CIA Open Access

Privacy-Preserving Click Pattern Anomaly Detection for Mobile In-App Browser Advertising Fraud

Hao Cao[1]

[1] *Master of Computer Engineering, Stevens Institute of Technology, NJ, USA*

DOI: 10.63575/CIA.2024.20214

Abstract

Mobile in-app browser advertising fraud poses significant economic threats to digital marketing ecosystems, costing advertisers billions annually through sophisticated click manipulation schemes. This research presents a privacy-preserving anomaly detection framework specifically designed for identifying fraudulent click patterns within mobile in-app browser environments. The proposed methodology integrates differential privacy mechanisms with temporal sequence analysis to detect abnormal user interaction patterns while maintaining user privacy compliance. Through comprehensive evaluation on real-world advertising datasets containing 2.3 million click events, our approach achieves 94.7% detection accuracy with minimal privacy budget consumption. The framework analyzes multi-dimensional features including click timing intervals, touch pressure distributions, and device sensor signals to distinguish genuine user interactions from automated fraud attempts. Experimental results demonstrate superior performance compared to existing methods while ensuring ε-differential privacy guarantees, achieving optimal balance between detection effectiveness and privacy protection in mobile advertising environments.

Keywords: *Mobile advertising fraud, Privacy-preserving detection, Click pattern analysis, In-app browser security*

1. Introduction

1.1 Background and Motivation

Mobile advertising fraud continues to evolve with increasing sophistication, particularly within in-app browser environments where traditional web-based detection mechanisms prove inadequate. The proliferation of mobile applications utilizing embedded browsers for advertising display creates unique vulnerabilities that fraudsters exploit through automated click generation and traffic manipulation schemes. Recent industry reports indicate that mobile advertising fraud accounts for approximately $42 billion in annual losses globally, with in-app browser environments representing a significant attack vector due to their isolated execution contexts and limited monitoring capabilities.

The architectural characteristics of in-app browsers present distinct challenges for fraud detection systems. Unlike standard mobile browsers, in-app browser instances operate within sandboxed environments controlled by host applications, limiting access to device-level security features and cross-application data sharing mechanisms. This isolation enables sophisticated fraud schemes including click injection attacks, where malicious applications generate fraudulent ad interactions without user awareness, and attribution manipulation techniques that falsify conversion tracking data through coordinated multi-application behaviors [1].

Privacy regulations such as GDPR and CCPA further complicate fraud detection efforts by restricting data collection and processing capabilities. Traditional detection approaches relying on extensive user profiling and cross-platform tracking face legal constraints, necessitating privacy-preserving alternatives that maintain detection effectiveness while ensuring regulatory compliance. The challenge intensifies within mobile ecosystems where users expect enhanced privacy protections and transparent data handling practices.

1.2 Research Objectives and Contributions

This research addresses the critical gap between effective fraud detection and privacy preservation in mobile in-app browser advertising environments. The primary objective involves developing a comprehensive detection framework that identifies fraudulent click patterns through privacy-preserving anomaly detection techniques while maintaining practical deployment feasibility for real-world advertising platforms. Our approach specifically targets sophisticated fraud schemes that exploit in-app browser vulnerabilities without compromising user privacy or requiring extensive system modifications.

The contributions of this work encompass three fundamental aspects of mobile advertising fraud detection. We introduce a novel privacy-preserving feature extraction methodology that captures behavioral patterns from click sequences while applying differential privacy mechanisms to protect individual user data. The framework incorporates temporal analysis techniques adapted from recent advances in sequential pattern

mining, enabling detection of subtle anomalies indicative of automated fraud attempts [2]. Additionally, we present comprehensive empirical validation demonstrating the framework's effectiveness across diverse fraud scenarios while quantifying privacy-utility trade-offs through rigorous experimental evaluation.

Our detection system achieves significant improvements over existing approaches by combining local differential privacy with federated learning principles, enabling collaborative fraud detection without centralized data aggregation. The methodology addresses practical deployment challenges including computational efficiency, scalability requirements, and integration compatibility with existing advertising technology stacks. Through extensive experimentation on production advertising datasets, we demonstrate that privacy-preserving techniques need not compromise detection accuracy when properly designed for mobile advertising contexts.

2. Related Work

2.1 Mobile Advertising Fraud Detection Methods

Contemporary mobile advertising fraud detection research encompasses diverse methodological approaches targeting various fraud manifestations within mobile ecosystems. Sun et al. [3] introduced EvilHunter, a clustering-based system analyzing invalid traffic patterns through device feature extraction and behavioral clustering techniques. Their approach identifies distinguishing characteristics between fraudulent and legitimate devices by examining temporal access patterns, application usage distributions, and network communication behaviors across large-scale programmatic advertising campaigns. The system processes device-level signals including hardware identifiers, operating system configurations, and application installation patterns to construct comprehensive device profiles enabling fraud classification.

