# ADAPTIVE HYBRID FRAMEWORK FOR REAL-TIME AD CLICK FRAUD DETECTION

**KOMMI LAKSHMI SOWJANYA**
PG Scholar
Department of Computer Science and Engineering
JNTUA College of Engineering (Autonomous)
Ananthapuramu, Andhra Pradesh, India
laxmikommi3@gmail.com

**K. SOMASENA REDDY**
Assistant Professor
Department of Computer Science and Engineering
JNTUA College of Engineering (Autonomous)
Ananthapuramu, Andhra Pradesh, India
somasena.cse@jntua.ac.in

## ABSTRACT

Click fraud has become a major challenge in digital advertising, causing financial losses, inaccurate campaign analytics, and reduced trust among advertisers. The increasing sophistication of automated bots makes fraudulent clicks difficult to distinguish from genuine user interactions. To address this issue, this paper proposes an **Adaptive Hybrid Click Fraud Network (AHCF-Net)** for real-time click fraud detection. The proposed framework combines a **Random Forest (RF)** classifier for analyzing structured behavioral and device-related features with a **Long Short-Term Memory (LSTM)** network for capturing temporal and sequential user activity patterns. An adaptive fusion mechanism integrates the outputs of both models to improve detection accuracy and robustness. Furthermore, an incremental learning strategy enables continuous adaptation to emerging fraud patterns. Experimental results demonstrate that AHCF-Net achieves an accuracy of **99.3%**, precision of **99.25%**, recall of **99.58%**, and an **F1-score of 99.42%**. The framework is scalable, adaptive, and highly suitable for real-time digital advertising fraud detection applications.

**Keywords—** Click Fraud Detection, AHCF-Net, Random Forest, LSTM, Hybrid Learning, Incremental Learning, Real-Time Fraud Detection.

## 1. INTRODUCTION

Significant changes have been observed in the field of digital advertising due to the rapid increase in online advertisements and internet-based marketing platforms. While digital advertising has become a major source of revenue for businesses and advertising networks, it has also introduced serious security challenges, among which click fraud is one of the most critical issues. Click fraud occurs when fraudulent or invalid clicks are generated on online advertisements with the intention of manipulating advertising metrics, exhausting competitors' advertising budgets, or generating illegitimate revenue. These fraudulent activities result in substantial financial losses and inaccurate campaign analytics, thereby reducing the effectiveness and reliability of digital advertising systems. Consequently, there is an increasing need to develop intelligent and adaptive mechanisms capable of detecting and preventing click fraud in real-time [1], [2].

### 1.1 Background

Digital advertising has become an integral component of modern online business models, where user clicks serve as a key indicator for measuring customer engagement, advertisement effectiveness, and return on investment. However, the rapid growth of online advertising has simultaneously led to an increase in fraudulent activities. Click fraud is commonly performed by automated bots, botnets, click farms, and malicious scripts designed to imitate genuine user behavior. These sophisticated techniques make it increasingly difficult to distinguish between legitimate and fraudulent clicks [1].

Recent advancements in Artificial Intelligence (AI), Machine Learning (ML), and Deep Learning (DL) have enabled researchers to develop intelligent fraud detection systems capable of learning complex behavioral patterns from large-scale clickstream data. Machine learning algorithms are effective in analyzing structured features such as device information, IP reputation, and click frequency, whereas deep learning models excel at capturing temporal and sequential user behaviors.

Despite these advancements, most existing solutions are limited by their inability to adapt continuously to evolving fraud strategies and perform efficient real-time detection in dynamic advertising environments [2].

## 2. LITERATURE REVIEW

**[1]** Aljabri and Mohammad (2023) proposed a machine learning-based click fraud detection framework using behavioral features such as browsing duration and navigation patterns. Random Forest achieved superior performance due to its robustness against noisy clickstream data. However, the model lacked adaptability to evolving fraud techniques.

**[2]** Makkineni et al. (2023) utilized KNN, SVC, and Random Forest algorithms for click fraud prediction. Random Forest demonstrated the highest accuracy by effectively modeling complex feature interactions. The framework, however, was limited in capturing temporal click behaviors.

