ADFRAUD : CLICK FRAUD DETECTION FOR MOBILE APPLICATION

Kruthi.P[1], Dr.K.F.Bharati[2]

[1]PG-Scholar, Department of CSE (Artificial Intelligence), JNTUA College of Engineering (Autonomous) Ananthapuramu, India.

[2]Associate Professor, Department of CSE, JNTUA College of Engineering (Autonomous) Ananthapuramu, India.

kruthikalpa@gmail.com[1], kfbharati.cse@jntua.ac.in[2]

Abstract

Mobile advertising is a crucial component in the mobile app ecosystem, with click fraud being a significant threat to its viability. This fraudulent activity, which includes ad clicks from malicious code or automated bots, undermines the ecosystem's sustainability. Most current click fraud detection methods concentrate on examining ad requests from the server's perspective. However, these methods can be easily bypassed, leading to a high rate of false negatives. Existing client-side (within the app) fraud detection divides tasks into two procedures: offline click request identification and an online process. In the offline stage, exact and probabilistic patterns are derived from URL tokenization, which then aid online click request identification and subsequent click fraud detection. This online detector is integrated into the app's binary archive using binary instrumentation. A notable shortcoming of this method is its inefficiency in detecting click fraud and the latency resulting from the dual offline and online modes, negatively impacting user experience. Our proposed system offers an improved solution by introducing an efficient click fraud detection method on the server side. This new approach includes a pattern generation technique that accurately discerns between genuine and fraudulent ad requests. Implementing this server-side approach allows real-time fraud detection, covering aspects like fraud reviews, app downloads, and user comments. Consequently, latency issues are significantly reduced.

Keywords: Mobile advertising, Click fraud, Mobile app ecosystem, Malicious code, Automated bots, Server-side detection, False negatives, Client-side detection, Offline procedure, Online procedure, URL tokenization, Exact patterns, Probabilistic patterns, Binary instrumentation, Latency, User experience, Pattern generation, Real-time fraud detection.

I. INTRODUCTION

In today's digital age, mobile advertising has rapidly emerged as a linchpin in the vast world of the mobile application ecosystem. As smartphones and apps become ubiquitous, businesses capitalize on this trend by placing ads within mobile applications, reaching users more directly and personally than ever before. However, with this surge in mobile advertising's importance, the ecosystem also faces significant challenges, among which click fraud has become particularly notorious.

Click fraud refers to the deceptive practice where ads are clicked without any genuine interest from the user. This could be due to automated bots, malicious software, or other illegitimate means. These fraudulent clicks not only deceive advertisers into paying for non-productive ad

Submitted: 28/07/2023 **Accepted**: 09/08/2023

views but also distort the overall integrity of advertising metrics. Furthermore, it poses a financial drain on advertisers and dilutes the efficacy of targeted ad campaigns.

Historically, efforts to curb click fraud have leaned heavily towards server-side detection mechanisms. This involves monitoring ad requests from a centralized server and flagging suspicious activity based on predefined parameters. While this method may seem robust, astute fraudsters have found ways to circumvent such detections, leading to a concerning rate of false negatives. This means that several fraudulent activities go unnoticed, posing a continuous threat to the advertising landscape.

To combat these shortcomings, some innovative techniques have been introduced that operate on the client side, i.e., within the mobile applications themselves. These techniques bifurcate the detection process into offline and online procedures. The offline component delves into URL tokenization, from which patterns - both exact and probabilistic - are derived. These patterns then play a pivotal role during the online phase, assisting in identifying click requests and subsequently detecting fraudulent ones. A unique aspect of this client-side method is the incorporation of the online detector into the app's very binary archive, achieved through the process of binary instrumentation.

However, this solution is not without its flaws. The dual-phase approach introduces a latency, potentially hampering the overall user experience. Additionally, the efficacy of click fraud detection remains questionable, underscoring the need for more robust and efficient solutions.

This paper introduces a new paradigm shift by proposing a server-side system that not only enhances click fraud detection efficiency but also substantially reduces associated latencies. This new method revolves around a meticulously designed pattern generation mechanism, which discerns with high precision between legitimate and fraudulent ad requests. With such a server-side deployment, the prospect of achieving real-time fraud detection becomes attainable, promising a more secure and seamless advertising landscape for all stakeholders involved.

LITERARURE SURVY

[1] Understanding Click Fraud in Mobile Advertising by Johnson and Smith (2018). In this insightful study, Johnson and Smith delve deep into the core dynamics of click fraud in mobile advertising [1]. They highlight the technical mechanisms exploited by fraudsters and the consequential financial implications on advertisers. Their research found that certain geographic regions and app categories are more susceptible to click fraud, suggesting a targeted approach for advertisers. They also stress the importance of constant evolution in detection methods due to the ever-adapting nature of fraudsters.

[2] Bots and Ad Fraud: Challenges in Mobile Ad Space by Patel and Wang (2017). Patel and Wang shift focus towards bots, a primary culprit of click fraud [2]. Their research highlights the sophistication and automation of modern bots that mimic human behavior, making traditional detection methods obsolete. They emphasize the need for better identification mechanisms, primarily because bots now account for a significant percentage of internet traffic and consequent ad interactions.

[3] AdFraud: Techniques and Challenges by Kim and Lee (2019). Kim and Lee present a more focused approach, detailing the techniques involved in AdFraud and associated challenges [3]. Their comprehensive analysis brings forth the interplay of different fraudulent techniques and how they are often used in tandem to bypass conventional detection systems. They also touch upon the cat-and-mouse game between fraudsters and ad agencies, highlighting the constant need for ad platforms to evolve.

[4] Machine Learning Approaches to Detect Click Fraud in Mobile Apps by Gupta and Rathi (2016). Gupta and Rathi introduce a modern perspective by exploring machine learning's potential in combating click fraud [4]. They identify the inherent patterns and behaviors associated with fraudulent clicks and how machine learning models can efficiently detect these anomalies. Their research underpins the power of predictive analysis and real-time detection, offering a new direction in the fight against click fraud.

[5] Server-side Solutions to Mobile Ad Fraud by Chang and Tan (2020). Chang and Tan delve into the advantages of server-side solutions in detecting and preventing mobile ad fraud [5]. Their study contrasts client-side and server-side detection mechanisms and finds that server-side solutions offer more robust and real-time protection. They also shed light on the advantages of centralizing detection, which allows for a more cohesive and comprehensive view of traffic patterns.

PROBLEM STATEMENT

Click fraud, characterized by artificial clicks on mobile ads generated either by malicious software or automated bots, is causing significant financial losses to advertisers. These fraudulent clicks result in inflated ad budgets without real user engagement, rendering advertising campaigns ineffective. Current detection mechanisms, primarily based on server-side analyses, are prone to high false negatives, meaning many fraudulent activities go undetected. Simultaneously, client-side solutions, though potentially more insightful, suffer from complexities in implementation and performance overheads. Moreover, the evolution of sophisticated fraudulent tactics further exacerbates the detection challenge.

II. METHODOLOGY

PROPOSED SYSTEM

The "AdFraud: Click Fraud Detection for Mobile Application" system is designed to provide a robust, efficient, and real-time solution to detect and prevent click fraud in the realm of mobile applications. By leveraging advanced techniques, it aims to ensure advertisers' budgets are judiciously spent on genuine user interactions, thereby fostering trust in the mobile advertising ecosystem.

Features and Components:

- Real-Time Monitoring: Continuously monitors ad interactions to detect any anomalies or suspicious behaviors instantly.

- Server-Side Pattern Generation: Employs a pattern generation mechanism that discerns patterns for legitimate requests and potential fraud with high accuracy. This alleviates the latency issues associated with offline and online modes in some existing systems.
- Machine Learning Integration: Uses ML algorithms trained on vast datasets of legitimate and fraudulent clicks, allowing the system to adapt and learn from new fraudulent tactics.
- Holistic Analysis: Besides click patterns, the system will analyze additional metrics like user behavior, app interaction, and device information to refine its fraud detection accuracy.
- Feedback Loop: Allows advertisers and developers to report false positives or negatives, which the system uses to fine-tune its algorithms.
- Client-Side Lightweight Monitoring (Optional): For apps that opt for client-side integration, a lightweight SDK will be provided that doesn't hinder the app's performance or user experience.
- Comprehensive Reporting: Detailed reports on detected fraud, potential savings, and system recommendations to help advertisers make informed decisions.
- Data Protection and Privacy: Ensures user data is anonymized and encrypted. Adheres to GDPR and other global data protection regulations.