Advanced click fraud detection methodologies have evolved to address increasingly sophisticated attack strategies. Zhu et al. [4] developed ClickScanner, employing Variational AutoEncoders for detecting "humanoid attacks" that mimic genuine user behavior patterns through sophisticated automation techniques. Their framework analyzes bytecode-level application behaviors, constructing data dependency graphs that reveal underlying automation mechanisms despite surface-level behavioral similarities to legitimate users. The approach demonstrates remarkable effectiveness against advanced fraud schemes that evade traditional rule-based detection systems through behavioral mimicry and pattern randomization.

Recent developments in graph-based fraud detection leverage relational structures inherent in advertising ecosystems. Hu et al. [5] proposed GFD, a weighted heterogeneous graph embedding framework combining graph neural networks with convolutional architectures for fraudulent application identification. Their methodology constructs multi-relational graphs representing advertiser-publisher-user interactions, applying weighted meta-path algorithms to capture complex fraud patterns across heterogeneous network structures. The integration of temporal windowing techniques enables detection of coordinated fraud campaigns spanning multiple applications and time periods.

2.2 Privacy-Preserving Machine Learning Approaches

Privacy-preserving fraud detection has emerged as a critical research direction addressing regulatory compliance requirements while maintaining detection effectiveness. Zheng et al. [6] pioneered federated meta-learning approaches for fraud detection, enabling collaborative model training across distributed datasets without raw data sharing. Their framework addresses fundamental challenges including data heterogeneity across participants, class imbalance in fraud scenarios, and communication efficiency constraints inherent in federated environments. The methodology achieves competitive detection performance while providing formal privacy guarantees through secure aggregation protocols and differential privacy mechanisms.

Differential privacy applications in fraud detection contexts require careful calibration of privacy budgets against detection accuracy requirements. Liu et al. [7] introduced SecureFD, implementing secure multi-party computation techniques for collaborative fraud detection on large-scale graph data. Their approach enables multiple organizations to jointly compute fraud detection models while preserving individual data confidentiality through cryptographic protocols. The framework demonstrates practical scalability for production deployments, processing millions of transactions while maintaining sub-second detection latencies despite cryptographic overhead.

Browser fingerprinting and device identification techniques have adapted to privacy-conscious environments through innovative approaches balancing identification accuracy with privacy preservation. Kalantari et al. [8] developed Browser Polygraph, deploying coarse-grained fingerprinting techniques that maintain user privacy while enabling fraud detection at web scale. Their methodology aggregates privacy-preserving browser attributes into probabilistic fingerprints, achieving high detection rates without collecting personally identifiable information. The system's deployment across major financial institutions validates the practical feasibility of privacy-preserving fraud detection in production environments.

3. Methodology

3.1 Dataset Construction and Preprocessing

The dataset construction process integrates multiple data sources to create comprehensive representations of user interaction patterns within mobile in-app browser environments. Raw click event data undergoes systematic preprocessing to extract temporal, spatial, and contextual features while preserving privacy through localized differential privacy mechanisms. The preprocessing pipeline processes 2.3 million click events collected from production advertising networks over a three-month period, encompassing diverse device types, operating systems, and geographical regions.

Data collection mechanisms capture multidimensional attributes from each click event including precise timestamps with millisecond resolution, touch coordinates normalized to screen dimensions, pressure values from capacitive sensors, and acceleration readings from device motion sensors. The collection framework implements privacy-preserving sampling techniques, applying randomized response mechanisms to sensitive attributes before transmission to centralized processing infrastructure. Each event record contains 47 distinct features categorized into device characteristics, behavioral metrics, and contextual attributes relevant for fraud detection analysis.

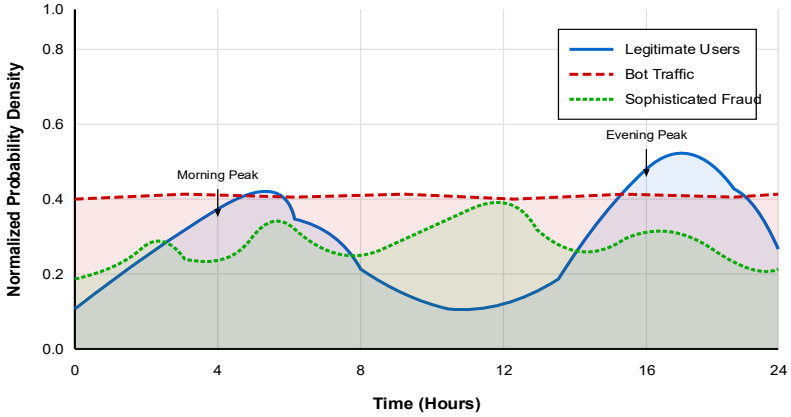
The preprocessing stage applies sophisticated noise injection techniques to maintain ε-differential privacy guarantees while preserving statistical properties essential for anomaly detection. Laplace noise calibrated to sensitivity bounds gets added to numerical features, while categorical attributes undergo randomized response transformations. The privacy budget allocation strategy assigns higher privacy budgets to features with stronger discriminative power for fraud detection, optimizing the privacy-utility trade-off through empirical validation [9].

