International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435 (Online) Volume 15, Issue 05, May 2026)

AI-Based Click Fraud Detection System

N. Aravindhan[1], R. Madhan kumar[2]

[1] Assistant Professor, [2] II MCA, Department of Master of Computer Applications, Er. Perumal Manimekalai College of Engineering, Hosur, Tamil Nadu, India.

***Abstract* -- Click fraud is a deceptive activity in which illegitimate clicks are generated on online advertisements through bots, automated programs, or hired individuals. These artificial clicks distort marketing analytics, drain advertising budgets, and reduce the overall effectiveness of digital campaigns. Existing click-monitoring systems rely heavily on static rules, IP filtering, and manual analysis, which struggle to detect complex or evolving fraud patterns. These traditional methods often fail when fraudsters use sophisticated behaviors such as random timing, varied devices, or distributed networks, leading to high false positives and missed detections. To overcome these limitations, this paper proposes a Multilayer Perceptron (MLP)-based click fraud detection system that leverages deep learning to analyze user interaction behavior. The system processes diverse features, including device metadata, session activity, click frequency, and contextual ad interaction patterns, to learn hidden relationships within the data. By utilizing a multilayer neural architecture, the model detects subtle anomalies and generates a fraud probability score for each click. The performance of the system is evaluated using accuracy, precision, recall, and F1-score metrics to ensure robust fraud detection. The proposed approach provides an adaptive and intelligent solution that helps advertisers minimize financial losses, enhance monitoring accuracy, and maintain the credibility of digital advertising platforms. Furthermore, the system is scalable for high-traffic environments and supports continuous retraining, enabling it to adapt to emerging fraud patterns in dynamic online ecosystems.**

***Keywords* -- Click Fraud, Machine Learning, Deep Learning, Cybersecurity, Online Advertising, AI Detection**

I. Introduction

The rapid advancement of digital marketing has transformed how businesses interact with consumers, with online advertising becoming a primary channel for promotion and revenue generation. Among various advertising strategies, the pay-per-click (PPC) model has gained widespread adoption due to its cost-effectiveness and measurable performance. In this model, advertisers are charged only when users click on their advertisements, making it highly attractive for targeted marketing campaigns.

However, the PPC model is highly vulnerable to click fraud, a malicious activity where illegitimate clicks are generated intentionally to exhaust advertising budgets or manipulate campaign performance metrics. These fraudulent clicks are often produced using automated bots, distributed networks, or organized human click farms, making detection increasingly complex and challenging.

Traditional click fraud detection mechanisms rely on static rules, such as limiting the number of clicks per IP address or identifying abnormal traffic spikes. While effective against simple attacks, these methods fail to detect sophisticated fraud patterns that mimic legitimate user behavior. As attackers continue to evolve their techniques, there is a growing demand for intelligent and adaptive detection systems.

Artificial Intelligence (AI), particularly machine learning and deep learning, provides a promising solution by enabling systems to analyze large datasets and uncover hidden behavioral patterns. This paper presents an AI-based click fraud detection system that leverages advanced algorithms to improve detection accuracy, reduce false positives, and ensure scalability for real-world applications.

II. Literature Review

Click fraud detection has been widely researched, with various approaches proposed to address the problem. Early methods relied on rule-based systems that used predefined thresholds to detect suspicious activities. For example, a high number of clicks from a single IP address within a short duration would be flagged as fraudulent. Although these methods are simple and computationally efficient, they lack adaptability and fail to detect complex fraud behaviors.

With the emergence of machine learning, researchers began utilizing classification algorithms such as Decision Trees, Random Forests, Naïve Bayes, and Support Vector Machines (SVM). These models are capable of learning patterns from historical data and improving detection accuracy. Among these, Random Forest has demonstrated strong performance due to its robustness and ability to handle large datasets. However, traditional machine learning models often struggle to capture sequential and temporal patterns in clickstream data.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435 (Online) Volume 15, Issue 05, May 2026)

Recent advancements in deep learning have introduced neural networks such as Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks for fraud detection. These models can analyze complex user behavior and identify subtle anomalies that indicate fraudulent activity. Additionally, behavioral analysis techniques focusing on session duration, click intervals, and geographic inconsistencies have further improved detection performance.

Despite these advancements, challenges such as scalability, real-time detection, and adaptability to evolving fraud techniques remain unresolved, necessitating the development of more intelligent and scalable systems.

III. Problem Statment

Click fraud poses a critical challenge to the online advertising industry, leading to significant financial losses and reduced trust among stakeholders. Existing detection systems often suffer from low accuracy, high false positive rates, and limited adaptability to evolving fraud patterns.

