
EFFICIENT AND DEPLOYABLE CLICK FRAUD DETECTION FOR MOBILE APPLICATIONS

A. DURGA DEVI MADAM¹, Jami Harisha Jyothi²,

¹Assistant professor, MCA DEPT, Dantuluri Narayana Raju College, **Bhimavaram, Andharapradesh**

Email:- adurgadevi760@gmail.com

²PG Student of MCA, Dantuluri Narayana Raju College, **Bhimavaram, Andharapradesh**

Email:- hariharishajami@gmail.com

ABSTRACT

Mobile advertising plays a vital role in the mobile app ecosystem. A major threat to the sustainability of this ecosystem is click fraud, i.e., ad clicks performed by malicious code or automatic bot problems. Existing click fraud detection approaches focus on analyzing the ad requests at the server side. However, such approaches may suffer from high false negatives since the detection can be easily circumvented, e.g., when the clicks are behind proxies or globally distributed. In this paper, we present Ad Sherlock, an efficient and deployable click fraud detection approach at the client side (inside the application) for mobile apps. Ad Sherlock splits the computation-intensive operations of click request identification into an offline procedure and an online procedure. In the offline procedure, Ad Sherlock generates both exact patterns and probabilistic patterns based on URL (Uniform Resource Locator) tokenization. These patterns are used in the online procedure for click request identification and further used for click fraud detection together with an ad request tree model. We implement a prototype of Ad Sherlock and evaluate its performance using real apps. The online detector is injected into the app executable archive through binary instrumentation. Results show that Ad Sherlock achieves higher click fraud detection accuracy compared with state of the art, with negligible runtime overhead.

1 INTRODUCTION

Mobile advertising plays a vital role in the mobile app ecosystem. A recent report shows that mobile advertising expenditure worldwide is projected to reach \$247.4 billion in 2020 [1]. To embed ads in an app, the app developer typically includes ad libraries provided by a third-party mobile ad provider such as AdMob [2]. When a mobile user is using the app, the embedded ad library fetches ad content from the network and displays ads to the user. The most common charging model is PPC (Pay-Per-Click) [3], where the developer and the ad provider get paid from the advertiser when a user clicks on the ad.

2 RELEATED WORK

A literature survey on "Efficient And Deployable Click Fraud Detection for Mobile Applications" would typically involve reviewing existing research and publications related to fraud detection in mobile applications, especially focusing on click fraud. Here's a structured approach to conducting such a survey:

2.1 Introduction to Click Fraud Detection

- Define click fraud in the context of mobile applications.
- Explain its significance and impact on mobile advertising.
- Discuss the challenges specific to detecting click fraud in mobile apps (e.g., device diversity, network variability, etc.).

3 IMPLEMENTATION STUDY

Existing System:

since existing machine-learning algorithms used by server-side approaches are not suitable for the client side. Second, the click fraud detection should be able to execute under practical user scenarios, instead of a controlled environment dedicated to fraud detection. In MAdFraud [5], a controlled environment (i.e., only one app is running and the HTTP requests are collected for offline analysis) is used to measure the ad fraud behaviour of a vast number of apps. However, in our case, the click fraud detection should happen inside the mobile client without outside support, i.e., be deployable in real-world scenarios.

Proposed System

We propose two pattern classes: exact patterns and probabilistic patterns. Both of them are built from invariant substrings in the HTTP header. We refer to these substrings as tokens. Exact patterns consist of a set of sequential tokens and match an HTTP request if and only if the request contains all tokens in the set with the same ordering. Probabilistic patterns consist of a set of tokens, each of which is associated with an ad score, and a non-ad score. We describe the details of pattern generation in the following sections.

Advantages of Proposed System

1. AdSherlock produces both accurate examples and probabilistic examples in light of URL (Uniform Resource Locator) tokenization.

2. AdSherlock instruments the internet based misrepresentation identifier into the application pairs which are then delivered by the application store.

