem and identify potential critical targets in it [15].

In recent years, with the maturity of machine learning algorithms, some researchers focused on the use of machine learning algorithms to characterize the pattern differences of abnormal click data streams. Some researchers argue that unusual click-stream traffic is often a simple reuse of legitimate data traffic, so they try to detect click fraud by detecting patterns that repeat themselves in the same click-stream of ads [8], [16]. However, at present many click frauds do not rely on legitimate click data streams, and attackers can also construct data streams similar to the pattern of legitimate click data streams through fitting classifiers. In this case, the performance of the above systems will be greatly degraded.

Based on the observation that there is a big difference in the patterns of data recorded by the motion sensor of mobile devices when legitimate click events and click fraud events occur, we propose to use the motion sensor signals as a side-channel to identify click fraud events. In order to prevent the above-mentioned attacks from forging sensor signals that can fool the classifier by imitating legitimate data, we introduce MFCC algorithm to extract the specific frequency-domain features of motion sensors. Since it is difficult for attackers to construct motion sensor signals whose frequency and time-domain features are all similar to valid motion sensor signals, this algorithm can protect our system from fraud by attackers.

III. Implementation

The implementation of ClickGuard is revealed in this section. This system collects training data from self-designed applications, extracts features from the data and then training the inference models which can be used to detect click fraud attacks.

A. Data Collection

Android and iOS native applications and a web application are designed to collect data under different platforms. When users performing click actions, these applications collect sensor signals from accelerometers, gyroscopes and direction sensors. Then, these signals and corresponding labels are transmitted to the server as classification tags.

Considering that the system will face high-frequency concurrent requirements and complex I / O requests, we use Node.js to build the server. Node.js is a lightweight JavaScript runtime environment based on the event loop mechanism. This mechanism utilizes asynchronous capabilities to prevent I / O blocking and can meet high concurrency requirements.

B. Data Pre-processing

The built-in motion sensors in smartphones are tiny electronic components which sense changes in physical posture and convert it into electrical signals. Noise interference in data collection stage is unavoidable. There are two main sources of

noise, one is the precision of motion sensors of the devices, the other is the vibration generated intentionally or unintentionally during the click actions. Therefore, it is necessary to carry out noise reduction processing on original data to reduce the impact of noise as much as possible. In this stage, the attributes of the data set are normalized by the normalization method such as scaling and parameters shifting, to reduce the impact of noise.

To extract the feature vectors effectively, it is necessary to ensure that the sampling rate of raw data is even and stable. However, it is difficult to achieve this requirement due to two factors. First, since the sensor readings are recorded through the API provided by the mobile operating systems, when devices are under high load conditions, data records corresponding to one click sample will be decreased. Second, different mobile phones use different sensor chips and different operating systems which will make the sensor readings collected from different devices have different sampling rates. To eliminate these differences, cubic spline interpolation is used to preprocess the motion sensor data, so that each click event generate same number of records. Besides, Spline interpolation can use the low-order polynomial splines to achieve smaller interpolation error, which can avoid the Runge phenomenon from high-order polynomial.

C. Time-domain Feature Extraction

Before model training, the features of the original data should be extracted. As mentioned earlier, features are extracted from the time and frequency domains, respectively.

The raw sensor readings from 3 sensors can be divided into 4 signal sequences: Acceleration with Gravity (AccG), Acceleration without Gravity (Acc), Rotational Velocity (Rot), and Orientation (Ori). Each sequence above has 4 axes: x, y, z, and e. Axis e is an additional virtual axis generated from the Euclidean norm of each sequence. The Euclidean distance is extracted to reveal the energy received by devices. Thus, we can define the data generated by a single keystroke event into the form of equation 1.

$$\boldsymbol{F}^{(i)} = [\boldsymbol{Acc}^{(i)} \quad \boldsymbol{AccG}^{(i)} \quad \boldsymbol{Rot}^{(i)} \quad \boldsymbol{Ori}^{(i)}] \tag{1}$$

In time-domain, two types of features are selected for analysis, namely column feature and matrix feature. Column features refer to those characteristics which are extracted from the data of each axis of the sensors. In this paper, column features listed in TABLE I are extracted. These column features reflect the law of change and statistical characteristic of the time-domain signal in each dimension.