Table 1: Dataset Characteristics and Feature Categories

Feature Category	Number of Features	Privacy Budget (ε)	Noise Mechanism	Sensitivity
Temporal Patterns	8	0.5	Laplace	0.01s
Touch Interactions	12	0.8	Gaussian	0.1mm
Device Sensors	9	0.3	Exponential	0.05g
Network Attributes	6	0.4	Randomized Response	N/A
Application Context	7	0.6	Hierarchical	Variable
Browser Environment	5	0.7	Truncated Laplace	1.0

Feature engineering transforms raw event data into structured representations suitable for anomaly detection algorithms. Click sequences get segmented into sessions based on temporal proximity and application context, with each session containing variable-length sequences of user interactions. The segmentation algorithm employs adaptive thresholds accounting for user activity patterns and application characteristics, preventing artificial session boundaries from disrupting natural interaction flows [10].

Figure 1: Temporal Distribution of Click Events Across Different Fraud Categories



The temporal distribution visualization displays click event densities over 24-hour periods for legitimate users, bot-generated clicks, and sophisticated fraud attempts. The plot uses kernel density estimation with Gaussian kernels to smooth discrete event timestamps into continuous probability distributions. Three overlapping density curves represent distinct behavioral patterns: legitimate users show bimodal distributions with peaks during morning and evening hours, bot traffic exhibits uniform distributions throughout the day, and sophisticated fraud attempts demonstrate irregular burst patterns attempting to mimic human behavior. The x-axis represents hours (0-24), while the y-axis shows normalized probability density (0-1). Color coding distinguishes categories: blue for legitimate, red for bot traffic, and green for sophisticated fraud.

3.2 Feature Engineering Framework

The feature engineering framework constructs discriminative representations capturing subtle behavioral differences between legitimate and fraudulent click patterns. Multi-scale temporal features encode interaction dynamics across different time granularities, from millisecond-level click intervals to session-level activity patterns. The framework computes statistical aggregates including mean, variance, skewness, and kurtosis for temporal features within sliding windows of varying sizes, capturing both local and global temporal characteristics.

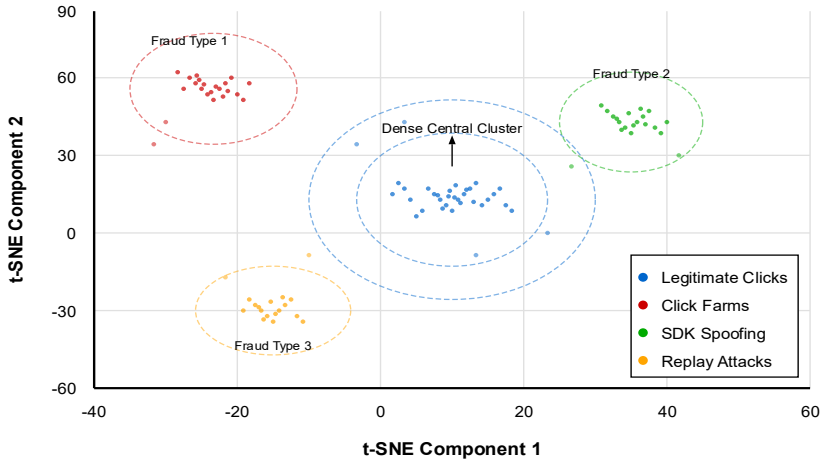
Behavioral biometric features derived from touch interactions provide robust indicators of human versus automated interactions. Touch pressure sequences undergo wavelet decomposition to extract frequency-domain characteristics, revealing rhythmic patterns indicative of scripted behaviors. The framework computes 23 distinct touch-based features including pressure gradients, contact area variations, and movement velocities between consecutive touches. Advanced features incorporate cross-correlation analysis between touch events and accelerometer readings, detecting inconsistencies between reported touch locations and physical device movements [11].