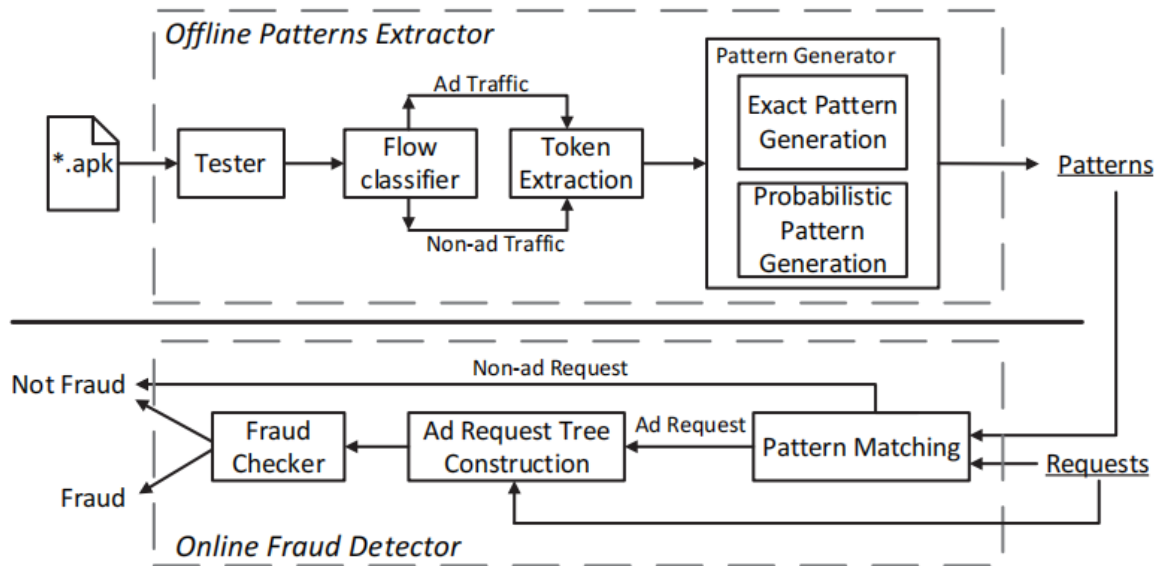


Fig1: SYSTEM ARCHITECTURE

IMPLEMENTATION

We have implemented a prototype of AdSherlock. The offline pattern extractor is implemented in Python and runs on Ubuntu 14.04 equipped with a 3.30GHz quad-core CPU and 12GB memory. The online fraud detector is implemented within a simple Android application, targeting Android API level 19 and running on a Nexus 5 device equipped with 2.26 GHz quad-core and 2GB memory. The online fraud detector is injected into the application archive through binary instrumentation. It intercepts the network traffic at runtime and logs the user touchscreen input events into the buffer. The network traffic is then fed into the pattern matching part to identify ad requests. The touchscreen input events, i.e Motion events are used by the fraud checker to detect click frauds.

