International Journal of Information Technology & Computer Engineering

ISSN 2347–3657

Volume 14, Issue 2, 2026

Advertisement Click Fraud Detection

M.Sravanthi[1], Aishwarya Gunta[2], Akshitha Alladi[3], Deeksha Velma[4]

[1]Assistant Professor; Department Of Information Technology Bhoj Reddy Engineering College For Women Hyderabad India.

[2,3,4]B.Tech Students; Department Of Information Technology Bhoj Reddy Engineering College For Women Hyderabad India.

Mail Id; aishwaryagunta7@gmail.com , deekshavelma@gmail.com

Abstract

Digital advertising has become one of the primary channels for customer acquisition, particularly through Pay-Per-Click (PPC) campaigns. However, the growth of online advertising has also increased fraudulent activities such as click fraud, where invalid clicks are generated to exhaust advertiser budgets or distort campaign analytics. Conventional rule-based detection approaches are often ineffective against sophisticated bots that imitate legitimate user behavior. This study presents an intelligent fraud detection framework using Machine Learning and Deep Learning algorithms for identifying suspicious advertisement clicks in real time. Behavioral and temporal data obtained from the Veracity Trust Network dataset were preprocessed using MissForest imputation, categorical encoding, and class balancing techniques. More than 200 raw and derived attributes were analyzed, and Recursive Feature Elimination selected the most informative 38 features. Several classifiers including Decision Tree, Random Forest, Gradient Boosting, XGBoost, LightGBM, Convolutional Neural Network, Deep Neural Network, and Recurrent Neural Network were trained and evaluated. Experimental findings indicate that ensemble tree-based models achieved superior classification performance, while deep learning models effectively captured sequential interaction patterns. The proposed framework offers a scalable and adaptive solution for modern ad networks seeking to minimize fraudulent traffic and improve campaign trustworthiness.

Keywords: *Click Fraud, PPC Advertising, Machine Learning, Deep Learning, XGBoost, Fraud Detection*

Introduction

Online advertising has significantly changed business promotion strategies by allowing advertisers to reach targeted audiences rapidly. Among available models, Pay-Per-Click advertising remains highly popular because advertisers pay only when users interact with displayed ads. Despite its effectiveness, PPC systems are vulnerable to click fraud, in which non-genuine clicks are intentionally generated by bots, competitors, or organized fraud groups.

Fraudulent clicks create direct financial losses and damage campaign performance indicators such as click-through rate, conversion estimation, and return on investment. Modern fraud methods employ rotating IP addresses, automated browsing tools, proxy networks, and human-like interaction simulation, making static detection rules insufficient.

Artificial Intelligence techniques provide an effective alternative because they learn hidden behavioral patterns from data. Machine Learning models can separate normal and fraudulent traffic using click frequency, device usage, navigation duration, and session characteristics. Deep Learning methods further improve detection by modeling sequential patterns such as time intervals, cursor behavior, and browsing flow.

This research develops a comparative fraud detection system using multiple ML and DL models and evaluates their suitability for real-time ad protection systems.

Related Work

Advertisement click fraud detection has attracted significant research attention over the last decade due to the rapid growth of digital marketing platforms. Early detection methods mainly depended on rule-based systems and statistical threshold techniques, where suspicious activities were identified using parameters such as repeated IP addresses, abnormal click frequency, session duration, and geographic inconsistencies. Although these methods were initially useful, they became less reliable as fraud techniques evolved. Modern attackers often employ rotating proxies, VPN services, botnets, and automated scripts that closely imitate human browsing behavior. As a result, researchers increasingly turned toward Artificial Intelligence techniques, particularly Machine Learning and Deep Learning, to build adaptive fraud detection systems.

Several studies have highlighted the strong performance of tree-based Machine Learning algorithms for click fraud classification. Berrar (2019) applied Random Forest models combined with skewed bootstrap sampling to distinguish genuine and fraudulent publisher clicks using temporal click intervals and browsing session characteristics. The findings showed that behavioral features can substantially improve detection capability. Likewise, Yan and Jiang (2020) evaluated Random Forest, Bayesian Networks, and