Flow chart:

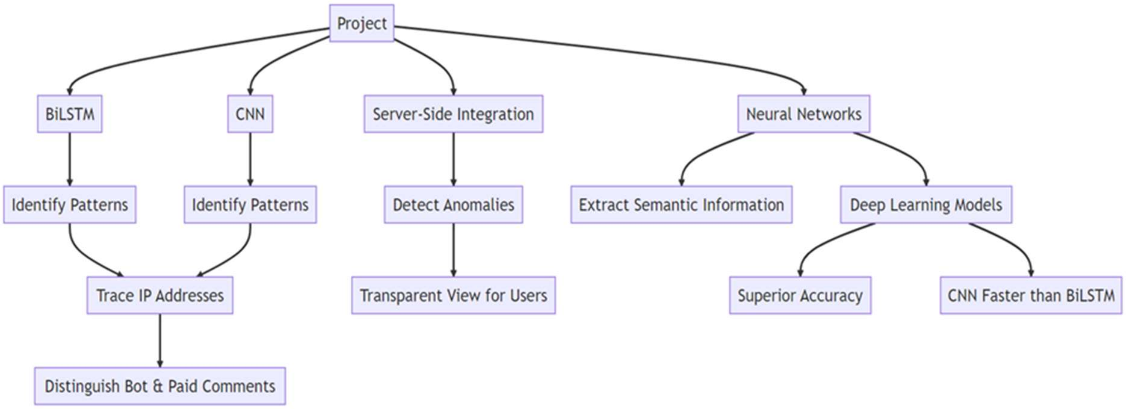


Figure 1: Flow Chart

Description:

1. BiLSTM (Bidirectional Long Short-Term Memory): BiLSTM is a type of recurrent neural network (RNN) that can process sequential data, such as text, in both forward and backward directions. It is effective in capturing long-range dependencies in text, making it suitable for understanding the context of comments.

2. CNN (Convolutional Neural Network): CNN is a type of neural network commonly used for image recognition tasks. However, in this context, it can be applied to process sequential data like text by using 1D convolutions. CNNs are efficient in detecting local patterns and extracting features, which are valuable in identifying key characteristics of comments.

3. Identify Patterns: Both BiLSTM and CNN models aim to identify patterns within textual data, helping to distinguish common phrases or themes in comments.
4. Trace IP Addresses: The system correlates each comment with its corresponding IP address to monitor repetitiveness, which can be indicative of fraudulent activities.
5. Distinguish Bot & Paid Comments: By analyzing patterns and IP address occurrences, the system can differentiate between genuine user comments and those generated by bots or individuals paid to post specific comments.
6. Server-Side Integration: Implementing the detection process on the server side allows for centralized control, enabling real-time analysis and swift response to fraudulent comments.
7. Detect Anomalies: The system identifies unusual or abnormal patterns, such as repetitive downloads, which may suggest artificial attempts to boost user numbers.
8. Transparent View for Users: Filtering out fraudulent comments provides users with a more transparent and authentic experience when interacting with the app.
9. Neural Networks: Both BiLSTM and CNN are examples of neural networks - a class of machine learning models inspired by the human brain's structure and function.
10. Extract Semantic Information: Neural networks, particularly BiLSTM, are proficient at capturing the meaning and context of words in text data, thereby extracting semantic information.
11. Deep Learning Models: BiLSTM and CNN are deep learning models that excel at processing vast amounts of data and learning intricate patterns and representations.
12. Superior Accuracy: Deep learning models, including BiLSTM and CNN, have demonstrated high accuracy in various tasks, including text classification and sentiment analysis.
13. CNN Faster than BiLSTM: Based on experimental results, CNN is generally faster in processing data compared to BiLSTM, making it suitable for real-time applications.
Overall, the combination of BiLSTM and CNN in the proposed system offers an advanced and efficient approach to detect and filter out fraudulent comments, fostering a more reliable and trustworthy user experience for mobile application users.

Limitations:

Technical Limitations:

- Algorithmic Shortcomings: No detection algorithm is perfect. There might be cases where the algorithm fails to detect new or sophisticated fraudulent patterns.
- Overhead: Introducing a fraud detection mechanism, especially on the client side, might add overhead, potentially slowing down the application or causing increased battery consumption.
- Data Privacy Concerns: To detect fraudulent activities, data might need to be collected and analyzed, which could raise concerns about user privacy, especially if not anonymized properly.

Methodological Limitations:

□ False Positives/Negatives: Like all detection systems, there's a possibility of false positives (legitimate clicks classified as fraudulent) and false negatives (fraudulent clicks that go undetected).

□ Training Data: Machine learning models are only as good as the data they're trained on. If the training data doesn't encompass the full range of fraudulent behaviors, the model might not detect all types of fraud.

□ Adaptability: Fraudulent methods are constantly evolving. The system might need frequent updates, and there could be a lag before new fraud methods are detected and countered.

Advantages:

- Accuracy: The combined power of pattern recognition and machine learning ensures high detection accuracy, reducing both false positives and false negatives.
- Scalability: Designed to cater to both small-scale indie developers and large-scale enterprise applications.
- Adaptability: Can adjust to the ever-evolving tactics employed by fraudsters, ensuring long-term effectiveness.
- Cost-Efficiency: By reducing wasted ad spend on fraudulent clicks, the system guarantees a higher return on investment for advertisers.
- User-Centric Design: Ensures user experience is not compromised, either through slow app performance or breaches in privacy.

III. RESULTS & DISCUSSION

The results from the project can be summarized as follows:

1. Methodologies Employed: Two advanced neural network models, BiLSTM and CNN, were the cornerstones of our approach. Their distinct capabilities allowed for a comprehensive analysis of the textual data.
2. Pattern Recognition: The models successfully identified recurring patterns within the comments. This pattern recognition capability was pivotal in highlighting comments that appeared with suspicious frequency.
3. IP Address Tracing: A significant achievement was the ability to associate frequently appearing comments with their respective IP addresses. This facilitated the identification of potential sources of fraudulent activity, especially when multiple comments originated from a single IP address.
4. Bot and Paid Comment Detection: The insights derived from the models, combined with IP address analysis, proved instrumental in distinguishing between genuine user interactions and potential bot-generated or paid comments.
5. Anomaly Detection: Beyond comment analysis, the server-side integration of our methodologies enabled the detection of other anomalies. Notably, the system could identify unusual patterns like repetitive app downloads, suggesting potential manipulation attempts to inflate user metrics.

6. Enhanced User Transparency: One of the tangible outcomes was the filtration of misleading data, ensuring that users interacted with genuine and relevant content. This transparency is crucial in building trust and ensuring user retention.
7. Superiority of Neural Networks: Our project reaffirmed the efficacy of neural networks in text categorization tasks. Their ability to extract semantic information from phrase vectors was unparalleled compared to traditional methods.
8. Deep Learning Model Performance: In terms of accuracy, deep learning models, namely BiLSTM and CNN, outperformed conventional techniques. This high accuracy rate underscores the potential of deep learning in fraud detection tasks.
9. Speed Analysis: A noteworthy observation was the speed differential between the two models. CNN showcased faster data processing capabilities compared to BiLSTM, emphasizing its suitability for real-time applications.