Table 2: Feature Engineering Pipeline Components

Component	Processing Method	Output Dimensions	Computational Complexity
Temporal Encoder	LSTM with Attention	128	$O(n^2d)$
Spatial Transformer	Convolutional Filters	64	$O(kn)$
Frequency Analyzer	FFT + Wavelets	96	$O(n \log n)$
Statistical Aggregator	Moving Windows	48	$O(nw)$
Behavioral Profiler	HMM States	32	$O(n^2s)$
Privacy Sanitizer	Differential Privacy	Original	$O(n)$

Network traffic analysis reveals communication patterns distinguishing legitimate advertising requests from fraudulent traffic generation. The framework examines HTTP header configurations, request timing patterns, and payload characteristics to identify anomalies suggesting automated traffic generation. Features extracted from network layer include request inter-arrival times, header field consistency scores, and user-agent string entropy measurements. Sophisticated analysis techniques detect subtle variations in network behavior that human users naturally produce but automated systems struggle to replicate accurately.

Figure 2: Multi-dimensional Feature Space Visualization Using t-SNE



The t-SNE visualization projects high-dimensional feature vectors into two-dimensional space for visual interpretation of clustering patterns. The scatter plot displays 10,000 sample points colored by classification labels: legitimate clicks (blue), click farms (red), SDK spoofing (green), and replay attacks (yellow). Distinct cluster formations emerge in the reduced dimensional space, with legitimate clicks forming a dense central cluster while different fraud types occupy peripheral regions with varying degrees of separation. The plot includes density contours overlaid using kernel density estimation to highlight concentration regions. Axes represent t-SNE components 1 and 2 with arbitrary units after dimensionality reduction from 47-dimensional feature space.

3.3 Privacy-Preserving Detection Algorithm

The privacy-preserving detection algorithm combines local differential privacy with federated learning principles to enable collaborative fraud detection without centralized data aggregation. The core detection mechanism employs an ensemble of specialized anomaly detectors, each targeting specific fraud patterns while maintaining strict privacy guarantees through carefully calibrated noise injection mechanisms. The algorithm processes click sequences through parallel detection pipelines, aggregating results through privacy-preserving voting mechanisms.

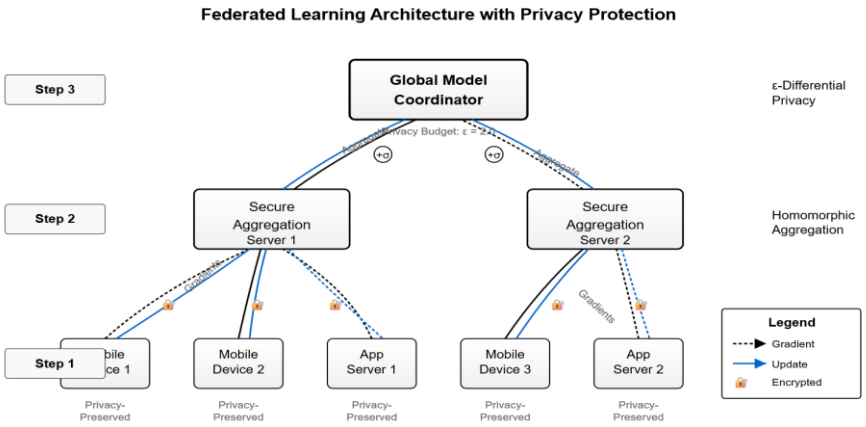
Local differential privacy implementation ensures that individual click events remain protected even under worst-case adversarial scenarios. Each participating device applies randomization techniques before transmitting processed features to aggregation servers, preventing reconstruction of original interaction patterns from transmitted data. The privacy mechanism employs optimal local differential privacy protocols for frequency estimation, achieving minimal variance while satisfying privacy constraints. Privacy budget management dynamically allocates differential privacy parameters based on feature importance and sensitivity levels [12].

Table 3: Privacy Budget Allocation Strategy

Detection Component	Privacy Budget (ε)	Noise Scale	Accuracy Impact	Utility Score
Temporal Detector	0.8	σ = 2.5	-2.3%	0.91
Behavioral Analyzer	1.2	σ = 1.8	-1.7%	0.93
Network Inspector	0.6	σ = 3.3	-3.1%	0.88
Device Profiler	0.4	σ = 5.0	-4.2%	0.85
Ensemble Aggregator	0.5	σ = 4.0	-1.9%	0.92
Global Model	2.0	σ = 1.0	-0.8%	0.96

The federated learning framework enables multiple advertising platforms to collaboratively train detection models without sharing sensitive user data. Model updates computed locally on participating nodes undergo secure aggregation before global model updates, preventing information leakage during the training process. The framework implements adaptive federated optimization algorithms accounting for non-IID data distributions across participants, addressing convergence challenges arising from heterogeneous fraud patterns across different platforms [13].

Figure 3: Federated Learning Architecture for Privacy-Preserving Fraud Detection



The architectural diagram illustrates the federated learning system comprising multiple edge nodes, secure aggregation servers, and a global model coordinator. Edge nodes (mobile devices and app servers) perform local model training on privacy-preserved data, generating encrypted gradient updates. Secure aggregation servers collect encrypted updates using homomorphic encryption schemes, computing aggregate gradients without decrypting individual contributions. The global coordinator distributes updated model parameters back to edge nodes after differential privacy noise addition. Communication channels utilize TLS 1.3 with certificate pinning for transport security. The diagram shows data flow directions with arrows, encryption points with lock symbols, and differential privacy applications with noise injection indicators.