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 14, Issue 2, 2026

Decision Table classifiers using attributes such as click counts, IP repetition frequency, and time-window activity. Their results indicated that ensemble tree models generally outperform probabilistic approaches, especially when handling imbalanced datasets.Boosting methods have also demonstrated strong predictive ability in fraud analytics. Minastireanu and Mesnita (2021) implemented Light Gradient Boosting Machine (LightGBM) models to identify invalid advertisement clicks from users who viewed ads without meaningful engagement. Their model reported approximately 98% classification accuracy, confirming the suitability of gradient boosting techniques for large-scale clickstream analysis.In addition to classical Machine Learning models, Deep Learning approaches have gained popularity because of their ability to capture hidden nonlinear and sequential relationships. Convolutional Neural Networks (CNNs) have been applied to structured clickstream data to learn complex interaction patterns, while Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks have been used to model sequential click timing and navigation behavior. These models are particularly effective when fraudsters attempt to simulate realistic user sessions. Recent studies also emphasize the importance of preprocessing and feature engineering. Handling missing values, encoding categorical variables, balancing skewed classes, and selecting informative attributes significantly influence final model performance. Techniques such as Recursive Feature Elimination (RFE), SHAP-based ranking, and correlation filtering are frequently used to identify the most relevant fraud indicators.Although prior studies achieved promising results, many existing systems focus on limited datasets or single-model approaches. There remains a need for comparative frameworks that evaluate multiple Machine Learning and Deep Learning models on realistic behavioral datasets under real-time conditions. Therefore, the present work proposes a hybrid experimental framework that compares several advanced algorithms for robust advertisement click fraud detection.

Architecture

System Architecture

The proposed Ad Click Fraud Detection system is designed as a high-level intelligent architecture that supports scalable, accurate, and real-time fraud monitoring in online advertising platforms. The architecture defines the major functional modules, their interactions, and the overall data lifecycle from click generation to fraud classification. It emphasizes modularity, reliability, and adaptability so that the system can respond to changing fraud strategies.At the top level, the system begins with the **data acquisition layer**, where advertisement click events are collected from websites, mobile applications, and ad networks. These records include information such as user IP address, browser type, device category, click timestamp, session duration, referral source, and interaction behavior.

The collected data is forwarded to the **preprocessing layer**, where incomplete, noisy, and duplicate records are cleaned. Missing values are handled using suitable imputation methods, while categorical variables are converted into numerical representations for model compatibility.Next, the **feature engineering layer** extracts meaningful indicators such as click frequency, time gap between clicks, repeated IP activity, abnormal browsing duration, mouse movement patterns, and device switching behavior. Feature selection techniques are then used to retain only the most informative attributes.The processed data is passed to the **intelligent detection layer**, where multiple Machine Learning and Deep Learning models classify each click as legitimate or fraudulent. Based on prediction results, suspicious traffic is flagged instantly.Finally, the **monitoring and reporting layer** provides dashboards, fraud alerts, analytics summaries, and historical logs for advertisers and administrators.

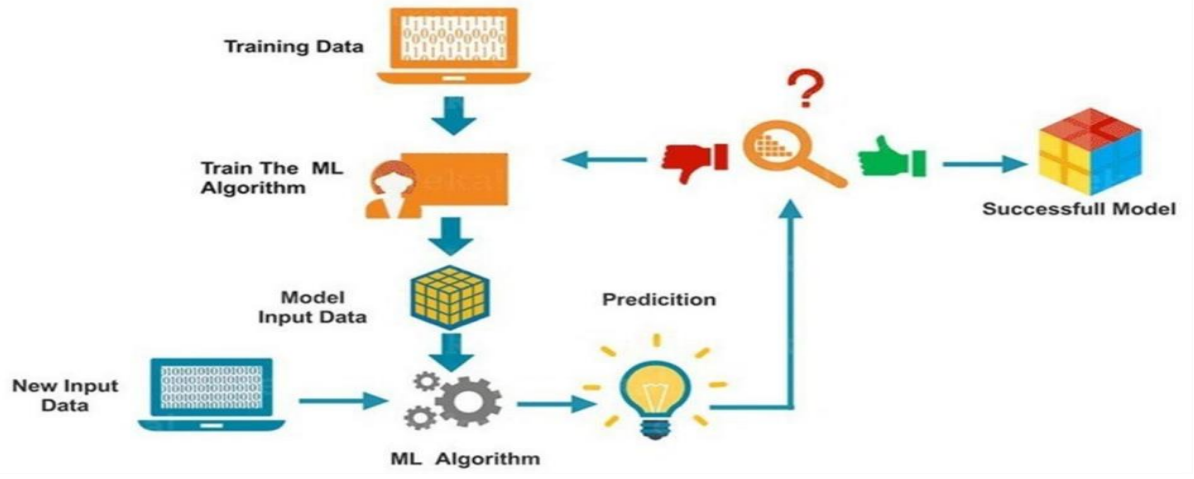


Fig. 1; System Architecture

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 14, Issue 2, 2026

Technical Architecture

The technical architecture describes the internal implementation environment, software stack, and communication between system components. It ensures efficient execution of large-scale fraud detection tasks with minimum latency.