Matrix features reflect the cross-correlation coefficient between different axes. Matrix features mainly consist of the following features:

- **1-norm**: the maximum value of the sum of the absolute values of each column in the matrix,
- **Infinity Norm**: the maximum value of the absolute value of each row in the matrix,

TABLE I
COLUMN FEATURES EXTRACTED FROM TIME-DOMAIN

Feature	Description
Max	Maximum value
Min	Minimum value
RMS	Root-mean-square value
RMSE	Root-mean-square error
Mean	Average value of each tap
PNum	Number of local peaks
TNum	Number of local troughs
SMA	Signal magnitude area
Skew	Asymmetry of the curve
Kurt	Peakedness of the curve
ATP	Average time to a peak
ATT	Average time to a trough

- **Frobenius Norm**: square root of the sum of squares of all the elements in the matrix.

In addition, given the data interrelation between the axes of different sensors, the correlation coefficients of *(ori,acc)*, *(ori,accg)*, *(ori,rot)*, *(acc,accg)*, *(acc,rot)*, *(acc,rot)* and *(accg,rot)* are calculated in 4 axes (*x*, *y*, *z* and *e*), totaling 24 characteristics.

D. Frequency-domain Feature Extraction

Subsection III-C describes the time-domain feature extraction process of raw signals. For the frequency domain features, frequency spectrums of the original signals are obtained by FFT, and then cepstrum coefficients are calculated by cepstral analysis and MFCC algorithm.The specific extraction process of frequency-domain features will be introduced in detail in section IV.

E. Model Training

The features obtained in subsection III-C and III-D will be provided as input to the machine learning algorithms. These algorithms use the above input as a basis to construct classifiers. In order to get better classification results, we tried several common classification algorithms:

- Sequential Minimal Optimization
- Bagging
- Random Forest
- Logistic Model Tree
- Convolutional Neural Networks

The basic theories of the classification algorithms are different from each other, which guarantees the heterogeneity of the algorithm, thereby ensuring that the robustness of the entire system can be comprehensively evaluated from multiple angles. And thus we can choose the very classification algorithm which is suitable for our task.

IV. MEL FREQUENCY CEPSTRAL COEFFICIENTS

A. Justification

In order to extract frequency-domain features of the original motion sensor data, some of the techniques of speech signal processing are considered, since there are similarities between

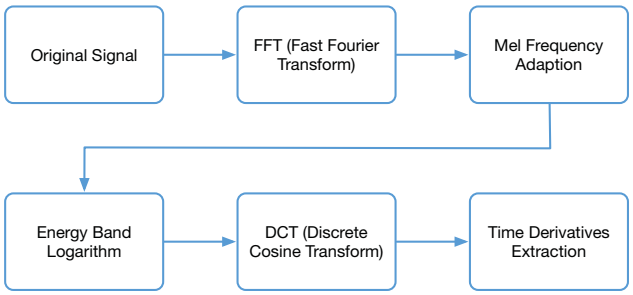

Fig. 1. Diagram of MFCC extraction.

speech signals and motion signals [17]. Speech signals and motion signals are both produced by human body, and both types of signals have correlation in frequency-domain [18]. And the energy of both signals is concentrated in low frequency, which guarantees that key frequency-domain features can be extracted by cepstal analysis. In addition, both speech and motion signals are difficult to reproduce, which means that features extracted from these signals must be robust against the uncertainty of the original data.

Techniques of speech signal processing, speech recognition and blind signal separation have been well developed these days. Techniques like LPC, PLPCC and MFCC can be used for feature extraction. Mel Frequency Cepstral Coefficients (MFCC) is selected for feature extraction. MFCC is a technique which simulates the processing characteristics of the human ears on speech signals to a certain extent. MFCC arranges a set of bandpass filters according to the size of the critical bandwidth from low frequency to high frequency, and then filters the input signals by the filterbank. Since the motion signals are also concentrated in low frequency in a specific way, MFCC can be retrofitted to extract the features of frequency-domain from motion sensor data with a higher robustness and efficiency [19].

B. Cepstral Analysis

To extract features which can represent the characteristics of the original signals, peaks in frequency domains are focused, since peaks denote the dominant components in the signals. In frequency domain analysis, the most common feature is the formant of the signal spectrum. Generally, to extract formants of frequency spectrum, the envelope of the frequency spectrum needs to be extracted first. The envelope contains information such as the position and change process of the formants.

To separate the spectral envelope from the frequency spectrum, a inverse FFT can be performed on original spectrum first. The inverse FFT of the spectrum generates a new signal called pseudo-frequency-domain signal. The pseudo-frequency-domain signal completely separates the envelope and higher order components of the frequency spectrum, and then the envelope can be extracted by a pseudo-frequency-domain low-pass filter.