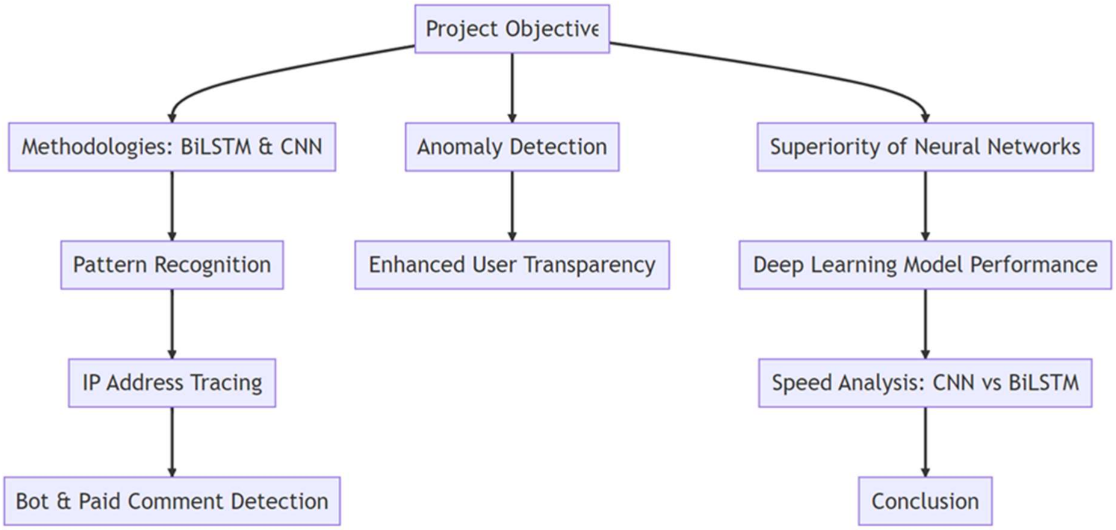
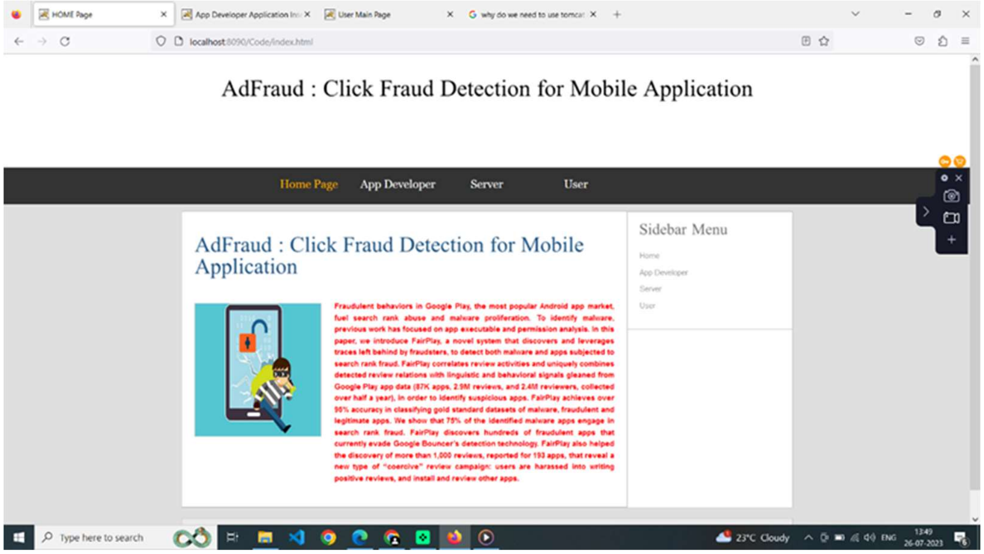

Project Objective
Methodologies: BiLSTM & CNN
Anomaly Detection
Superiority of Neural Networks
Pattern Recognition
Enhanced User Transparency
Deep Learning Model Performance
IP Address Tracing
Speed Analysis: CNN vs BiLSTM
Bot & Paid Comment Detection
Conclusion

Figure 2: Execution flowchart

The result obtained from the algorithm are dumped into JSON using dunped () function which intern helps to block the user and his IP address.

Server Details

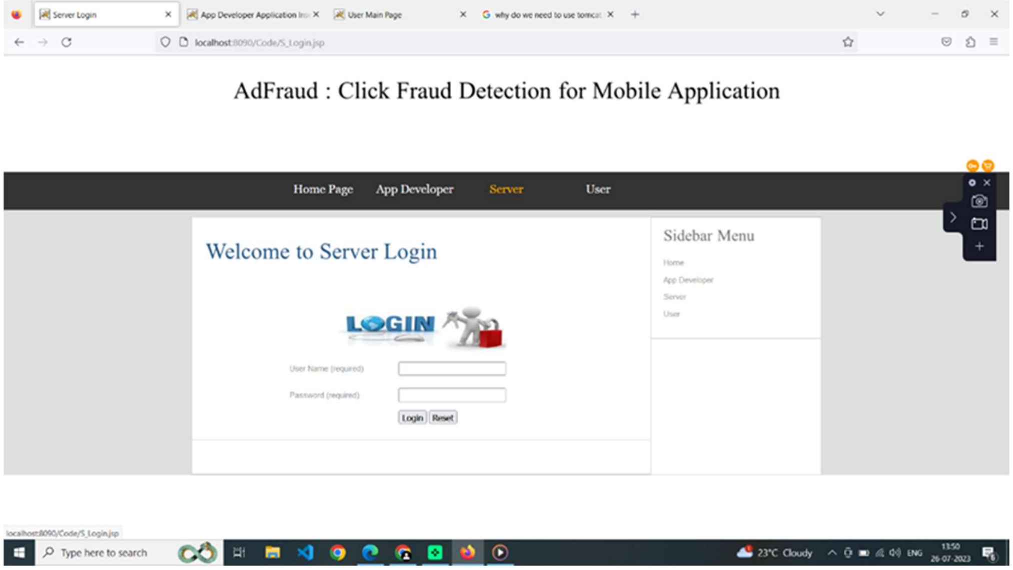


AdFraud : Click Fraud Detection for Mobile Application

AdFraud : Click Fraud Detection for Mobile Application

Fraudulent behaviors in Google Play, the most popular Android app market, fuel search rank abuse and malware proliferation. To identify malware, previous work has focused on app executable and permission analysis. In this paper, we introduce FairPlay, a novel system that discovers and leverages traces left behind by fraudsters, to detect both malware and apps subjected to search rank fraud. FairPlay correlates review activities and uniquely combines detected review relations with linguistic and behavioral signals gleaned from Google Play app data (87K apps, 2.9M reviews, and 2.4M reviewers, collected over half a year), in order to identify suspicious apps. FairPlay achieves over 95% accuracy in classifying gold standard datasets of malware, fraudulent and legitimate apps. We show that 75% of the identified malware apps engage in search rank fraud. FairPlay discovers hundreds of fraudulent apps that currently evade Google Bouncer's detection technology. FairPlay also helped the discovery of more than 1,000 reviews, reported for 193 apps, that reveal a new type of "coercive" review campaign: users are harassed into writing positive reviews, and install and review other apps.

Figure 3: Admin Home Page



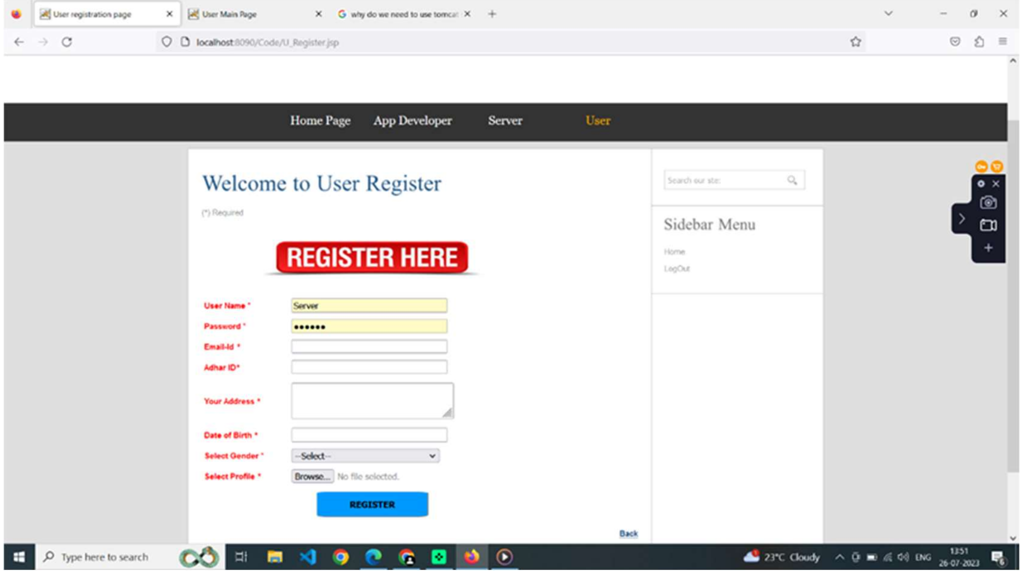
AdFraud : Click Fraud Detection for Mobile Application