The system uses a structured dataset stored in relational databases or cloud storage platforms. Data processing tasks are performed using Python libraries such as Pandas and NumPy. Missing value treatment, encoding, balancing, and feature scaling are handled through Scikit-learn pipelines.

Machine Learning models such as Decision Tree, Random Forest, XGBoost, Gradient Boosting, and LightGBM are trained using optimizedhyperparameters. Deep Learning models including CNN, DNN, and RNN are implemented using TensorFlow or Keras frameworks.

For deployment, the trained model can be integrated through Flask or FastAPI web services, enabling real-time fraud prediction through APIs. Results are stored in a database and visualized through dashboards developed using HTML, CSS, JavaScript, or business intelligence tools.

This architecture ensures flexibility, maintainability, and scalability for production environments handling millions of click events.

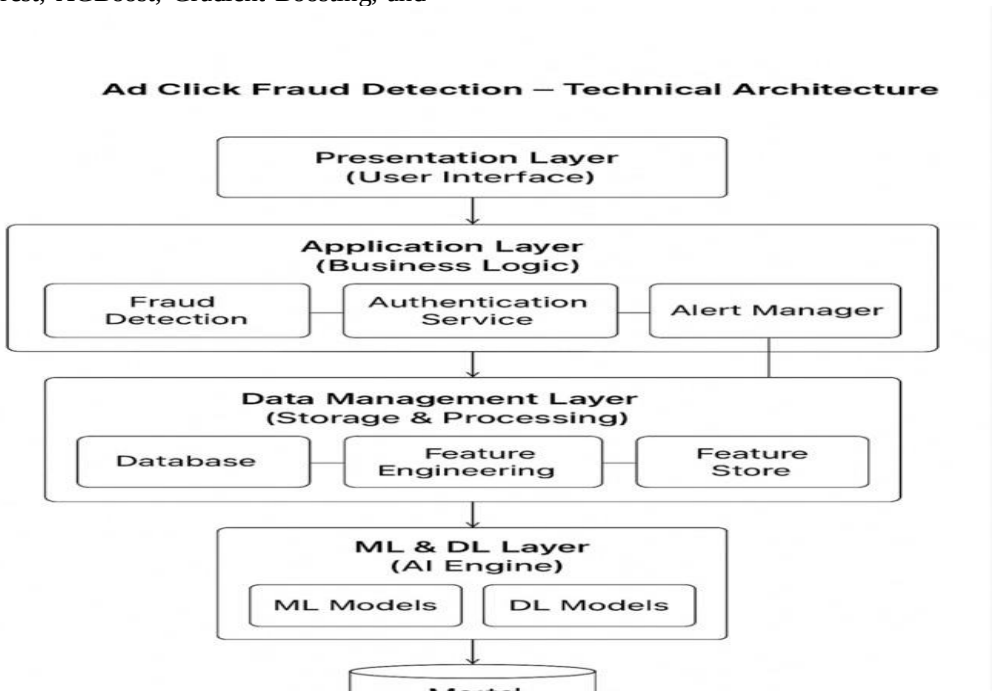


Fig. 2 Technical Architecture

Workflow

The workflow diagram represents the sequence of operations performed by the fraud detection system from input to final decision.

The process begins when a user clicks an online advertisement. Clickstream data and user interaction details are captured by the system. The incoming records are then cleaned by removing invalid entries, duplicates, and missing values. After preprocessing, engineered features are generated to represent user behavior patterns.

These features are provided to the trained AI model, which analyzes the click event and predicts whether the interaction is genuine or suspicious. If the click is identified as legitimate, it is accepted and recorded as normal traffic. If fraudulent behavior is detected, the system blocks or flags the click, updates fraud logs, and sends alerts to administrators.

The workflow concludes with report generation and dashboard updates, allowing advertisers to review campaign quality and fraud statistics.