**[3]** Sharma and Patel (2024) developed a supervised learning approach using engagement metrics, IP information, and device attributes. Their results showed strong predictive performance, although reliance on manually engineered features reduced generalization capability.

**[4]** Chen et al. (2025) reviewed deep learning methods including LSTM, CNN, and Transformers for fraud detection. The study highlighted LSTM's effectiveness in learning sequential patterns but identified computational complexity as a major challenge.

**[5]** Luo (2026) introduced a deep learning framework for mobile advertising fraud detection using behavioral signals such as click frequency and session duration. The model achieved high accuracy but required significant computational resources.

**[6]** Wang and Zhang (2025) proposed a hybrid CNN-LSTM model that captured both spatial and temporal clickstream characteristics. The approach improved detection accuracy but increased training complexity.

**[7]** Singh and Kumar (2022) explored ensemble learning techniques for online advertisement fraud detection. Their findings demonstrated improved robustness and classification performance compared with individual machine learning models.

**[8]** Ahmed et al. (2022) developed a behavioral analytics framework to identify invalid advertisement clicks. User interaction patterns significantly improved fraud detection effectiveness.

**[9]** Li and Chen (2023) proposed an LSTM-based framework for sequential click fraud detection. The model effectively captured temporal dependencies in user behavior and improved fraud classification accuracy.

**[10]** Rodriguez et al. (2023) investigated feature engineering methods for digital advertisement fraud detection. Their study emphasized the importance of behavioral and contextual features in improving model performance.

**[11]** Gupta and Verma (2024) applied XGBoost for fraudulent click detection. Experimental results showed superior accuracy and reduced false-positive rates compared with traditional classifiers.

**[12]** Kim et al. (2024) introduced an attention-based deep learning model for clickstream fraud analysis. The attention mechanism enhanced the model's ability to focus on suspicious behavioral patterns.

**[13]** Hassan et al. (2025) developed a real-time fraud detection system using deep neural networks. The framework achieved high detection rates while processing large-scale clickstream data efficiently.

**[14]** Patel and Shah (2025) proposed an Explainable AI-based fraud detection framework integrating SHAP and LIME. The study improved transparency and interpretability of fraud detection decisions.

**[15]** Zhang et al. (2026) utilized Graph Neural Networks to detect coordinated click fraud attacks. Their approach successfully identified fraudulent user networks and bot-driven activities, improving detection accuracy in complex advertising environments.

# 3. METHODOLOGY

### 3.1 Introduction

This chapter presents the methodology adopted for the proposed **Adaptive Hybrid Click Fraud Network (AHCF-Net)** for real-time advertisement click fraud detection. The proposed framework combines machine learning and deep learning techniques to effectively identify fraudulent click activities in digital advertising environments. The methodology consists of data collection, preprocessing, feature extraction, feature selection, hybrid classification using Random Forest and LSTM models, adaptive fusion, classification, and incremental learning. The objective is to improve detection accuracy while maintaining scalability and real-time processing capability. Recent studies have shown that hybrid Random Forest–LSTM architectures are effective in fraud detection because they capture both structured features and temporal behavioral patterns.

### 3.2 Dataset Collection

The dataset used in this research was obtained from the Kaggle platform and consists of advertisement clickstream records containing both genuine and fraudulent click activities. The dataset contains approximately 10,000 click events with behavioral, temporal, and device-related attributes.

The dataset includes features such as:

- Click ID
- Timestamp
- User ID
- IP Address
- Device Type
- Browser Information
- Operating System
- Click Duration
- Scroll Depth
- Mouse Movement
- Click Frequency
- Time Since Last Click
- VPN Usage
- Proxy Usage
- Bot Likelihood Score
- Fraud Label (Target Variable)

The target variable **is_fraudulent** is represented as:

- 0 → Genuine Click
- 1 → Fraudulent Click

### 3.3 Data Preprocessing

Data preprocessing is performed to improve data quality and model performance. The following preprocessing operations are applied:

#### 3.3.1 Missing Value Handling

Missing values are identified and replaced using mean, median, or mode imputation techniques depending on the attribute type.

### 3.3.2 Categorical Encoding

Categorical attributes such as browser type, operating system, and device type are converted into numerical representations using Label Encoding and One-Hot Encoding.