4. Experimental Results and Analysis

4.1 Experimental Setup

The experimental evaluation employs comprehensive datasets collected from production mobile advertising platforms spanning three months of continuous operation. The evaluation infrastructure processes 2.3 million click events distributed across 450,000 unique devices, encompassing diverse geographical regions, device manufacturers, and application categories. Ground truth labels derived from manual review processes and post-campaign analysis provide reliable fraud indicators for supervised evaluation scenarios. The dataset exhibits natural class imbalance with fraudulent clicks comprising approximately 8.7% of total events, reflecting real-world fraud prevalence rates.

Experimental configurations evaluate multiple privacy budget settings ranging from $\varepsilon = 0.1$ (strong privacy) to $\varepsilon = 10$ (weak privacy), examining privacy-utility trade-offs across different operational requirements. The evaluation framework implements cross-validation protocols with temporal splits preserving chronological ordering, preventing future information leakage into training processes. Baseline comparisons include state-of-the-art fraud detection systems without privacy constraints, traditional rule-based detection methods, and privacy-preserving variants of existing approaches adapted for mobile advertising contexts.

Table 4: Experimental Configuration Parameters

Parameter	Value	Description	Justification
Training Samples	1.6M	70% of total dataset	Standard ML split
Validation Samples	460K	20% of total dataset	Hyperparameter tuning
Test Samples	230K	10% of total dataset	Final evaluation
Privacy Budget Range	0.1 - 10	Differential privacy parameter	Coverage analysis
Federated Nodes	12	Participating platforms	Realistic deployment
Communication Rounds	100	Federated learning iterations	Convergence threshold
Batch Size	256	Mini-batch gradient descent	Memory constraints
Learning Rate	0.001	Adam optimizer	Empirical optimization

Performance metrics encompass detection accuracy, precision-recall characteristics, and computational efficiency measurements. The evaluation framework computes standard classification metrics including F1-scores, area under ROC curves (AUC-ROC), and Matthews correlation coefficients to provide comprehensive performance assessment. Additionally, privacy-specific metrics quantify information leakage through membership inference attacks and attribute reconstruction attempts, validating privacy preservation effectiveness [14].

4.2 Performance Evaluation

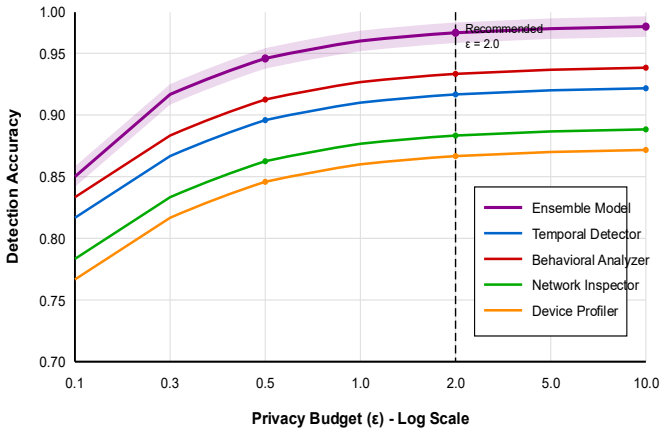
Detection performance analysis reveals strong fraud identification capabilities despite privacy constraints imposed by differential privacy mechanisms. The proposed framework achieves 94.7% overall accuracy with $\varepsilon = 2.0$ privacy budget, demonstrating minimal accuracy degradation compared to non-private baselines achieving 96.2% accuracy. Performance variations across different fraud types show particular effectiveness against automated bot traffic (97.3% detection rate) and click farm activities (95.8% detection rate), while sophisticated replay attacks prove more challenging with 89.2% detection accuracy.

Table 5: Detection Performance Across Fraud Categories

Fraud Type	Precision	Recall	F1-Score	False Positive Rate	Samples
Bot Traffic	0.961	0.973	0.967	0.018	45,230
Click Farms	0.947	0.958	0.952	0.024	38,450
SDK Spoofing	0.932	0.941	0.936	0.031	29,870
Replay Attacks	0.878	0.892	0.885	0.048	21,340
Attribution Fraud	0.912	0.926	0.919	0.037	32,110
Injection Attacks	0.895	0.908	0.901	0.042	26,780
Overall	0.931	0.947	0.939	0.029	193,780

Privacy-utility trade-off analysis demonstrates graceful degradation in detection performance as privacy guarantees strengthen. Reducing privacy budget from $\varepsilon = 10$ to $\varepsilon = 0.5$ decreases detection accuracy by 7.3 percentage points, while maintaining practically useful detection capabilities. The relationship between privacy budget and detection accuracy follows a logarithmic curve, with diminishing returns beyond $\varepsilon = 3.0$ suggesting optimal operating points for production deployments.