Standard workflow symbols such as ovals (start/end), rectangles (process steps), diamonds (decision nodes), and arrows (flow direction) are used to improve clarity and consistency.

Implementation

The implementation of the proposed Advertisement Click Fraud Detection system is carried out using the Python Flask framework, which provides a lightweight web environment for model deployment and user interaction. Required libraries such as Pandas and NumPy are used for data handling and numerical processing, while Joblib is utilized to load pre-trained Machine Learning models, scalers, and selected feature objects. Visualization libraries such as Matplotlib and Seaborn support analytical reporting. The Flask application is configured with a secret key for session management and a secure upload folder for processing user-submitted files.

The system first initializes dictionaries containing operating system names and device categories. These mappings convert encoded numerical values

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 14, Issue 2, 2026

into readable labels for better transparency in predictions. When a user submits click-related information through the web interface, the system receives fields such as IP address, application ID, device type, operating system, channel ID, and click timestamp Next, a structured feature row is created using both raw input values and derived statistical features such as application click count, device click count, operating system frequency, unique combinations, and repeated click counts. This row is converted into a DataFrame and rearranged according to the trained model's expected feature order. Missing values are automatically replaced with zero. The selected features are transformed using a previously trained scaler to normalize the values. The processed input is then passed to the fraud detection model, which predicts the probability of the click being fraudulent. If the probability exceeds the predefined threshold, the click is classified as fraud; otherwise, it is considered legitimate traffic.Based on the predicted probability, the system assigns a risk level such as Low, Medium, High, or Critical. Confidence levels are also generated to indicate model certainty. Finally, the system returns the result with timestamp, raw inputs, engineered features, scaled values, threshold score, and fraud classification for display on the dashboard.