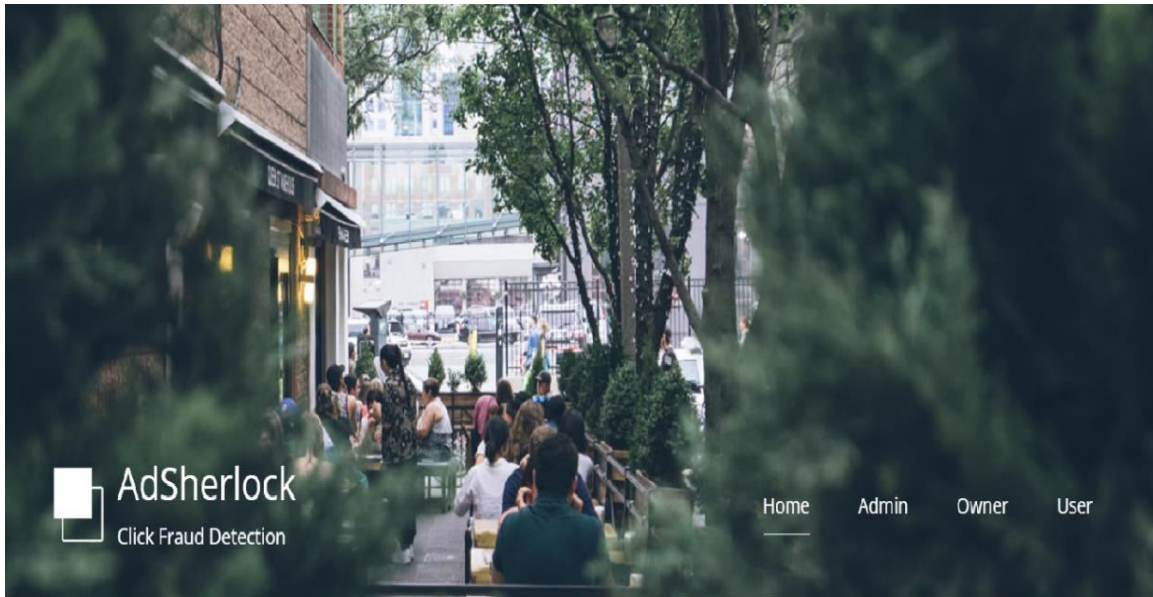
4.1 MODULES

Binary instrumentation: The app executable archive for a given mobile application contains, amongst other things, the application binary (classes.dex) and a metadata file. In order to inject the online fraud detector into the application, we propose to decompile the application and inject a small patch into the bytecode before repackaging the application. First, the original application is disassembled by using the baksmali tool to obtain a human-readable Smali bytecode from the dex file. Then, the Smali code is analyzed to find APIs calling the HTTP library.

5 RESULTS AND DISCUSSION

SCREENSHOTS

Home Page



Owner login page



Owner Login

UserName

Password

[Register](#)

Owner registration page



Registration Here

Name
Email
Mobile
Address
UserName
Password
User Type

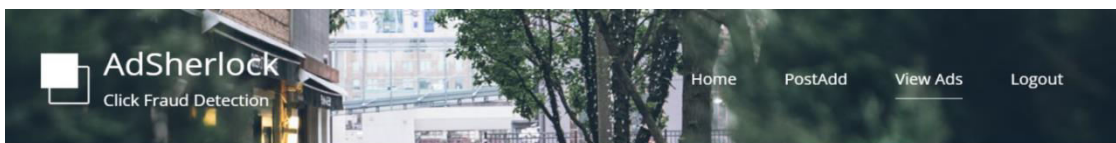
Post add page





POST ADD

Add Name
Add Description
Date No file chosen
Add Image

All posted Adds page



ALL POSTED ADS

Post Name	Description	Date	Post Image
google	google is giving 5000 dollars	2024-06-15	
pics art	poster presentation	2024-06-15	

User login page



User Login

UserName
Password
 [Register](#)

user registration page



Registration Here

Name
Email
Mobile
Address
UserName
Password
User Type

User home page



google



pics art



View profile page



View Profile

Name	Email	Mobile	Address
raj	aa@aa.com	07093153158	D.no:8-39,Maddu vari street.