Moreover, many traditional systems are not designed to process large-scale data in real-time environments, making them inefficient for modern high-traffic platforms. These limitations highlight the need for an intelligent system that can accurately detect fraudulent clicks while maintaining scalability and efficiency.

The objective of this research is to develop a robust AI-based click fraud detection system that improves detection accuracy, minimizes false positives, and adapts to dynamic fraud behaviors.

IV. Proposed System

The proposed system is designed as a modular framework that integrates data collection, preprocessing, feature engineering, model training, and real-time detection. It utilizes a Multilayer Perceptron (MLP) model to analyze user behavior and detect fraudulent clicks.

The system collects clickstream data from various sources and stores it in a MySQL database. The data is then preprocessed to remove inconsistencies and extract meaningful features. These features are used to train the deep learning model.

Once trained, the model is deployed in a real-time detection environment where incoming clicks are analyzed and classified as legitimate or fraudulent. The results are presented through a Flask-based web interface, enabling users to monitor and analyze fraud detection outcomes efficiently.

This architecture ensures flexibility, scalability, and ease of integration with existing digital advertising platforms.

V. System Architecture

Fig. 1. System Architecture of AI-Based Click Fraud Detection System

The system architecture consists of multiple layers, each responsible for a specific function within the detection pipeline. The Data Collection Layer captures clickstream data such as IP address, timestamp, device type, and session activity.

The Data Preprocessing Layer cleans and transforms raw data by handling missing values, removing duplicates, and normalizing features. The Feature Engineering Layer extracts meaningful attributes such as click frequency, session duration, and geographic anomalies.

The processed data is passed to the Model Training Layer, where the MLP model is trained using labeled datasets. The trained model is deployed in the Detection Layer, which classifies incoming clicks in real time. Finally, the Visualization Layer displays results through a Flask web interface.

This layered architecture ensures efficient data flow and accurate fraud detection.

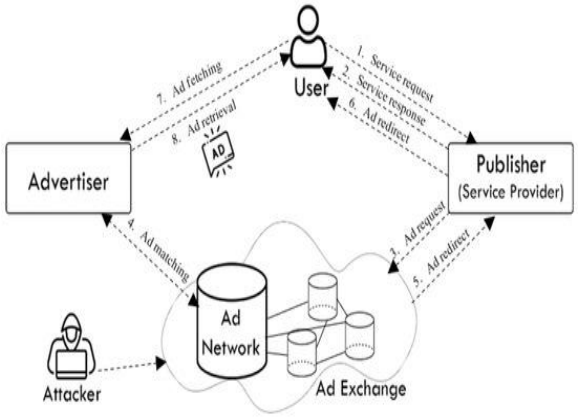
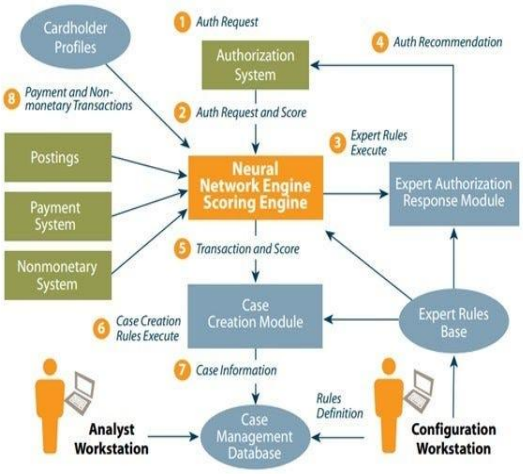


Fig. 2. Workflow of Click Fraud Detection System The workflow begins with data collection, where user click activity is captured and stored in the database. The collected data undergoes preprocessing to remove inconsistencies such as missing values and duplicates.

Next, feature extraction is performed to derive meaningful behavioral attributes. These features are fed into the trained MLP model, which generates a prediction score for each click. Based on this score, the system classifies the click as either fraudulent or legitimate.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435 (Online) Volume 15, Issue 05, May 2026)

The results are stored and visualized through the web interface. Additionally, a feedback mechanism allows continuous learning and model retraining, ensuring adaptability to evolving fraud patterns.



VI. Software Requirements

A. Programming Language

Python (Version 3.7 or higher)

B. Libraries

- NumPy
- Pandas
- Scikit-learn
- TensorFlow/Keras
- Matplotlib
- Seaborn

C. Database

MySQL for structured data storage

D. Web Framework

Flask for web deployment and visualization

E. System Specifications

- OS: Windows/Linux/Mac OS
- RAM: Minimum 4 GB (8 GB recommended)
- Storage: Minimum 10 GB
- Processor: Intel i3 or higher

VII. Methodology

The methodology involves several stages, starting with data collection from simulated or real-world datasets. The collected data includes features such as IP address, timestamp, device information, and user session behavior.