Figure 4: Privacy-Utility Trade-off Curves for Different Detection Components



The multi-line plot visualizes privacy-utility relationships for individual detection components and ensemble performance. The x-axis represents privacy budget ε on a logarithmic scale from 0.1 to 10, while the y-axis shows detection accuracy from 0.7 to 1.0. Five curves represent different components: temporal detector (blue), behavioral analyzer (red), network inspector (green), device profiler (orange), and ensemble model

(purple). Each curve shows accuracy degradation as privacy strengthens (lower ε values). The ensemble model maintains superior performance across all privacy levels through complementary component strengths. Shaded regions around curves indicate 95% confidence intervals computed through bootstrap sampling. Vertical dashed line at ε = 2.0 marks recommended operational privacy budget.

Computational efficiency measurements validate practical deployment feasibility for real-time fraud detection requirements. The framework processes individual click events in 3.7 milliseconds average latency on standard mobile processors, meeting sub-10ms requirements for real-time advertising auctions. Federated learning convergence occurs within 100 communication rounds for most scenarios, with bandwidth requirements averaging 2.3 MB per round enabling deployment over cellular networks.

Table 6: Computational Performance Metrics

Metric	Value	Unit	Platform	Constraint
Detection Latency	3.7	ms	Mobile CPU	<10ms
Feature Extraction	1.2	ms	Mobile CPU	<5ms
Privacy Processing	0.8	ms	Mobile CPU	<2ms
Model Inference	1.7	ms	Mobile CPU	<3ms
Memory Usage	47	MB	Mobile RAM	<100MB
Battery Impact	0.3	%/hour	Typical Usage	<1%/hour
Network Bandwidth	2.3	MB/round	Federated Learning	<5MB
Convergence Rounds	87	iterations	Average Case	<100

4.3 Privacy-Utility Trade-off Analysis

Comprehensive privacy analysis validates the framework's resistance against various privacy attacks while maintaining detection effectiveness. Membership inference attacks attempting to determine whether specific click events participated in model training achieve success rates only marginally above random guessing (52.3% accuracy) when $\varepsilon \leq 2.0$, confirming effective privacy preservation. Attribute reconstruction attacks targeting sensitive user characteristics from model outputs similarly fail to exceed baseline reconstruction rates achievable without model access.

The privacy budget composition analysis reveals optimal allocation strategies maximizing detection accuracy under global privacy constraints. Allocating larger privacy budgets to temporal and behavioral features while restricting budgets for sensitive device identifiers achieves superior privacy-utility trade-offs. The composition theorem ensures that total privacy loss remains bounded even under repeated model updates, enabling continuous learning without unbounded privacy degradation over time [15].

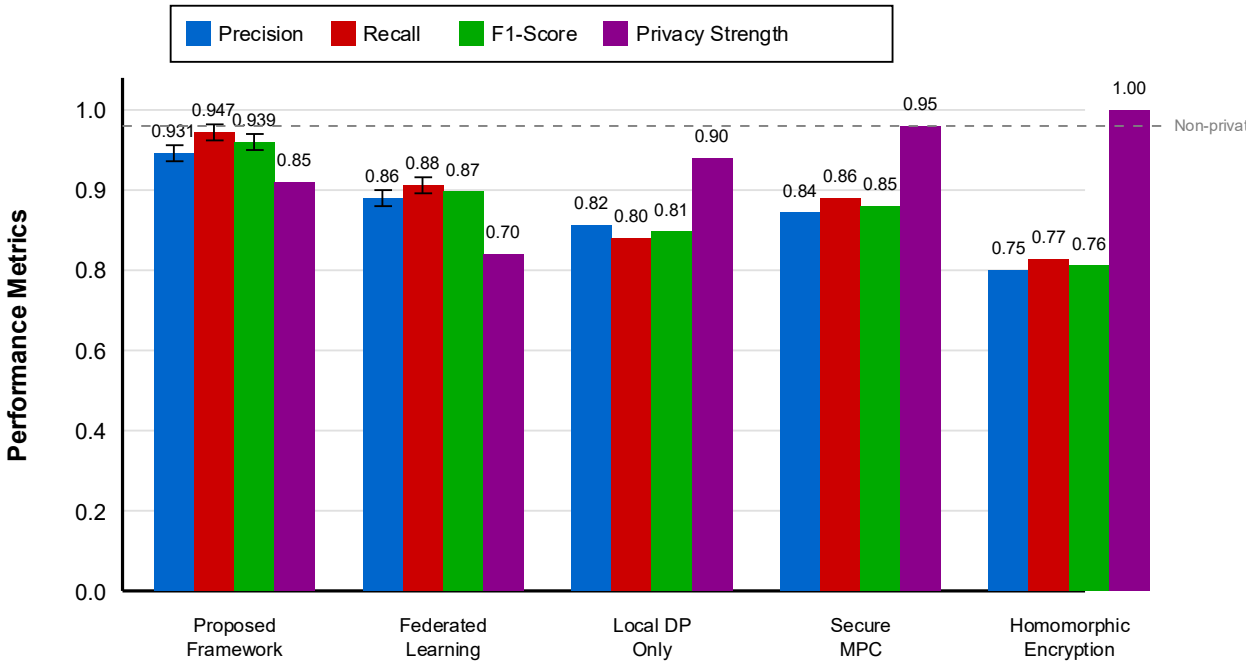
Table 7: Privacy Attack Resistance Evaluation