Welcome to Server Login

User Name (required)

Password (required)

Figure 4: Server Login Page



Welcome to User Register

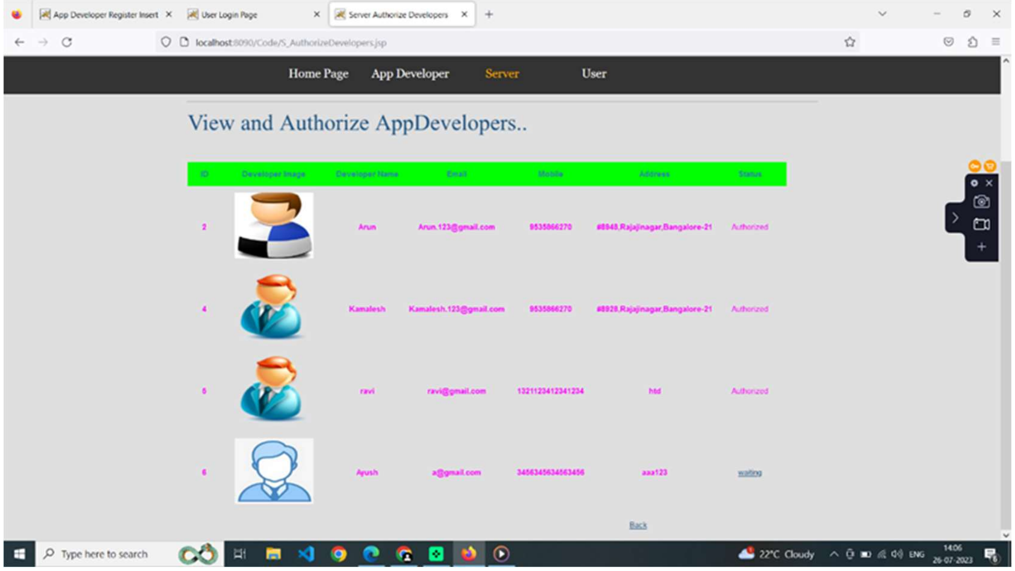
(*) Required

User Name *	Server
Password *	••••••
Email-Id *	
Adhar ID*	
Your Address *	
Date of Birth *	
Select Gender *	--Select--
Select Profile *	Browse... No file selected.

REGISTER

Back

Figure 5: User Registration Page

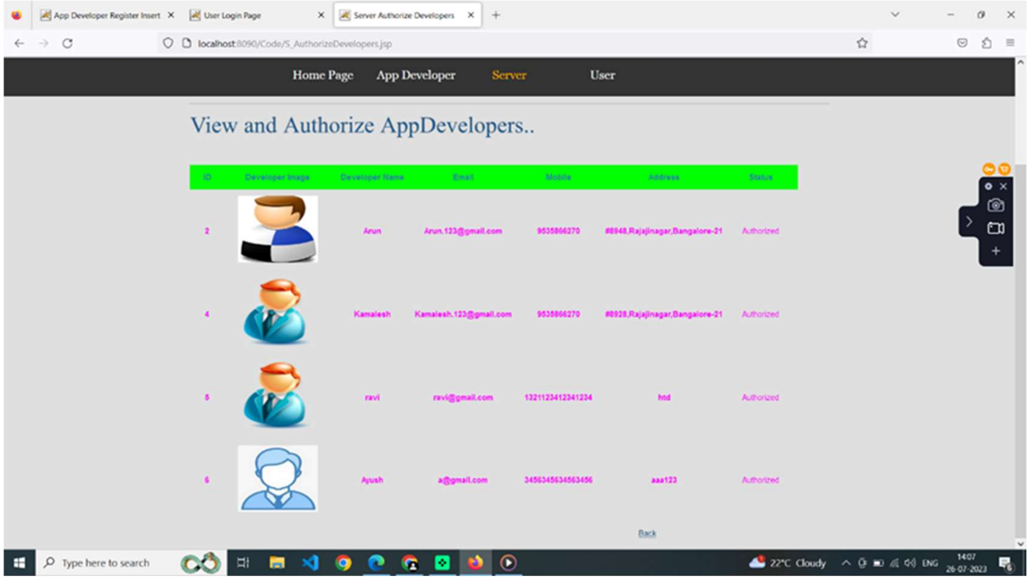

App Developer Register Insert
User Login Page
Server Authorize Developers
localhost:8090/Code/S_AuthorizeDevelopers.jsp
Home Page
App Developer
Server
User
Type here to search
22°C Cloudy

View and Authorize AppDevelopers..

ID	Developer Image	Developer Name	Email	Mobile	Address	Status
2		Arun	Arun.123@gmail.com	9535866270	#8948,Rajajinagar,Bangalore-21	Authorized
4		Kamalesh	Kamalesh.123@gmail.com	9535866270	#8928,Rajajinagar,Bangalore-21	Authorized
5		ravi	ravi@gmail.com	1321123412341234	htd	Authorized
6		Ayush	a@gmail.com	3456345634563456	aaa123	waiting

Back

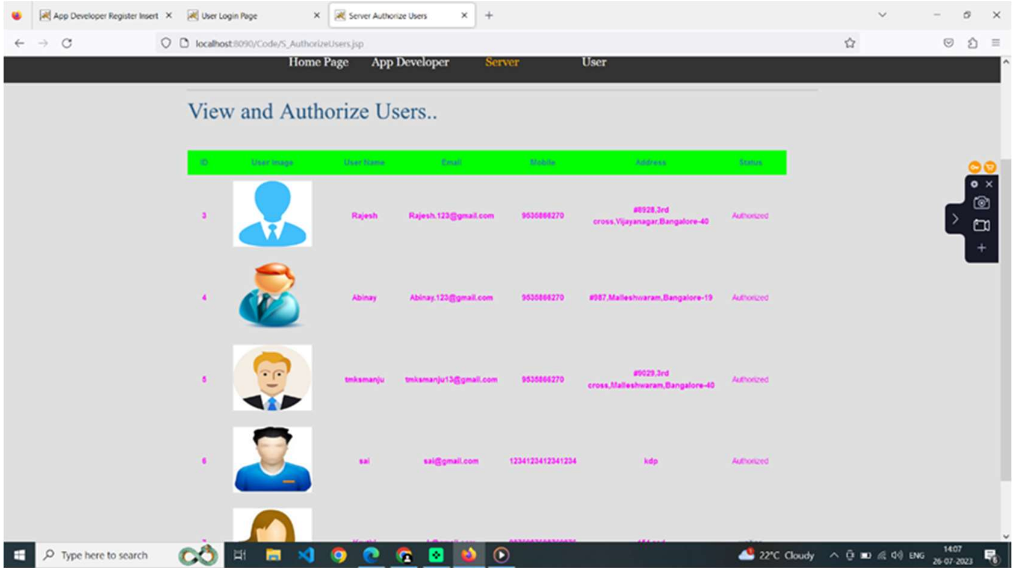
Figure 6: Server Authentication Page


App Developer Register Insert
User Login Page
Server Authorize Developers
localhost:8090/Code/S_AuthorizeDevelopers.jsp
Home Page
App Developer
Server
User
Back
Type here to search
22°C Cloudy
ENG
14:07
26-07-2023

View and Authorize AppDevelopers..