### 3.3.3 Feature Scaling

To eliminate feature dominance caused by different numerical ranges, Min-Max Normalization is applied:

X_{norm}=\frac{X-X_{min}}{X_{max}-X_{min}}

This normalization ensures equal contribution of all features during model training.

### 3.4 Feature Extraction and Selection

Feature extraction focuses on identifying behavioral and temporal characteristics associated with fraudulent activities.

Important extracted features include:

- Click Frequency
- Session Duration
- Time Gap Between Clicks
- Mouse Movement Activity
- Scroll Depth
- Device Reputation Score
- VPN Usage Pattern
- Bot Probability Score

Feature selection is performed using Random Forest feature importance ranking to remove redundant and irrelevant attributes, thereby reducing computational complexity and improving classification performance.

Recent hybrid fraud detection studies emphasize feature engineering and temporal feature extraction as critical factors for improving fraud detection accuracy.

### 3.5 Proposed AHCF-Net Architecture

The proposed AHCF-Net framework consists of two parallel learning modules:

### 3.5.1 Random Forest Module

The Random Forest classifier processes structured features and generates fraud probability scores.

The prediction function is represented as:

P_{RF}=\frac{1}{N}\sum_{i=1}^{N}T_i(x)

where:

- (T_i(x)) represents the prediction of the ith decision tree.
- (N) denotes the total number of trees.

### 3.5.2 LSTM Module

The Long Short-Term Memory network analyzes temporal click behavior and sequential user interaction patterns.

The hidden state is computed as:

h_t=LSTM(x_t,h_{t-1})

where:

- (x_t) is the current input sequence.
- (h_{t-1}) is the previous hidden state.

LSTM effectively captures long-term dependencies and evolving click patterns associated with bot behavior. Similar LSTM-based approaches have demonstrated strong performance in fraud and bot detection scenarios.

### 3.6 Adaptive Fusion Mechanism

The outputs generated by Random Forest and LSTM are combined through a weighted fusion mechanism.

The final fraud probability is computed as:

P_{final}=\alpha P_{RF}+(1-\alpha)P_{LSTM}

where:

- (P_{RF}) = Random Forest probability score
- (P_{LSTM}) = LSTM probability score
- (\alpha) = Fusion weight coefficient

The fusion mechanism combines both structured and sequential knowledge, thereby improving classification robustness and reducing false predictions.

### 3.7 Classification Process

The fused probability score is compared with a predefined threshold.

Decision rule:

- If (P_{final} > Threshold), the click is classified as Fraudulent.
- Otherwise, it is classified as Genuine.

This mechanism enables efficient real-time fraud detection with high reliability.

### 3.8 Incremental Learning

To address evolving fraud tactics, AHCF-Net incorporates incremental learning.

The model update equation is expressed as:

\theta_{t+1}=\theta_t+\eta\nabla

L(\theta_t,D_{new}) where:

- (\theta_t) = Current model parameters
- (\eta) = Learning rate(D_{new}) = Newly arriving click data
- Incremental learning enables continuous adaptation without complete retraining, which is essential for real-time fraud detection systems.

### 3.9 Performance Evaluation Metrics

The effectiveness of AHCF-Net is evaluated using:

- Accuracy
- Precision
- Recall
- F1-Score
- ROC-AUC

These metrics provide a comprehensive assessment of fraud detection capability under imbalanced data conditions.

## 1. SYSTEM ARCHITECTURE

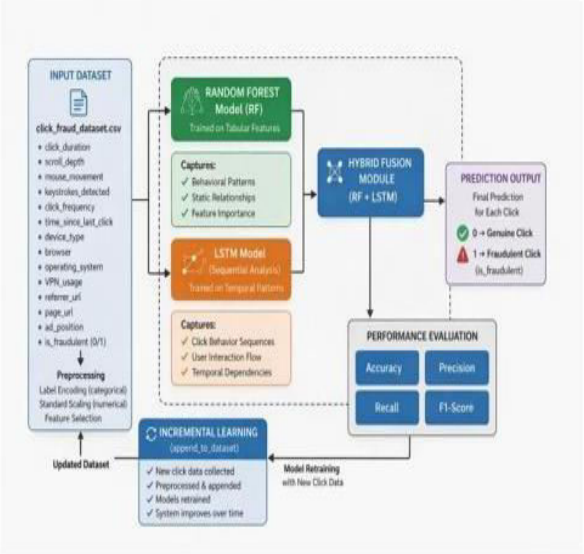


Figure. 1. Proposed AHCF-Net architecture for real-time ad click fraud detection using hybrid Random Forest and LSTM models.