Data preprocessing is performed to clean and normalize the dataset, followed by feature engineering to extract meaningful patterns. The dataset is then split into training and testing sets to evaluate model performance.

The MLP model is trained using supervised learning techniques, where labeled data is used to learn patterns associated with fraudulent and legitimate clicks. The model is evaluated using performance metrics such as accuracy, precision, recall, and F1-score.

Finally, the trained model is deployed for real-time fraud detection, with continuous monitoring and retraining to adapt to new fraud patterns.

VIII. Implemented

The system is implemented using Python (version 3.7 or higher), which provides extensive support for data analysis and machine learning. Libraries such as NumPy and Pandas are used for data processing, while Scikit-learn is used for implementing machine learning algorithms. TensorFlow/Keras is used for building the MLP model.

MySQL is used as the backend database for storing clickstream data, ensuring efficient data management. Visualization tools such as Matplotlib and Seaborn are used to analyze data patterns and model performance. Flask is used to develop a web-based interface for real-time interaction with the system.

The implementation is designed to be scalable and efficient, making it suitable for deployment in real-world environments.

IX. Results And Discussion

The performance of the proposed system is evaluated using standard metrics such as accuracy, precision, recall, and F1-score. The results indicate that the system achieves high accuracy in detecting fraudulent clicks while maintaining low false positive rates.

The MLP model demonstrates strong capability in capturing complex behavioral patterns and identifying subtle anomalies. The system effectively detects bot-generated clicks and abnormal user activities.

However, certain limitations are observed, including the need for large datasets for training and increased computational requirements for deep learning models.

Despite these challenges, the system provides a reliable and scalable solution for click fraud detection.

X. Conclusion

This paper presents an AI-based click fraud detection system that utilizes deep learning techniques to identify fraudulent activities in online advertising. The proposed system achieves high accuracy and adaptability, making it suitable for modern digital environments.

By integrating Python-based tools, MySQL database, and Flask framework, the system ensures scalability and real-time performance. The proposed approach significantly reduces financial losses caused by click fraud and enhances the reliability of digital advertising platforms.

XI. Future Work

Future enhancements may include the integration of real-time data streaming technologies such as Apache Kafka and Spark. Advanced deep learning models, including LSTM and reinforcement learning, can be explored to further improve detection accuracy.

Additionally, deploying the system on cloud platforms can enhance scalability and accessibility. Unsupervised learning techniques can also be incorporated to detect unknown fraud patterns and improve system robustness.

REFERENCES

[1] Angelin Rosy M, Anusha M, "Visual Cryptography and OTP Against Phishing Threats," International Journal of Research and Analytical Reviews, vol. 12, no. 2, pp. 11–16, May 2025, E-ISSN: 2348-1269, P-ISSN: 2349-5138.

[2] S. S. Ravi, V. Ravi, and K. R. Rao, "Detection of click fraud in online advertising using machine learning techniques," Journal of Information Security and Applications, vol. 47, pp. 1–13, 2019, doi: 10.1016/j.jisa.2019.102364.

[3] N. Metwally, D. Agrawal, and A. El Abbadi, "Duplicate detection in click streams," Proceedings of the 14th International World Wide Web Conference (WWW), 2005, pp. 12–21, doi: 10.1145/1060745.1060748.

[4] C. C. Aggarwal, "Outlier analysis," Springer, 2017, doi: 10.1007/978-3-319-47578-3.

[5] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), 2016, pp. 785–794, doi: 10.1145/2939672.2939785.

[6] L. Breiman, "Random forests," Machine Learning, vol. 45, no. 1, pp. 5–32, 2001, doi: 10.1023/A:1010933404324.

[7] P. Lewis, E. Perez, A. Piktus, et al., "Retrieval-augmented generation for knowledge-intensive NLP tasks," Advances in Neural Information Processing Systems (NeurIPS), vol. 33, pp. 9459–9474, 2020, doi: 10.48550/arXiv.2005.11401.

[8] S. Gupta, R. Ranjan, and R. Singh, "A comprehensive survey of retrieval-augmented generation (RAG): Evolution, current landscape and future directions," arXiv preprint, arXiv:2410.12837, 2024, doi: 10.48550/arXiv.2410.12837.

[9] A. Oentaryo, E.-P. Lim, D. Lo, K.-L. Chua, and M. Finegold, "Predicting response in mobile advertising with hierarchical importance-aware factorization machine," Proceedings of the 7th ACM International Conference on Web Search and Data Mining (WSDM), 2014, doi: 10.1145/2556195.2556254.

[10] A. Casella and W. Wang, "Performant LLM agentic framework for conversational AI," arXiv preprint, arXiv:2503.06410, 2025, doi: 10.48550/arXiv.2503.06410.