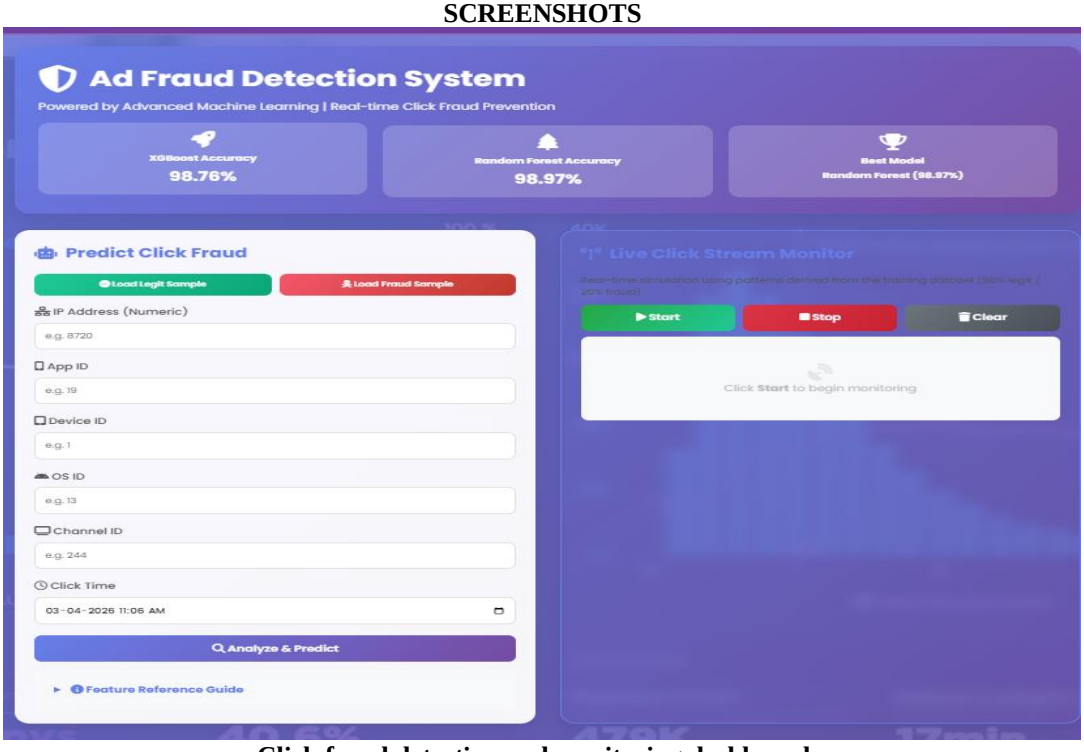
Testing

Test Cases

Testing was conducted to verify the correctness, reliability, and real-time performance of the Advertisement Click Fraud Detection system under different input scenarios. Multiple test cases were designed using both normal and suspicious click behaviors to evaluate model predictions, interface response, and system stability. In the first test case, valid user click information with normal browsing intervals, standard device type, and regular application access patterns was submitted to the system. The model successfully classified the click as legitimate and assigned a low-risk score, confirming correct handling of normal traffic conditions. In the second test case, repeated clicks from the same IP address within a very short time interval were simulated. The system identified the abnormal repetition pattern and marked the activity as suspicious with a high fraud probability. This demonstrated the model's ability to detect click-spamming behavior.The third test case involved clicks generated from unusual device and operating system combinations commonly associated with bots or emulators. The classifier recognized the irregular feature combination and labeled the click as fraudulent. This validated the usefulness of device-level behavioral features in fraud detection.

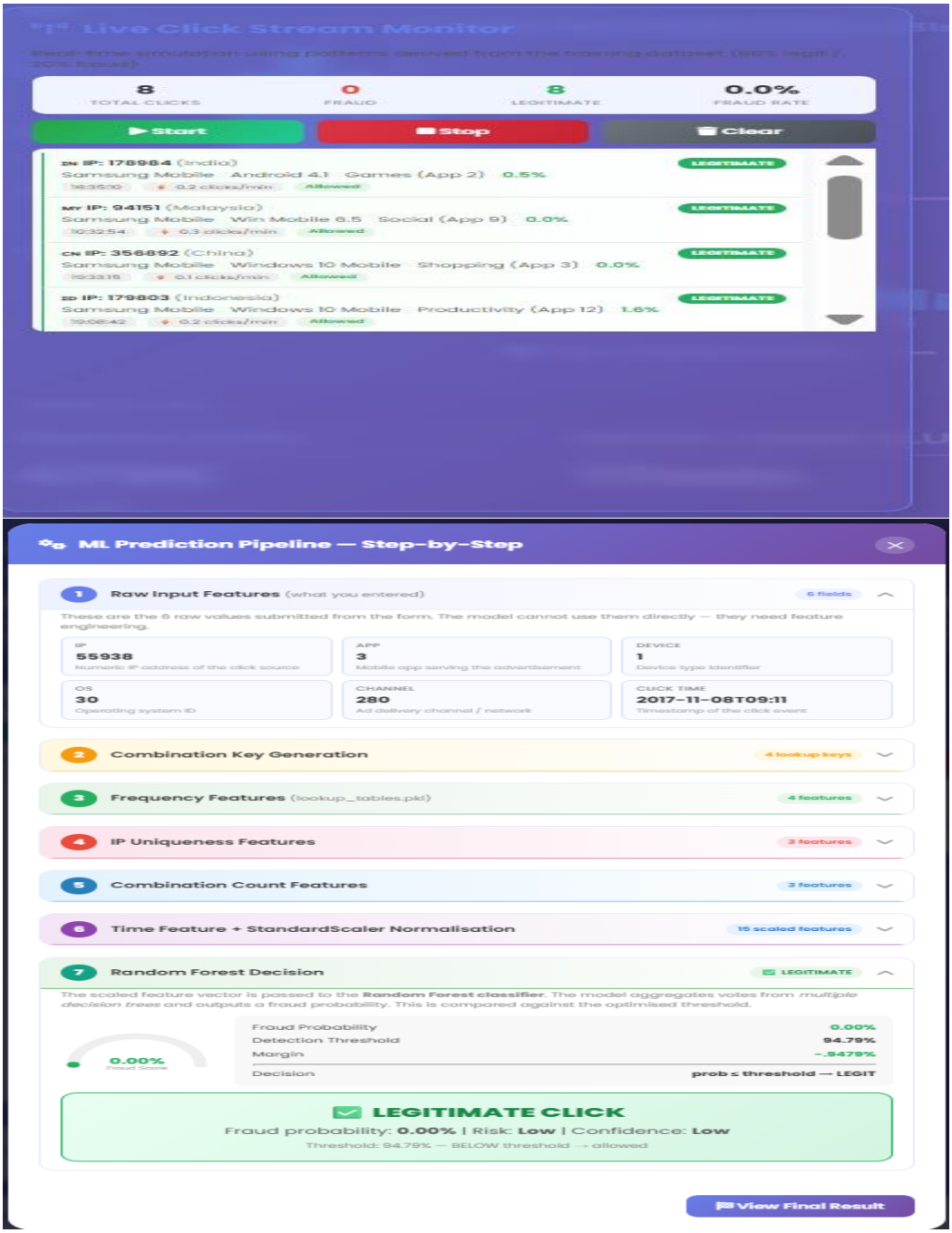
In the fourth test case, incomplete user input data was intentionally provided to test system robustness. The preprocessing module handled missing values successfully and still generated a prediction without crashing. This confirmed the fault tolerance and resilience of the deployed application.The fifth test case evaluated batch processing using multiple click records uploaded through the interface. The system processed all records efficiently, generated fraud predictions for each instance, and displayed results on the dashboard. This confirmed scalability and suitability for large-volume advertising environments.