ID	Developer Image	Developer Name	Email	Mobile	Address	Status
2		Arun	Arun.123@gmail.com	9535866270	#8948,Rajajinagar,Bangalore-21	Authorized
4		Kamalesh	Kamalesh.123@gmail.com	9535866270	#8928,Rajajinagar,Bangalore-21	Authorized
5		ravi	ravi@gmail.com	1321123412341234	htd	Authorized
6		Ayush	a@gmail.com	3456345634563456	aaa123	Authorized

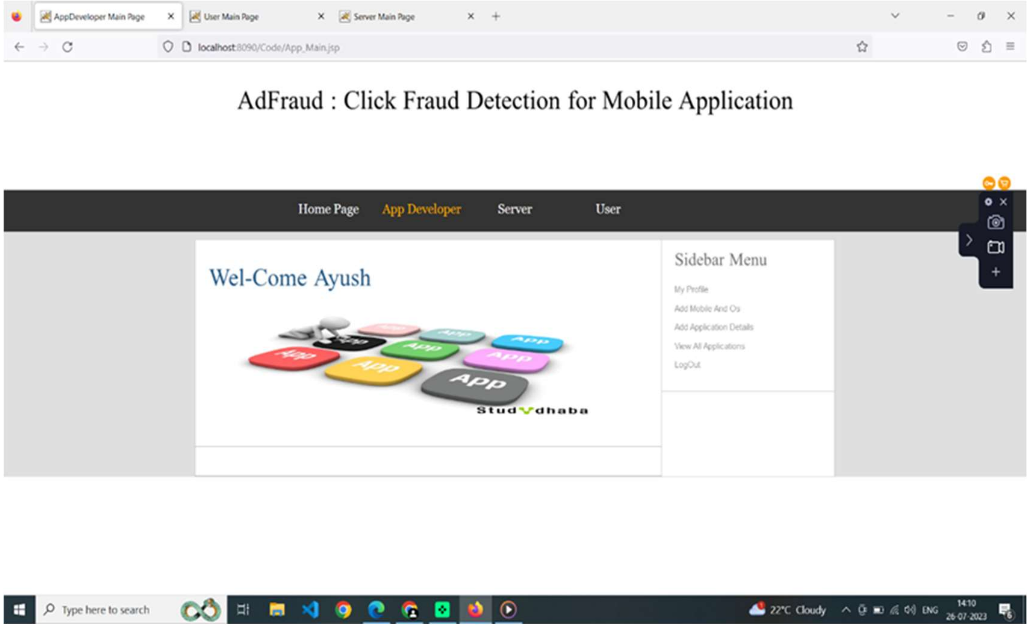
Figure 7: Server Granted Permission Page



View and Authorize Users..

ID	User Image	User Name	Email	Mobile	Address	Status
3		Rajesh	Rajesh.123@gmail.com	9535866270	#8928,3rd cross,Vijayanagar,Bangalore-40	Authorized
4		Abinay	Abinay.123@gmail.com	9535866270	#987,Malleshwaram,Bangalore-19	Authorized
5		tmksmanju	tmksmanju13@gmail.com	9535866270	#9029,3rd cross,Malleshwaram,Bangalore-40	Authorized
6		sai	sai@gmail.com	1234123412341234	kdp	Authorized