C. Implementation of MFCC Extraction

Figure 1 shows the implementation of MFCC feature extraction. Fast Fourier Transform is performed to get the frequency spectrum of the original signals. Generally, only amplitude spectrum and power spectrum are focused, and the phase spectrum is discarded since the frame interval is fixed. The frequency-domain data after FFT is in linear scale, which should be transformed to Mel scale for later processing. Mel scale is a non-linear frequency scale based on human sensory. To use MFCC features in motion sensor data analysis, the formula should be adapted to motion signal frequency bands. After that, the energy band logarithm is computed to get the logarithm of the former spectrum. Since the energy in higher frequencies is relatively lower than that in lower frequencies, logarithm can amplify this energy difference.

Next step is to calculate the Discrete Cosine Transform. Before this step, only Fourier transform is used for cepstrum calculation. However, DCT is used in MFCC since the result of DCT is real with no imaginary part. In addition, for motion signals, DCT can further compress the feature sets and get useful data in fewer coefficients. The last step of MFCC extraction is time derivatives extraction. Time derivatives of cepstrum coefficients can express the time continuity of the signals.

V. Evaluation

In this section, the performance of ClickGuard is evaluated by a series of experiments from different aspects.

A. Dataset Acquisition

To evaluate the performance of ClickGuard, motion sensor signals of smart phones should be collected. To collect artificially generated click signals, we invited 450 participants to participate in the experiment. All volunteers are college students and users who use smartphones all year round. During the experiments, volunteers open the applications and click the corresponding keys on the soft keyboard according to the instructions from applications. The experiments collected 5,000 groups of data of numerical sequences corresponding to 50,000 click events. The above data constitutes about 4,957,200 labeled time-domain signals. On the other hand, to generate fraud signals, malicious applications and click malware are created to generate non-organic click signals. Benign and malicious signals are then mixed and shuffled to construct a random sample sequence.

B. Baseline Scenario

In the baseline scenario, only time-domain features introduced in table I are extracted in feature extraction phase and only the base machine learning algorithms introduced in section III are adapted in model training phase. And 10-fold cross-validation is introduced to take full advantage of the original sensor data.

Figure 2 shows the performance of each algorithm in this scenario. The result shows that neural networks obtain the best inference accuracy when only considering time-domain

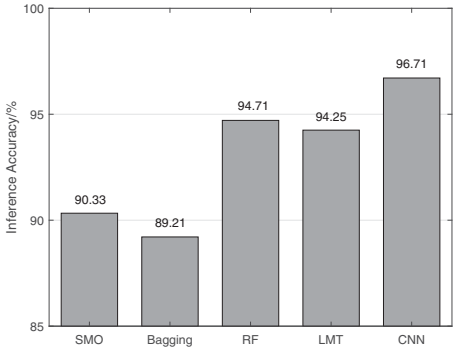

Fig. 2. Accuracies in Baseline Scenario.

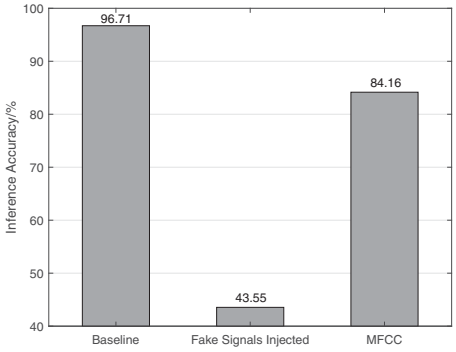

Fig. 3. Accuracies after injecting fake signals.

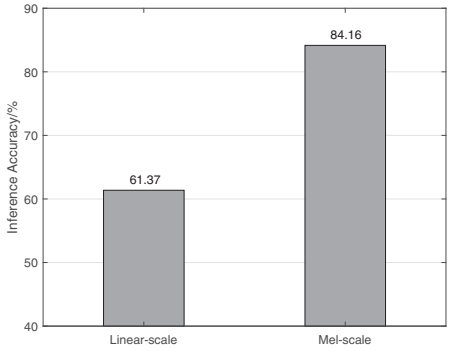

Fig. 4. Accuracies under different wrapping scales.

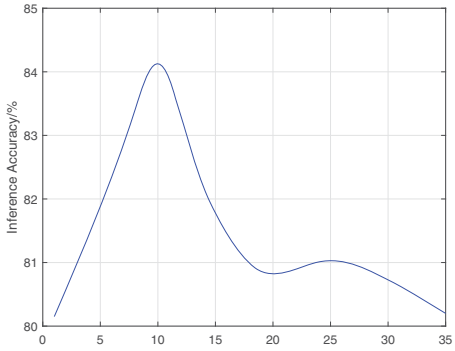

Fig. 5. Accuracies when extracting different numbers of MFCCs.

features. Algorithms based on decision tree, such as Logistic Model Tree and Random Forest, also achieve good performance, which indicates that classification algorithms based on tree structures can better meet the need of click fraud inference. However, the performance of SMO and Bagging is relatively poor, which indicates that SVM-based boost algorithms cannot be well adapted for this task.