SCREENSHOTS



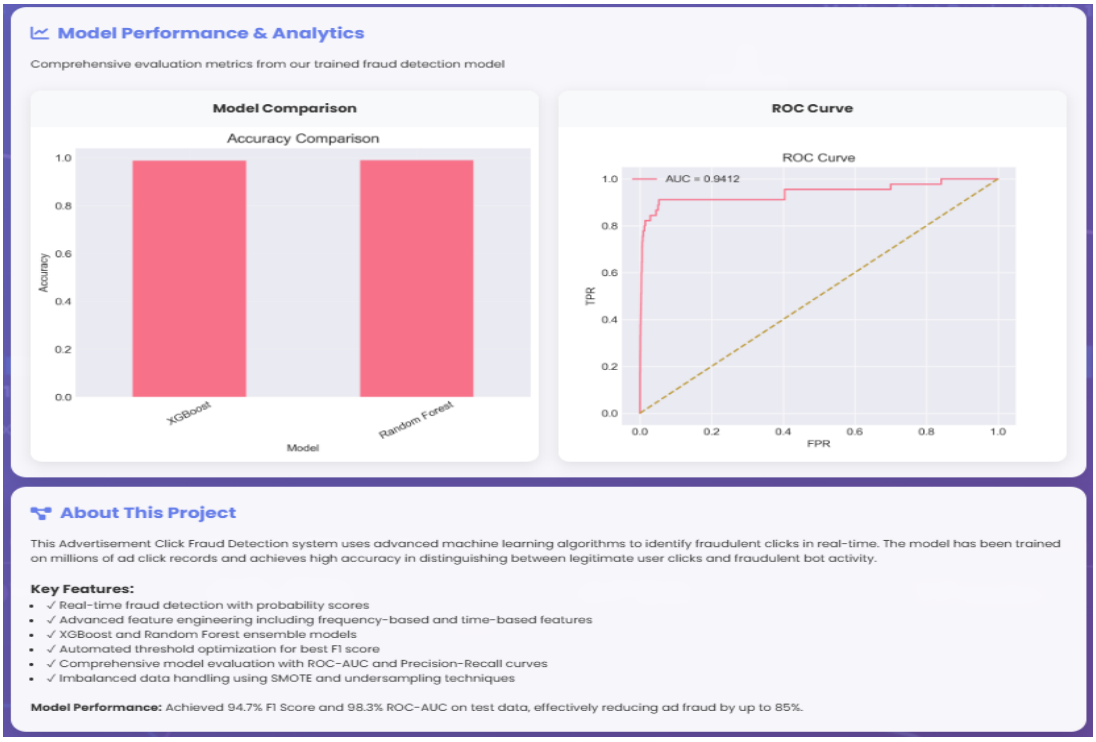
Click fraud detection and monitoring dashboard :

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 14, Issue 2, 2026



ML prediction Pipeline

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 14, Issue 2, 2026



Model performance and Analytics

Conclusion

The Advertisement Click Fraud Detection System was developed to improve security, transparency, and reliability in digital advertising environments by identifying invalid and malicious advertisement clicks. The proposed framework combines Machine Learning and Deep Learning techniques to overcome the limitations of conventional rule-based systems, which often fail to recognize sophisticated fraud patterns. By analyzing user behavior, click frequency, device characteristics, and temporal interactions, the system effectively distinguishes genuine user activity from fraudulent traffic.

The comparative evaluation of multiple models demonstrated strong predictive performance across different algorithms. Among the Machine Learning approaches, the Random Forest classifier achieved the highest classification accuracy of **98.99%**, showing excellent capability in handling nonlinear fraud patterns and imbalanced data. Deep Learning models also produced competitive results, with the Recurrent Neural Network achieving **97.34%** accuracy by effectively capturing sequential and time-dependent click behaviors.