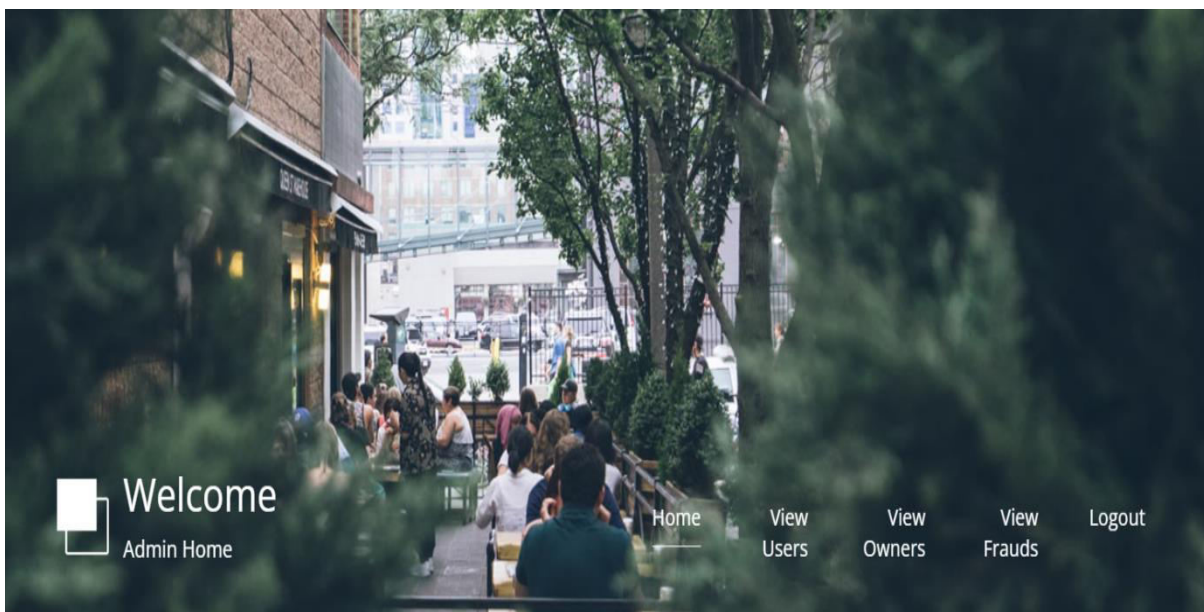
Admin login page



Admin Login

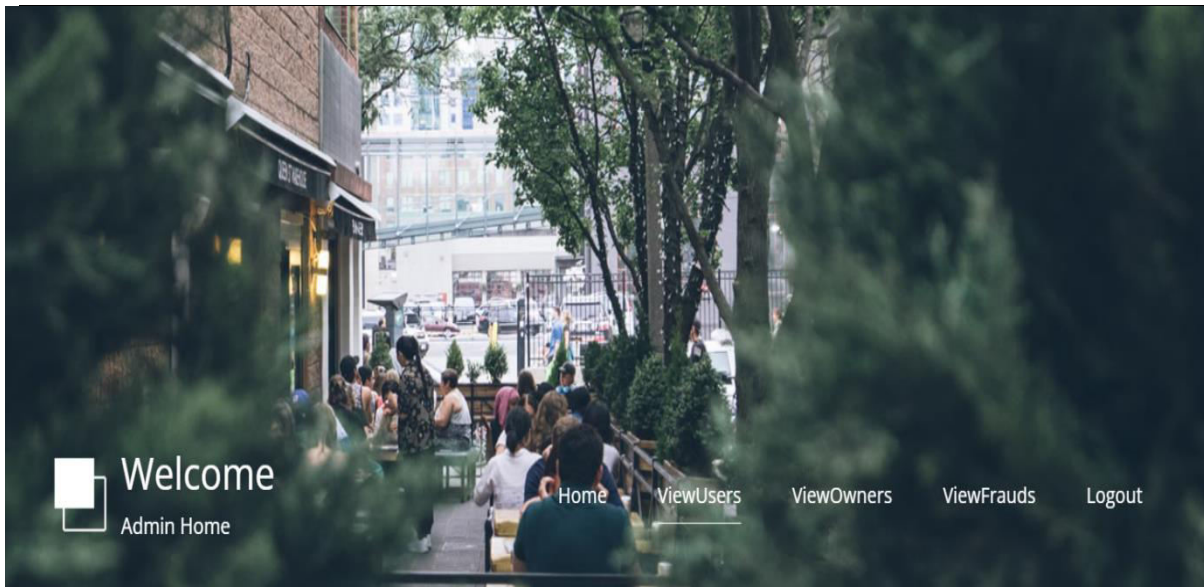
UserName
Password

Admin home page



Welcome to Admin Home