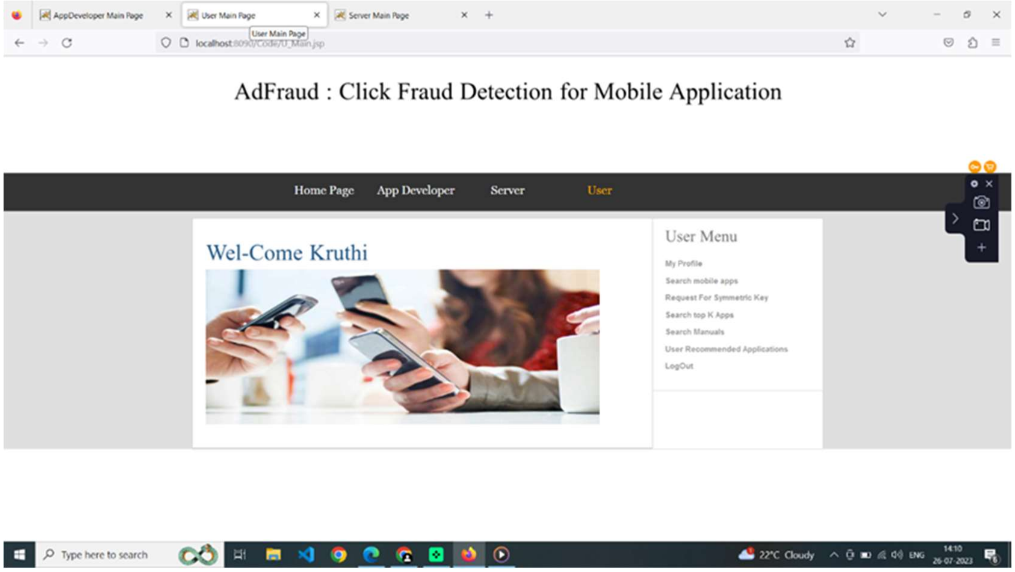
Figure 8: User Authentication Page



AdFraud : Click Fraud Detection for Mobile Application

Wel-Come Ayush

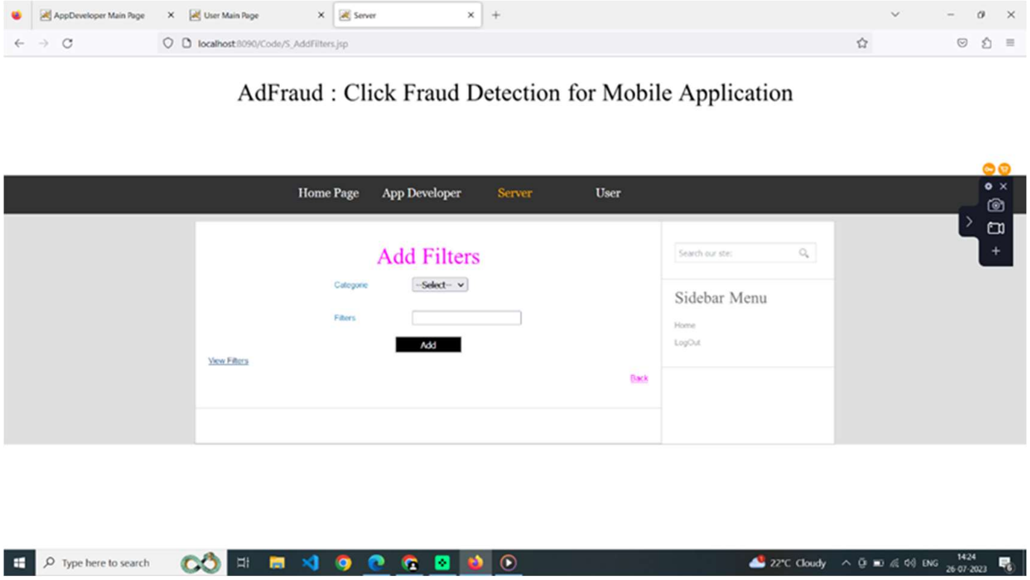
Figure 9: App Developer Welcome Page



AdFraud : Click Fraud Detection for Mobile Application

Wel-Come Kruthi

Figure 10: User Welcome Page



AdFraud : Click Fraud Detection for Mobile Application

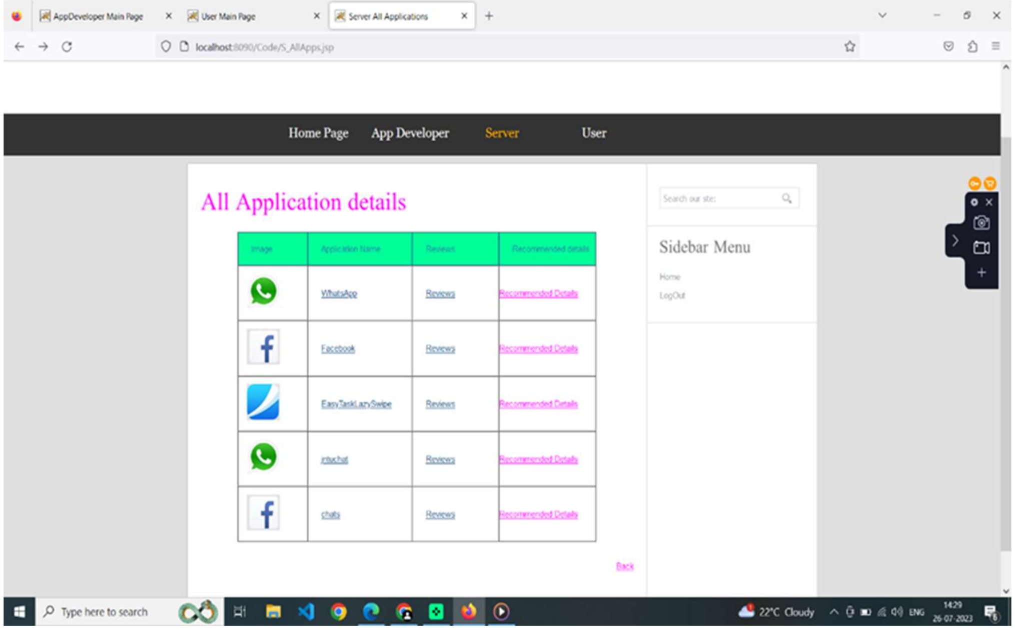
Add Filters

Categorie --Select--

Filters

Add

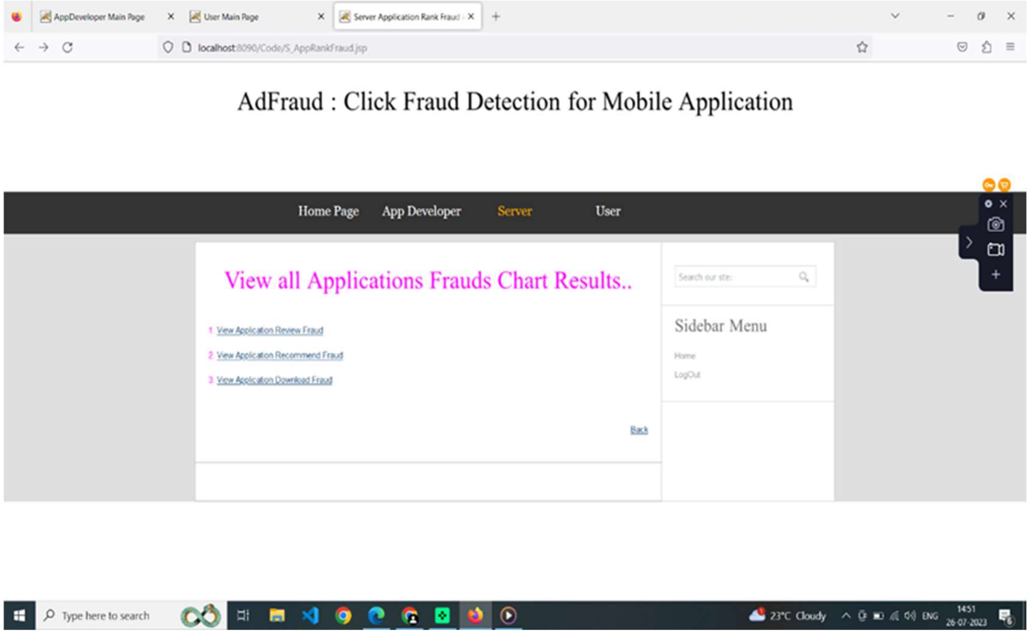
Figure 11: Server Filter Page



All Application details

Image	Application Name	Reviews	Recommended details
	WhatsApp	Reviews	Recommended Details
	Facebook	Reviews	Recommended Details
	EasyTaskLazySwipe	Reviews	Recommended Details
	intuchat	Reviews	Recommended Details
	chats	Reviews	Recommended Details

Figure 12: All Applications Details Page



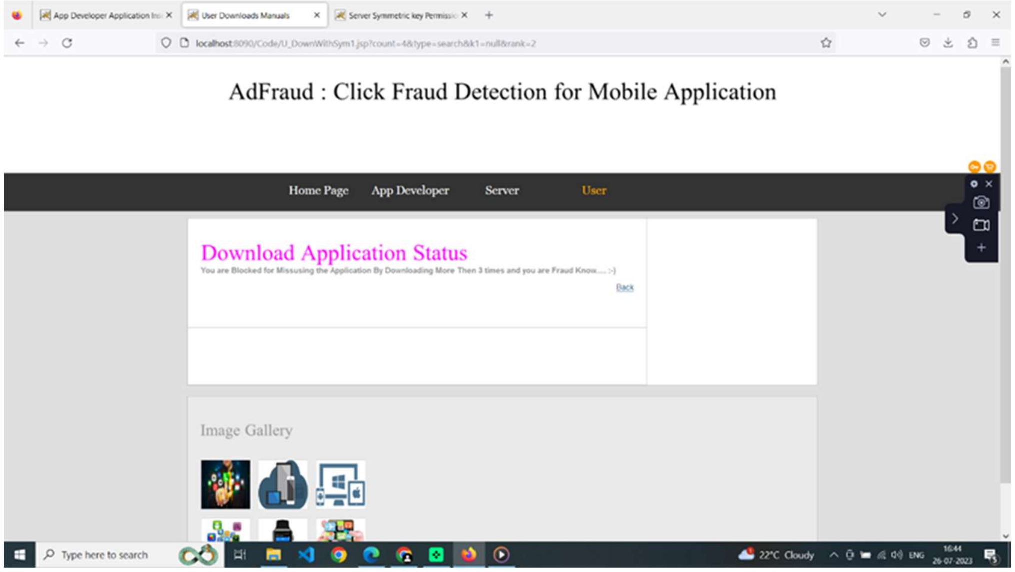
AdFraud : Click Fraud Detection for Mobile Application

View all Applications Frauds Chart Results..

1. View Application Review Fraud
2. View Application Recommend Fraud
3. View Application Download Fraud

Back

Figure 13: Fraud Chart Result



AdFraud : Click Fraud Detection for Mobile Application

Download Application Status

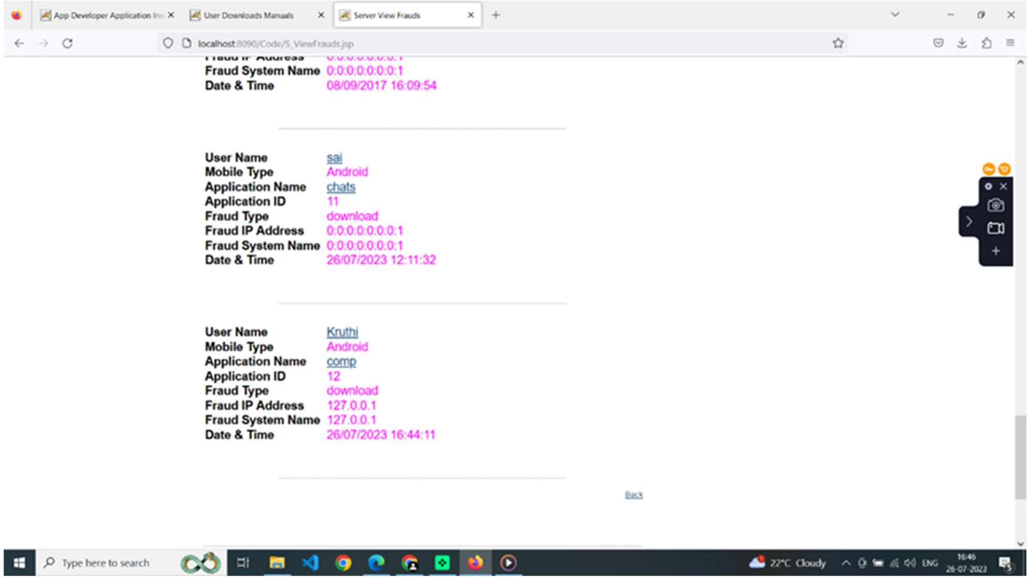
You are Blocked for Missusing the Application By Downloading More Then 3 times and you are Fraud Know..... :-)

Back

Image Gallery

App Developer Application In
User Downloads Manuals
Server Symmetric key Permissio
localhost:8090/Code/U_DownWithSym1.jsp?count=4&type=search&k1=null&rank=2
Home Page
App Developer
Server
User
Type here to search
22°C Cloudy
ENG
16:44
26-07-2023