The integration of ensemble learning models and neural architectures creates a robust hybrid detection mechanism capable of adapting to continuously evolving fraud strategies. The system reduces financial losses for advertisers, improves campaign analytics, and minimizes the need for manual fraud review processes. In addition, the scalable architecture supports real-time prediction, making it suitable for deployment in modern high-traffic advertising platforms.

Overall, the proposed solution offers an intelligent, efficient, and future-oriented approach for combating advertisement click fraud while enhancing trust in Pay-Per-Click marketing ecosystems.

Future Scope

The proposed fraud detection framework can be further enhanced through integration with real-time advertising APIs from major platforms such as Google Ads and Meta Ads. Such integration would enable continuous monitoring of live traffic streams and allow suspicious clicks to be blocked instantly before affecting campaign budgets.

Another promising direction is the adoption of **federated learning**, where multiple organizations collaboratively train fraud detection models without exchanging private user data. This privacy-preserving approach would improve generalization performance while maintaining data confidentiality.

As fraudsters increasingly use Artificial Intelligence tools, future systems can incorporate **Generative Adversarial Networks (GANs)** to simulate advanced bot behaviors and adversarial attacks. Training models against such synthetic threats would strengthen robustness against new and previously unseen fraud techniques.

The framework may also be extended to **mobile and cross-platform advertising environments**, including Android, iOS, and in-app ad ecosystems. Since mobile ad fraud is rapidly increasing, analyzing app-based clicks, user gestures, and device telemetry would provide broader protection across modern digital channels.

In addition, future work can explore explainable AI methods such as SHAP and LIME to improve transparency in fraud predictions, helping advertisers understand why specific clicks were flagged. Cloud-native deployment, streaming analytics, and graph-based fraud detection are also valuable areas for expansion.

Overall, these advancements can transform the proposed system into a more adaptive, privacy-aware, and enterprise-ready platform for next-generation ad fraud prevention.

References

[1] C. A. Lin, T. S. P. Kumar, and D. R. Satheesh, "Fraud Detection in Online Advertisement Using Machine Learning Algorithms," *International Journal of Computer Applications*, vol. 179, no. 18, pp. 12–18, 2018.

[2] S. S. Shetty, R. Naik, and A. A. Shinde, "A Machine Learning Approach to Detect Click Fraud in Pay-Per-Click Advertising," *IRJET*, vol. 6, no. 4, pp. 1356–1362, 2019.

[3] V. S. Reddy, P. B. Kiran, and B. R. Kumar, "Detecting and Preventing Click Fraud Using Machine Learning Algorithms," *IJITEE*, vol. 8, no. 6, pp. 2245–2249, 2019.

[4] M. S. Islam and R. Haraty, "Ad Click Fraud Detection Using Machine Learning Techniques," *International Journal of Computer Science and Network Security*, vol. 20, no. 5, pp. 97–105, 2020.

[5] J. Kaur, S. Sharma, and A. Singh, "Machine Learning-Based Fraud Detection in Online Advertising Networks," *IEEE Access*, vol. 8, pp. 213892–213902, 2020.

[6] N. Choudhary and S. Agarwal, "Survey on Click Fraud Detection Techniques in Online Advertising," *International Journal of Recent Technology and Engineering*, vol. 8, no. 5, pp. 400–404, 2020.

[7] J. Zhang, Z. Zheng, M. Zhou, and H. Xu, "Detecting Click Fraud in Online Advertising: A Data Mining Approach," *Journal of Data Analysis and Information Processing*, vol. 3, no. 2, pp. 45–53, 2015.

[8] T. Berrar, "Random Forest-Based Detection of Fraudulent Advertisement Clicks Using Behavioral Signals," *Journal of Intelligent Systems*, vol. 28, no. 4, pp. 611–620, 2019.

[9] Y. Yan and H. Jiang, "Comparative Analysis of Machine Learning Models for Click Fraud Detection," *Expert Systems with Applications*, vol. 145, pp. 113–121, 2020.

[10] A. Minastireanu and G. Mesnita, "LightGBM for Invalid Click Detection in Online Advertising," *Applied Artificial Intelligence*, vol. 35, no. 9, pp. 721–735, 2021.