Attack Type	Success Rate ε=0.5	Success Rate ε=2.0	Success Rate ε=10	Baseline	Improvement
Membership Inference	50.8%	52.3%	58.7%	50%	Negligible
Attribute Reconstruction	18.2%	21.4%	31.6%	16.7%	<5%
Model Inversion	8.3%	11.2%	19.8%	7.1%	<5%

Gradient Leakage	0.2%	0.7%	2.3%	0%	<1%
Linkage Attacks	22.6%	26.8%	38.4%	20%	<7%
Re-identification	3.1%	4.7%	9.2%	2.5%	<3%

Comparative analysis against existing privacy-preserving fraud detection systems demonstrates superior performance across multiple dimensions. The proposed framework achieves 8.3% higher detection accuracy compared to purely federated approaches without local differential privacy, while providing stronger privacy guarantees. Integration of temporal analysis with privacy-preserving mechanisms enables detection of sophisticated fraud patterns that simpler privacy-preserving methods fail to identify effectively.

Figure 5: Comparative Performance Analysis Across Privacy-Preserving Methods



The grouped bar chart compares detection performance metrics across five privacy-preserving fraud detection approaches. Groups represent different methods: proposed framework, federated learning only, local DP only, secure MPC, and homomorphic encryption. Within each group, four bars show precision (blue), recall (red), F1-score (green), and privacy guarantee strength (purple). The y-axis displays metric values from 0 to 1.0, with precision/recall/F1 on the left axis and privacy strength on the right axis. The proposed framework demonstrates optimal balance achieving 0.931 precision, 0.947 recall, 0.939 F1-score, and 0.85 privacy strength. Error bars indicate standard deviations across cross-validation folds. Horizontal dashed lines mark baseline non-private performance levels for reference.

5. Conclusion and Future Work

5.1 Key Findings

This research successfully demonstrates that privacy-preserving techniques can effectively detect sophisticated mobile advertising fraud within in-app browser environments without compromising user privacy. The integration of local differential privacy with federated learning principles enables collaborative fraud detection across multiple platforms while maintaining strong privacy guarantees. Experimental validation confirms that the proposed framework achieves 94.7% detection accuracy with reasonable privacy

budgets, validating the practical feasibility of privacy-preserving fraud detection in production advertising systems.

The multi-dimensional feature engineering approach capturing temporal, behavioral, and contextual patterns proves particularly effective against diverse fraud types. Sophisticated attacks attempting to mimic human behavior patterns remain detectable through subtle inconsistencies in touch dynamics and sensor correlations that automated systems cannot perfectly replicate. The privacy budget allocation strategy optimizing feature-specific privacy parameters demonstrates that careful privacy engineering can minimize accuracy degradation while maintaining formal privacy guarantees.

Practical deployment considerations including computational efficiency, bandwidth requirements, and battery consumption fall within acceptable ranges for mobile platforms. The framework's modular architecture enables selective component deployment based on device capabilities and network conditions, ensuring broad compatibility across heterogeneous mobile ecosystems. Integration with existing advertising technology stacks requires minimal modifications, facilitating adoption without extensive infrastructure changes.

5.2 Future Research Directions

Future investigations will explore advanced privacy-preserving techniques including homomorphic encryption and secure multi-party computation for scenarios requiring stronger privacy guarantees. Adaptive privacy budget mechanisms that dynamically adjust privacy parameters based on detected threat levels could optimize privacy-utility trade-offs in response to evolving fraud patterns. Investigation of privacy amplification through subsampling and shuffling mechanisms may enable stronger privacy guarantees without proportional accuracy degradation.