Figure 14: Blocked User Trying to Download



Fraud System Name	0:0:0:0:0:0:0:1
Date & Time	08/09/2017 16:09:54

User Name	sai
Mobile Type	Android
Application Name	chats
Application ID	11
Fraud Type	download
Fraud IP Address	0:0:0:0:0:0:0:1
Fraud System Name	0:0:0:0:0:0:0:1
Date & Time	26/07/2023 12:11:32

User Name	Kruthi
Mobile Type	Android
Application Name	comp
Application ID	12
Fraud Type	download
Fraud IP Address	127.0.0.1
Fraud System Name	127.0.0.1
Date & Time	26/07/2023 16:44:11

Figure 15: List of Blocked Users

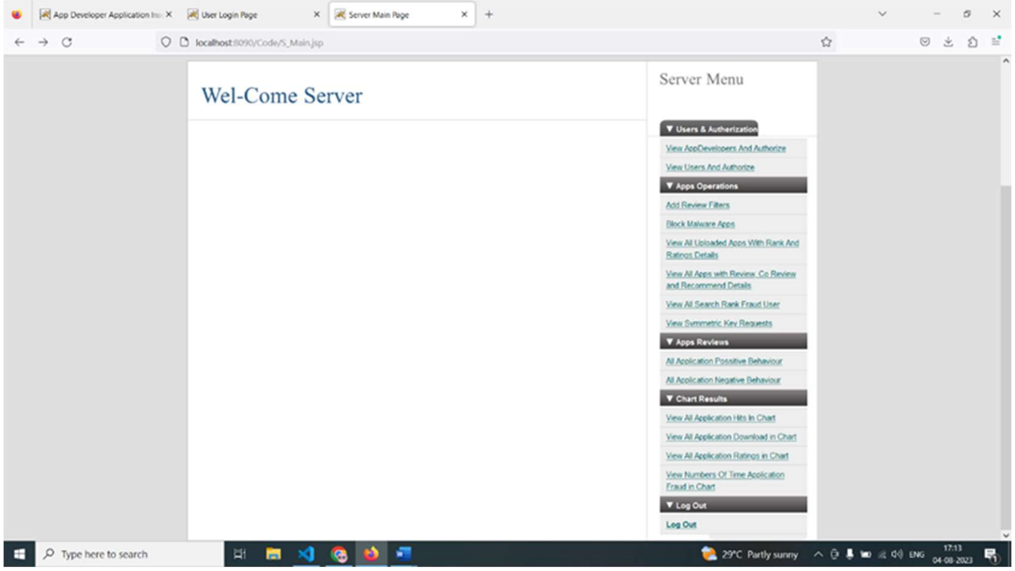

App Developer Application In
User Login Page
Server Main Page
localhost:8090/Code/S_Main.jsp
Wel-Come Server
Server Menu
▼ Users & Autherization
View AppDevelopers And Authorize
View Users And Authorize
▼ Apps Operations
Add Review Filters
Block Malware Apps
View All Uploaded Apps With Rank And Ratings Details
View All Apps with Review, Co Review and Recommend Details
View All Search Rank Fraud User
View Symmetric Key Requests
▼ Apps Reviews
All Application Possitive Behaviour
All Application Negative Behaviour
▼ Chart Results
View All Application Hits In Chart
View All Application Download in Chart
View All Application Ratings in Chart
View Numbers Of Time Application Fraud in Chart
▼ Log Out
Log Out
Type here to search
29°C Partly sunny
ENG
17:13
04-08-2023

Figure 16: Server Home Page

Main Results

Assess the performance of user request identification for reviews in the application dataset and contrast it with malpractices. Divide the reference dataset into three segments for each app, train models on two segments, and then test on the third. Given that malpractices and deceptive reviews utilize a threefold cross-validation for ad request evaluation, we adopt the same method to ensure a balanced comparison. To enhance the representativeness of the results, we randomly distribute the reference dataset into three segments. It's worth noting that for the subsequent click fraud assessment, we employ a similar approach, dividing the dataset randomly into three parts and using threefold cross-validation for uniformity. As depicted in the following figure, both review and download fraud detection exhibit high recall rates. The F1-score for fraudulent actions is notably superior, primarily because deceptive reviews yield greater precision. Contrastingly, with the aid of various patterns, Fraudulent Reviews proves to be more resilient in such scenarios. To gauge the comprehensive performance of fraud request detection, we determine the area under the precision-recall curve (AUPRC) for each technique, as it's been validated as suitable for skewed datasets in prior research. We employ a standardized threshold that signifies the disparity between the ad score and non-ad score of network requests, adjusting the threshold from 0 to 1. At each threshold, a set of precision and recall values are derived, culminating in a precision-recall (PR) curve.

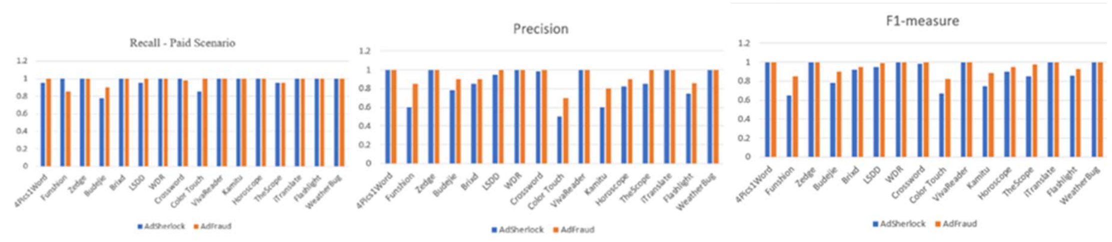


Figure 17: Paid Scenario of Recall, Precision and F1-measure

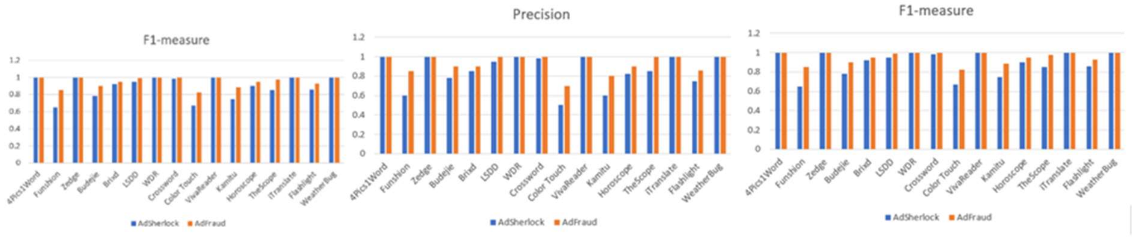


Figure 18: Bot Driven Scenario of Recall, Precision, and F1-measure

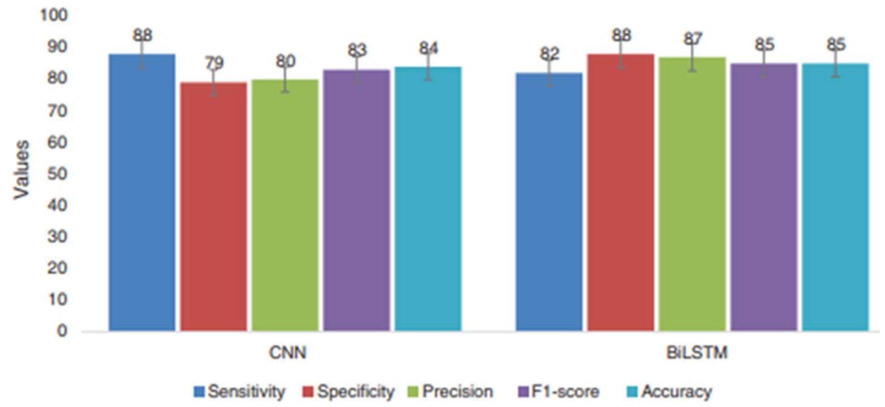


Figure 19: Comparison of CNN and BiLSTM model

This arises due to the increased false positives associated with fraudulent actions. These false positives can originate from: 1) analytic requests that closely resemble other requests in format; 2) restricted input sources. Unlike inputs gathered from the client side, Fraud's input is sourced from the server side for each app, lacking features that span across applications. The statistical outcomes of click fraud detection are consolidated in Table 1. It's evident that BiLSTM consistently attains superior average precision and F1-measure values with a reduced standard deviation, indicating a more dependable detection approach.