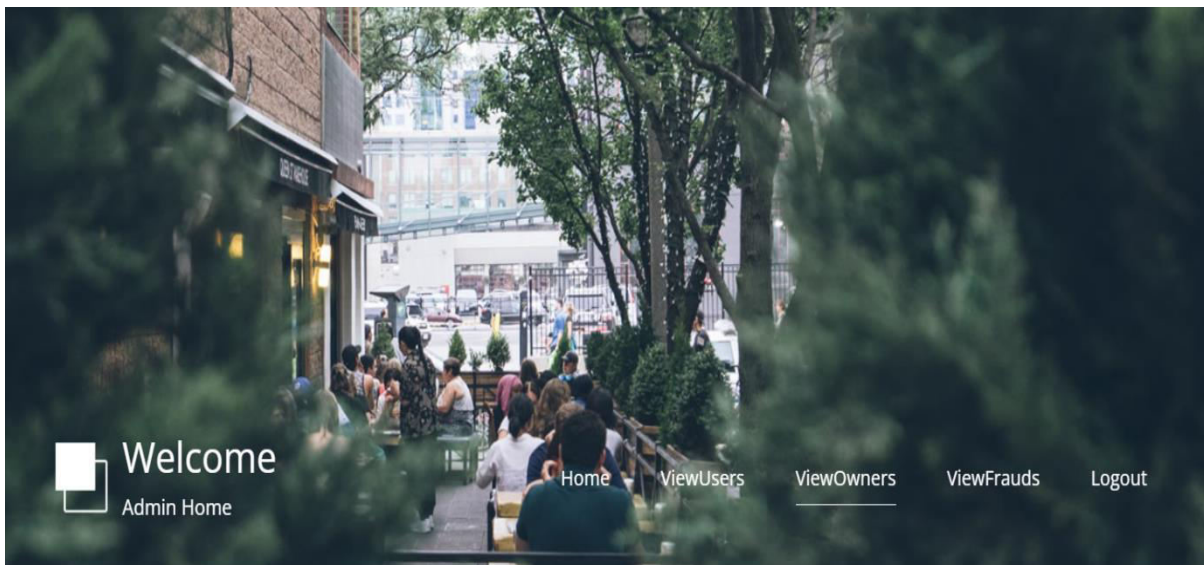
View users Page



View All Users

Name	Email	Mobile	Address
raj	aa@aa.com	07093153158	D.no:8-39,Maddu vari street.