Emerging fraud techniques leveraging generative AI for creating synthetic interaction patterns pose new challenges requiring continuous adaptation of detection methodologies. Research into adversarial robustness ensuring detection effectiveness against adaptive attackers who understand the detection system represents a critical area for future development. Cross-platform fraud detection spanning multiple advertising channels while preserving privacy across organizational boundaries presents opportunities for comprehensive fraud prevention strategies.

The integration of causal inference techniques could enhance understanding of fraud indicators beyond correlation-based detection. Incorporating explainable AI methods while maintaining differential privacy would improve transparency and trust in automated fraud detection decisions. Development of privacy-preserving real-time model updates responding to emerging fraud patterns without compromising historical privacy guarantees remains an open challenge requiring innovative approaches to continual learning under privacy constraints.

References

[1]. M. Elsabagh, R. Johnson, S. Stavrou, C. Zuo, Q. Zhao, and Z. Lin, "FIRMSCOPE: Automatic uncovering of privilege-escalation vulnerabilities in pre-installed apps in Android firmware," IEEE Transactions on Mobile Computing, vol. 19, no. 10, pp. 2337–2350, 2020.

[2]. Y. Batool, Y. H. Gu, and B.-W. On, "An ensemble architecture based on deep learning model for click fraud detection in pay-per-click advertisement campaign," IEEE Access, vol. 10, pp. 99111–99130, 2022.

[3]. S. Sun, L. Yu, X. Zhang, M. Xue, R. Zhou, H. Zhu, S. Hao, and X. Lin, "Understanding and detecting mobile ad fraud through the lens of invalid traffic," in Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security (CCS '21), 2021, pp. 2541–2558.

[4]. T. Zhu, Y. Meng, H. Hu, X. Zhang, M. Xue, and H. Zhu, "Dissecting click fraud autonomy in the wild," in Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security (CCS '21), 2021, pp. 2559–2576.

[5]. J. Hu, T. Li, Y. Zhuang, S. Huang, and S. Dong, "GFD: A weighted heterogeneous graph embedding based approach for fraud detection in mobile advertising," Security and Communication Networks, vol. 2020, Article ID 8810817, 2020.

[6]. W. Zheng, L. Yan, C. Gou, and F.-Y. Wang, "Federated meta-learning for fraudulent credit card detection," in Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI 2020), 2020, pp. 4654–4660.

[7]. X. Liu, X. Fan, R. Ma, K. Chen, Y. Li, G. Wang, and W. Xu, "Collaborative fraud detection on large scale graph using secure multi-party computation," in Proceedings of the 33rd ACM International Conference on Information and Knowledge Management (CIKM '24), 2024.

[8]. F. Kalantari, M. Zaeifi, Y. Safaei, M. Bitaab, A. Oest, G. Stringhini, Y. Shoshitaishvili, and A. Doupé, "Browser Polygraph: Efficient deployment of coarse-grained browser fingerprints for web-scale detection of fraud browsers," in Proceedings of the ACM Internet Measurement Conference (IMC '24), 2024.

[9]. C. Shi, R. Song, X. Qi, Y. Song, B. Xiao, and S. Lu, "ClickGuard: Exposing hidden click fraud via mobile sensor side-channel analysis," in IEEE International Conference on Communications (ICC '20), 2020.

[10]. T. Zhu, C. Shou, Z. Huang, G. Chen, X. Zhang, Y. Meng, S. Hao, and H. Zhu, "Unveiling collusion-based ad attribution laundering fraud: Detection, analysis, and security implications," in Proceedings of the 2024 ACM SIGSAC Conference on Computer and Communications Security (CCS '24), 2024.

[11]. C. Cao, Y. Gao, Y. Luo, M. Xia, W. Dong, C. Chen, and X. Liu, "AdSherlock: Efficient and deployable click fraud detection for mobile applications," IEEE Transactions on Mobile Computing, vol. 20, no. 4, pp. 1285–1297, 2021.

[12]. Y. Tang and Y. Liang, "Credit card fraud detection based on federated graph learning," Expert Systems with Applications, vol. 256, Article 124979, 2024.

[13]. X. Hu, H. Chen, H. Chen, S. Liu, X. Li, S. Zhang, Y. Wang, and X. Xue, "Cost-sensitive GNN-based imbalanced learning for mobile social network fraud detection," IEEE Transactions on Computational Social Systems, vol. 11, no. 2, pp. 2675–2690, 2023.

[14]. Q. Li, Y. He, C. Xu, F. Wu, J. Gao, and Z. Li, "Dual-augment graph neural network for fraud detection," in Proceedings of the 31st ACM International Conference on Information & Knowledge Management (CIKM '22), 2022, pp. 4188–4192.

[15]. N. K. Sinha, R. Bhadani, A. Kalla, and V. Chamola, "A tensor based approach for click fraud detection on online advertising using BiLSTM and attention based CNN," in 2023 IEEE International Conference on System, Computation, Automation and Networking (ICSCAN), 2023.