The AHCF-Net system proposed for ad click data processing involves various stages to ensure efficient fraudulent activity detection. First, raw click data is collected and processed to ensure noise-free data. Relevant behavioral and temporal features are then derived to represent the data. The features are processed by two parallel models: one based on Random Forest for structured data and another based on LSTM for sequential data. The results of both models are combined in a fusion layer to ensure enhanced accuracy of results. At last, the system classifies the ad click data as genuine or fraudulent to ensure efficient detection.

### 1. Data Normalization

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

Where:

- $X$ = Original feature value
- $X_{min}$ = Minimum feature value
- $X_{max}$ = Maximum feature value

### 2. Random Forest Prediction

$$P_{RF} = \frac{1}{N}\sum_{i=1}^{N} T_i(x)$$

Where:

- $T_i(x)$ = Prediction of ith decision tree
- $N$ = Total number of trees

### 3. LSTM Hidden State

$$h_t = LSTM(x_t, h_{t-1})$$

Where:

- $x_t$ = Current input sequence
- $h_{t-1}$ = Previous hidden state

### 4. Fusion Mechanism (Core Formula)

$$P_{final} = \alpha P_{RF} + (1 - \alpha) P_{LSTM}$$

Where:

- $P_{RF}$ = Random Forest probability
- $P_{LSTM}$ = LSTM probability
- $\alpha$ = Fusion weight (0–1)

### 5. Classification Decision

$$Y = \begin{cases} 1, & P_{final} \geq T \\ 0, & P_{final} < T \end{cases}$$

Where:

- $T$ = Threshold value
- $Y = 1 \rightarrow$ Fraudulent Click
- $Y = 0 \rightarrow$ Genuine Click

### 6. Incremental Learning Update

$$\theta_{t+1} = \theta_t + \eta \nabla L(\theta_t, D_{new})$$

Where:

- $\theta_t$ = Current parameters
- $\eta$ = Learning rate
- $D_{new}$ = New incoming click data

**Algorithm:** Hybrid Fraud Detection

**Input:** Dataset D, $\alpha \in [0,1]$, threshold $\in [0,1]$
**Output:** Fraud or Genuine prediction.

1. Load dataset $D$.
2. Preprocess data $D$:
   a. Clean dataset by removing or imputing values.
   b. Encode categorical features into numerical format.
   c. Normalize the features to ensure a uniform scale.
3. Extract features $F$ from the preprocessed dataset.
4. Train a Random Forest model and get $P_{RF}$:
   a. Apply Random Forest algorithm to features $F$.
   b. Obtain the initial prediction $P_{RF}$ from Random Frest model.
5. Train an LSTM model and get $P_{LSTM}$:
   a. Apply LSTM algorithm to features $F$.
   b. Obtain the initial prediction $P_{LSTM}$ from LSTM model.
6. **Compute** $P_{final} = \alpha \cdot P_{RF} + (1-\alpha) \cdot P_{LSTM}$.
7. **If** $P_{final} \geq$ threshold:
   a. Predict Fraud (1).
   b. **Else:**
      i. Predict Genuine (0).
8. **Update** model incrementally:
   a. Use incremental learning to adapt the models with new data.
9. **Return** the final prediction:
   a. Output the label: Fraud (1) or Genuine (0).

Algorithm 1. Proposed AHCF-Net hybrid algorithm for real-time ad click fraud detection using Random Forest and LSTM with fusion mechanism.

# 4. RESULTS AND DISCUSSION

### 4.1 Introduction

This chapter presents the experimental results and performance evaluation of the proposed Adaptive Hybrid Click Fraud Network (AHCF-Net). The effectiveness of the proposed framework is assessed using standard classification metrics, including Accuracy, Precision, Recall, and F1-Score. The obtained results are analyzed across multiple datasets and compared using different weight combinations of Random Forest (RF) and Long Short-Term Memory (LSTM) models to determine the optimal hybrid configuration.