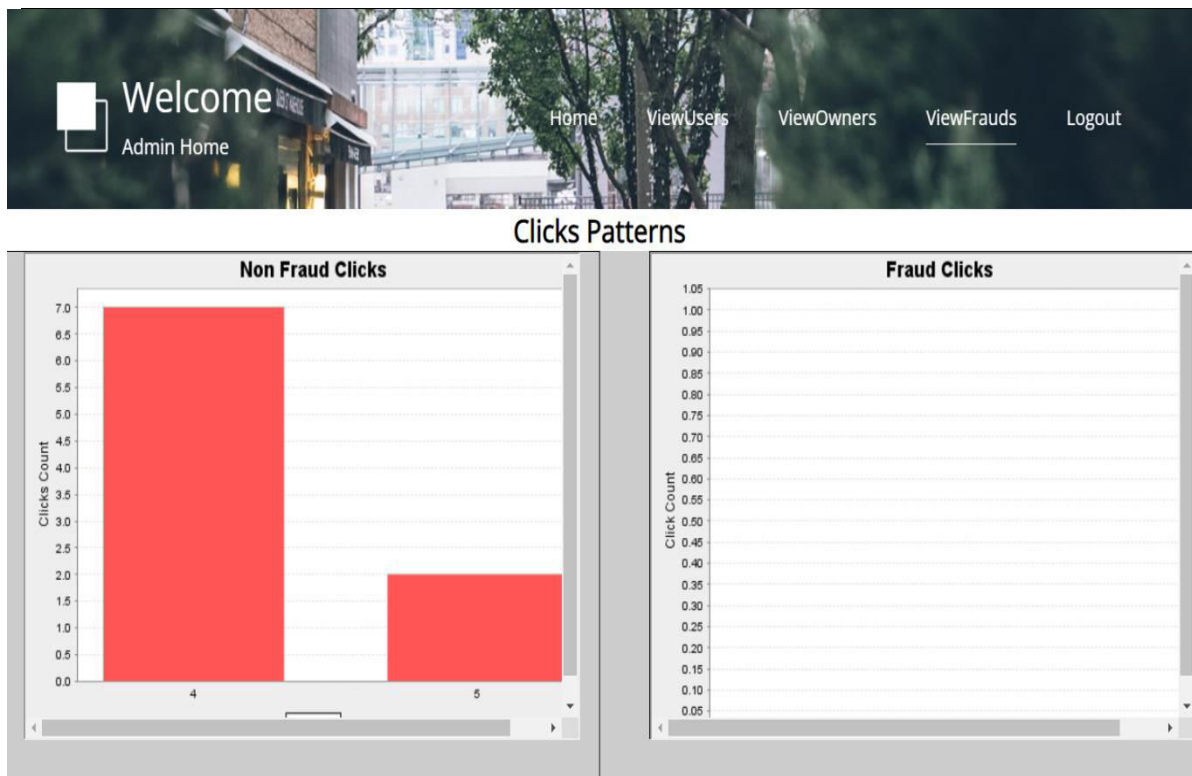
View owners Page



View All Owner

Name	Email	Mobile	Address
dinesh	aa@aa.com	9347225321	vskp

View fraud Page



6. CONCLUSION AND FUTURE WORK

CONCLUSION

AdSherlock is an efficient and deployable click fraud detection approach for mobile apps at the client side. As a client-side approach, AdSherlock is orthogonal to existing server-side approaches. It splits the computation intensive operations of click request identification into an offline process and an online process. In the offline process, AdSherlock generates both exact patterns and probabilistic patterns based on url tokenization. These patterns are used in the online process for click request identification, and further used for click fraud detection together with an ad request tree model. Evaluation shows that AdSherlock achieves high click fraud detection accuracy with a negligible runtime overhead. In the future, we plan to combine static analysis with traffic analysis to improve the accuracy of ad request identification and explore attacks designed to evade AdSherlock.

7. REFERENCES

- [1] "Mobile advertising spending worldwide." [Online]. Available: <https://www.statista.com/statistics/280640/mobile-advertisingspending-worldwide/>

-
- [2] “Google admob.” [Online]. Available: <https://apps.admob.com/>
- [3] M. Mahdian and K. Tomak, “Pay-per-action model for online advertising,” in Proc. of ACM ADKDD, 2007.
- [4] G. Cho, J. Cho, Y. Song, and H. Kim, “An empirical study of click fraud in mobile advertising networks,” in Proc. of ACM ARES, 2015.
- [5] J. Crussell, R. Stevens, and H. Chen, “Madfraud: Investigating ad fraud in android applications,” in Proc. of ACM MobySys, 2014.
- [6] R. Oentaryo, E.-P. Lim, M. Finegold, D. Lo, F. Zhu, C. Phua, E.-Y. Cheu, G.-E. Yap, K. Sim, M. N. Nguyen, K. Perera, B. Neupane, M. Faisal, Z. Aung, W. L. Woon, W. Chen, D. Patel, and D. Berrar, “Detecting click fraud in online advertising: A data mining approach,” *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 99–140, 2014.
- [7] B. Kitts, Y. J. Zhang, G. Wu, W. Brandi, J. Beasley, K. Morrill, J. Etedgui, S. Siddhartha, H. Yuan, F. Gao, P. Azo, and R. Mahato, *Click Fraud Detection: Adversarial Pattern Recognition over 5 Years at Microsoft*. Cham: Springer International Publishing, 2015, pp. 181–201.
- [8] A. Metwally, D. Agrawal, and A. El Abbadi, “Detectives: detecting coalition hit inflation attacks in advertising networks streams,” in Proc. of ACM WWW, 2007.
- [9] A. Metwally, D. Agrawal, A. El Abbad, and Q. Zheng, “On hit inflation techniques and detection in streams of web advertising networks,” in Proc. of IEEE ICDCS, 2007.
- [10] F. Yu, Y. Xie, and Q. Ke, “Sbotminer: large scale search bot detection,” in Proc. of ACM WSDM, 2010.