C. Data Source Manipulation

Although relatively high inference accuracies can be obtained based on time-domain characteristics, such a model still has security risks. Some studies have pointed out the possibility of deceiving the inference system based on time-domain features by generating fake signals which mimic the time-domain characteristics of normal signals [20]. Although it is difficult to generate fake signals whose waveform and pattern are consistent with normal motion sensor signals, it is easy to adjust the time-domain statistical characteristics of the fake signal according to the feature sets extracted by the inference model.

Based on the this, fake signals are generated based on the time-domain features provided in table I. Only time-domain statistical features in table I are adjusted, so that the generated signals are similar to normal signals in time-domain features. It should be noted that the waveform of fake signals has no similarity with original signals.

Figure 3 shows the inference accuracies after introducing fake signals. The results show that fake signals can effectively cheat the existing inference system based on time-domain characteristics.

D. Mel Frequency Cepstral Coefficients

Mel Frequency Cepstral Coefficients are extracted to take full advantage of frequency-domain characteristics of original signals and can be used to detect fake signals generated based on time-domain features described above.

1) Fake Signals: Figure 3 shows the inference accuracies after introducing MFCC features when fake signals are injected into the data source. The result shows that MFCC can effectively distinguish fake signals from normal motion sensor signals. It is not difficult to manipulate the time-domain characteristics of signals, but it is almost impossible to manipulate both time-domain and frequency-domain characteristics. The frequency-domain characteristics of a signal are expected to match the pattern of a normal signal only if the signal is also generated by the motion sensors. Furthermore, feature set generated by MFCC are more difficult to imitate, because it is difficult to obtain our MFCC extraction algorithm from the model by reverse analysis without knowing our MFCC wrapping frequencies.

2) Wrapping Scales: Figure 4 shows the inference accuracies under different wrapping scales, which indicates that

features in frequency-domain can improve the performance of inference, since the accuracy of linear scale is higher than the accuracy of only time domain features. However, the improvement brought by linear wrapping scale is not significant. The coefficients extracted from Mel-scale, however, improve the inference accuracy to 74.60%, because the energy of the original signals is mainly concentrated in the low frequency part of the signals. Thus, when utilizing Mel-scale which is dense in low-frequency and sparse in high-frequency, the system can better capture the energy which is distributed in low-frequency bands.

3) Number of Coefficients: The number of coefficients extracted from each frame should also be considered, as it is directly related to the size of the feature set, which will greatly affect the efficiency and performance of the classifier. Figure 5 shows the final inference accuracy when different coefficients are extracted from a frame. The results show that as the number of coefficients extracted from a frame increases, the inference accuracy increases first and then decreases. The performance is best when 10 coefficients are extracted from one frame.

VI. Conclusion

In this paper, we construct a system called ClickGuard to prevent click fraud attacks by side-channel of motion sensor signals from mobile devices. ClickGuard collects motion sensor readings during click events from mobile devices and extract features from these signals. Classifiers are constructed by several machine learning algorithms to isolate benign click events and click fraud attacks. To prevent attackers from forging time-domain statistical features, MFCC algorithm is introduced to ClickGuard to extract frequency-domain features. The evaluation of ClickGuard shows that it can distinguish click streams of real users from malicious programs with the accuracy of 96.71%. Even if attackers generate fake signals based on time-domain statistics, ClickGuard can also get the accuracy of 84.16%.

Acknowledgment

This work is supported in part by National Natural Science Foundation of China under Grant Nos. 61872174, 61832008, 61902175, 61872173, 61802169, 61772446; JiangSu Natural Science Foundation under Grant No. BK20190293, BK20180325. This work is partially supported by Collaborative Innovation Center of Novel Software Technology and Industrialization. This work is partially supported by the 2019 Science and Technology Project of SGCC "Research on End-to-End Security Threat Analysis and Precision Protection Technology of Ubiquitous Power Internet of Things". This work is partially supported by the HK PolyU 4-ZZFF and G-YBJV.

References

[1] 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," in *2018 International Conference on Innovations in Information Technology (IIT)*. IEEE, 2018, pp. 88–92.

[2] M. S. Iqbal, M. Zulkernine, F. Jaafar, and Y. Gu, "Protecting internet users from becoming victimized attackers of click-fraud," *Journal of Software: Evolution and Process*, vol. 30, no. 3, p. e1871, 2018.