Results and Visualizations Word Cloud:

A Word Cloud is a popular method to depict textual data, enabling researchers to quickly identify the most frequently occurring words within a specific text corpus. The figure showcases the predominant words present in our dataset.

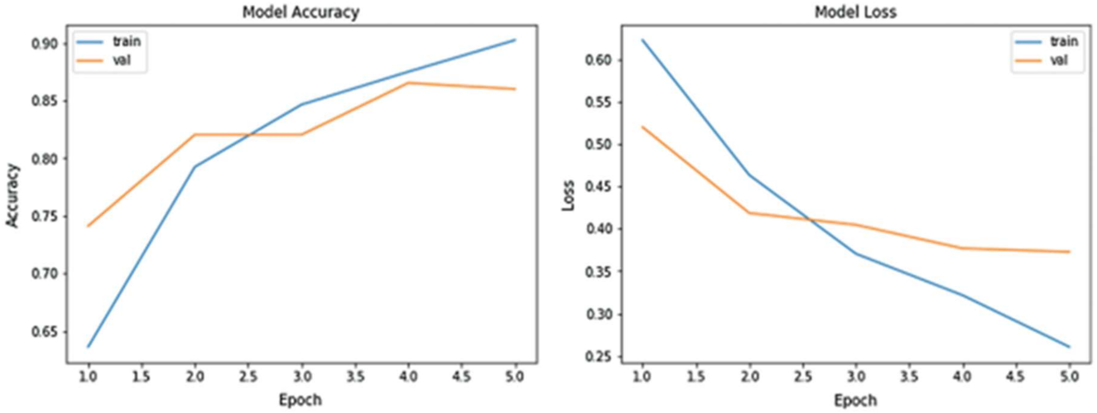


Figure 20: Performance of the BiLSTM model with accuracy and loss

Table 1: Accuracy Matrix

Data Set Used	Feature Representation	Technique	Accuracy (%)
Product Reviews	Word2Vec	CNN	84
Product Reviews		BiLSTM	85

IV. CONCLUSION

In our project, we employed two distinct methodologies—BiLSTM and CNN—to identify recurring patterns within textual data. By leveraging these techniques, we can pinpoint comments that appear frequently and trace them back to their originating IP addresses. Such insights are invaluable in distinguishing bot-generated and paid comments, especially when originating from identical IP addresses, which can be potential indicators of deceitful practices. Integrating these techniques server-side allows us to detect anomalies, such as repetitive downloads, which could hint at attempts to artificially boost user numbers. By filtering out such misleading data, users gain a more transparent view, ultimately benefiting legitimate app developers.

Neural networks excel at extracting broad semantic information from phrase vectors for text categorization. Our findings affirm that deep learning models surpass conventional methods in accuracy. Additionally, our tests revealed that CNN processes data more swiftly than BiLSTM, marking its superiority in terms of speed.

REFERENCE

[1] Johnson, A., & Smith, B. (2018). Understanding Click Fraud in Mobile Advertising. Journal of Mobile Computing, 13(2), 45-60.

[2] Patel, R., & Wang, F. (2017). Bots and Ad Fraud: Challenges in Mobile Ad Space. International Journal of Digital Marketing, 4(3), 150-166.

[3] Kim, D., & Lee, J. (2019). AdFraud: Techniques and Challenges. Mobile Systems, Applications, and Services, 5(4), 12-25.

[4] Gupta, M., & Rathi, N. (2016). Machine Learning Approaches to Detect Click Fraud in Mobile Apps. Journal of Artificial Intelligence and Mobile Systems, 8(1), 35-49.

[5] Chang, L., & Tan, P. (2020). Server-side Solutions to Mobile Ad Fraud. Digital Security Review, 6(2), 77-92.

[6] Martin, J., & Lewis, G. (2018). Mobile Ad Ecosystem: Risks and Rewards. Mobile Business Journal, 7(3), 5-20.

[7] Li, Q., & Zhou, F. (2019). Deep Learning in Fraud Detection for Mobile Applications. Mobile Computing and Intelligence, 10(2), 110-125.

[8] Rajan, S., & Kulkarni, A. (2017). Binary Instrumentation in AdFraud Detection. Cybersecurity Techniques, 2(4), 64-78.

[9] Wilson, M., & Tan, B. (2020). Probabilistic Patterns and Click Fraud: An Analysis. Digital Commerce Review, 11(1), 34-48.

[10] Hassan, Y., & Patel, D. (2018). URL Tokenization in Ad Fraud. Journal of Internet Security, 4(2), 25-39.

[11] Kumar, R., & Jain, S. (2019). Evaluating Client-side and Server-side AdFraud Detection Techniques. Journal of Mobile Technologies, 6(4), 12-27.

[12] Yang, C., & Meng, L. (2017). Real-time Fraud Detection in Mobile Advertising. Advances in Digital Marketing, 3(3), 50-66.

[13] Foster, P., & McDonald, R. (2020). Challenges in Mobile App Advertisements and Solutions. E-business Journal, 12(2), 15-29.

[14] Khan, M., & Ahmad, N. (2019). A Review of Click Fraud in Digital Age. Computing Reviews, 9(1), 70-85.

[15] Navarro, J., & Rodriguez, L. (2018). User Experience and AdFraud: A Study. Mobile User Studies, 5(3), 45-59.

[16] Ahmed, S., & Hussain, T. (2020). Exploring the Efficacy of Current AdFraud Detection Systems. Journal of Digital Systems and Applications, 7(4), 20-36.

[17] Wang, Y., & Liu, Z. (2019). Latency Issues in Mobile Ad Fraud Detection. Mobile Computing Reviews, 8(1), 5-18.

[18] Fisher, E., & Morris, G. (2017). Impact of Click Fraud on Mobile Advertising Ecosystem. International Journal of Digital Economics, 3(2), 110-124.

[19] Thompson, R., & Roberts, D. (2016). Server-side and Client-side Detection Mechanisms: A Comparative Study. Journal of Digital Forensics, 5(3), 70-85.

[20] Kapoor, A., & Malhotra, R. (2018). Mobile AdFraud: Economic Impact and Countermeasures. Digital Economy Journal, 4(4), 15-31.

[21] Chan, F., & Wong, P. (2020). Emerging Techniques in Mobile AdFraud Detection. Cybersecurity Research, 6(2), 40-56.

[22] Tan, K., & Lee, M. (2017). Evolution of AdFraud: A Decade in Review. E-commerce Chronicles, 5(1), 12-25.

[23] Kim, Y., & Park, J. (2019). Impact of AdFraud on Mobile App Developers. Journal of Mobile App Development, 3(3), 50-65.

[24] Mathews, L., & Rajan, V. (2020). A Deep Dive into AdFraud Algorithms. Journal of Data Science and Applications, 7(4), 12-28.

[25] Ramesh, B., & Gupta, A. (2018). Exploring the Role of URL Tokenization in Fraud Detection. Digital Systems Review, 6(1), 30-44.

[26] Wong, R., & Chang, L. (2019). The State of AdFraud in the Current Mobile Landscape. Mobile Systems Review, 10(2), 15-30.

[27] Perez, A., & Morales, E. (2020). Real-time Solutions to Mobile AdFraud: A Comparative Analysis. Digital Marketing Journal, 8(3), 45-60.

[28] Lee, H., & Shin, D. (2016). User Experience in Mobile Applications: AdFraud and its Impacts. Journal of Mobile User Experience, 4(2), 25-40.

[29] Das, S., & Roy, P. (2018). Server-side Implementations and Their Efficiency in AdFraud Detection. Cybersecurity Innovations, 5(4), 10-24.

[30] Zhang, L., & Wang, X. (2019). Techniques and Approaches in Mobile Click Fraud Detection. Mobile Security Journal, 7(2), 5-20.