[3] P. Pearce, V. Dave, C. Grier, K. Levchenko, S. Guha, D. McCoy, V. Paxson, S. Savage, and G. M. Voelker, "Characterizing large-scale click fraud in zeroaccess," in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, 2014, pp. 141–152.

[4] V. Dave, S. Guha, and Y. Zhang, "Measuring and fingerprinting click-spam in ad networks," in *Proceedings of the ACM SIGCOMM 2012 conference on Applications, technologies, architectures, and protocols for computer communication*, 2012, pp. 175–186.

[5] H. Haddadi, "Fighting online click-fraud using bluff ads," *ACM SIGCOMM Computer Communication Review*, vol. 40, no. 2, pp. 21–25, 2010.

[6] C. M. R. Haider, A. Iqbal, A. H. Rahman, and M. S. Rahman, "An ensemble learning based approach for impression fraud detection in mobile advertising," *Journal of Network and Computer Applications*, vol. 112, pp. 126–141, 2018.

[7] M. Kantardzic, C. Walgampaya, B. Wenerstrom, O. Lozitskiy, S. Higgins, and D. King, "Improving click fraud detection by real time data fusion," in *2008 IEEE International Symposium on Signal Processing and Information Technology*. IEEE, 2008, pp. 69–74.

[8] S. Nagaraja and R. Shah, "Clicktok: click fraud detection using traffic analysis," in *Proceedings of the 12th Conference on Security and Privacy in Wireless and Mobile Networks*, 2019, pp. 105–116.

[9] S. Rajashree, K. Soman, and P. G. Shah, "Security with ip address assignment and spoofing for smart iot devices," in *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. IEEE, 2018, pp. 1914–1918.

[10] A. M. Sathiyaseelan, V. Joseph, and A. Srinivasaraghavan, "A proposed system for preventing session hijacking with modified one-time cookies," in *2017 International Conference on Big Data Analytics and Computational Intelligence (ICBDAC)*. IEEE, 2017, pp. 451–454.

[11] J. Pan and J. Tang, "Sparse targets detection based on threshold orthogonal matching pursuit algorithm," in *2016 IEEE Sixth International Conference on Communications and Electronics (ICCE)*. IEEE, 2016, pp. 258–261.

[12] V. Dave, S. Guha, and Y. Zhang, "Viceroi: Catching click-spam in search ad networks," in *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, 2013, pp. 765–776.

[13] Q. Zhang, T. Ristenpart, S. Savage, and G. M. Voelker, "Got traffic? an evaluation of click traffic providers," in *Proceedings of the 2011 Joint WICOW/AIRWeb Workshop on Web Quality*, 2011, pp. 19–26.

[14] M. Hillick, "Scareware traversing the world via a web app exploit," *Bethesda (MD): SANS Institute InfoSec Reading Room*, 2010.

[15] M. Faou, A. Lemay, D. Décary-Hétu, J. Calvet, F. Labrèche, M. Jean, B. Dupont, and J. M. Fernande, "Follow the traffic: Stopping click fraud by disrupting the value chain," in *2016 14th Annual Conference on Privacy, Security and Trust (PST)*. IEEE, 2016, pp. 464–476.

[16] X. Zhang, X. Liu, and H. Guo, "A click fraud detection scheme based on cost sensitive bpnn and abc in mobile advertising," in *2018 IEEE 4th International Conference on Computer and Communications (ICCC)*. IEEE, 2018, pp. 1360–1365.

[17] L. Liu, M. Popescu, M. Skubic, M. Rantz, T. Yardibi, and P. Cuddihy, "Automatic fall detection based on doppler radar motion signature," in *2011 5th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth) and Workshops*. IEEE, 2011, pp. 222–225.

[18] A. M. Khan, Y.-K. Lee, S.-Y. Lee, and T.-S. Kim, "Human activity recognition via an accelerometer-enabled-smartphone using kernel discriminant analysis," in *2010 5th international conference on future information technology*. IEEE, 2010, pp. 1–6.

[19] R. San-Segundo, J. M. Montero, R. Barra-Chicote, F. Fernández, and J. M. Pardo, "Feature extraction from smartphone inertial signals for human activity segmentation," *Signal Processing*, vol. 120, pp. 359–372, 2016.

[20] J. Pawlick, E. Colbert, and Q. Zhu, "A game-theoretic taxonomy and survey of defensive deception for cybersecurity and privacy," *ACM Computing Surveys (CSUR)*, vol. 52, no. 4, pp. 1–28, 2019.