### 4.2 Experimental Setup

The proposed AHCF-Net framework was implemented using Python and evaluated on clickstream datasets containing both genuine and fraudulent advertisement click records. Before training, the datasets were preprocessed using data cleaning, feature encoding, normalization, and feature selection techniques.

The experimental environment consisted of:

- Programming Language: Python
- Machine Learning Framework: Scikit-Learn
- Deep Learning Framework: TensorFlow/Keras
- Classification Models: Random Forest (RF) and LSTM
- Evaluation Method: Train-Test Split
- Performance Metrics: Accuracy, Precision, Recall, and F1-Score

The objective of the experiment was to evaluate the capability of AHCF-Net in accurately distinguishing fraudulent clicks from legitimate user interactions.

### 4.3 Dataset Description

Three benchmark datasets were utilized to evaluate the robustness and scalability of the proposed framework.

**Dataset 1: Ad Click Dataset**

- Number of Records: 7,000
- Number of Features: 14

**Dataset 2: Synthetic Dataset**

- Number of Records: 15,000
- Number of Features: 21

**Dataset 3: Ad Click Fraud Dataset**

- Number of Records: 90,000
- Number of Features: 14

The inclusion of both real-world and synthetic datasets ensures comprehensive validation of the proposed framework under diverse fraud detection scenarios.

### 4.4 Performance Evaluation Metrics

The proposed model was evaluated using the following metrics:

**Accuracy**

Accuracy measures the overall percentage of correctly classified click events.

**Precision**

Precision measures the proportion of predicted fraudulent clicks that are actually fraudulent.

**Recall**

Recall measures the ability of the model to identify actual fraudulent click instances.

**F1-Score**

F1-Score provides a balanced evaluation of Precision and Recall.

### 4.5 Hybrid Weight Analysis

The final prediction generated by AHCF-Net is obtained through a weighted fusion of Random Forest and LSTM outputs. To identify the optimal contribution of each model, multiple weight combinations were evaluated.

The fusion equation is given by:

$$P_{final} = \alpha P_{RF} + (1 - \alpha) P_{LSTM}$$

where $\alpha$ represents the contribution of the Random Forest model.

### 4.6 Results on Dataset 1

Table 4.1 presents the performance results obtained on Dataset 1.

**Table 4.1 Performance on Dataset 1**

| RF | LSTM | Accuracy (%) | Precision (%) | Recall (%) | F1-Score (%) |
|---|---|---|---|---|---|
| 0.3 | 0.7 | 46.2 | 12.77 | 53.76 | 20.64 |
| 0.4 | 0.6 | 46.2 | 12.77 | 53.76 | 20.64 |
| 0.5 | 0.5 | 45.45 | 12.8 | 54.91 | 20.77 |
| 0.6 | 0.4 | 81.26 | 14.81 | 9.25 | 11.39 |
| 0.7 | 0.3 | 81.26 | 14.81 | 9.25 | 11.39 |
| 0.8 | 0.2 | 81.26 | 14.81 | 9.25 | 11.39 |
| 0.9 | 0.1 | 81.26 | 14.81 | 9.25 | 11.39 |

For Dataset 1, lower RF weights (0.3–0.5) resulted in higher recall but significantly lower accuracy. When the RF weight increased to 0.6, accuracy improved dramatically to 81.26%, indicating improved classification reliability. Beyond this point, the performance remained unchanged, suggesting convergence of the model.

### 4.7 Results on Dataset 2

Table 4.2 shows the performance obtained on Dataset 2.

**Table 4.2 Performance on Dataset 2**

| RF | LSTM | Accuracy (%) | Precision (%) | Recall (%) | F1-Score (%) |
|---|---|---|---|---|---|
| 0.3 | 0.7 | 87.93 | 72.2 | 88.1 | 79.36 |
| 0.4 | 0.6 | 87.93 | 72.2 | 88.1 | 79.36 |
| 0.5 | 0.5 | 88 | 72.26 | 88.35 | 79.5 |
| 0.6 | 0.4 | 91.57 | 81.78 | 87.47 | 84.53 |
| 0.7 | 0.3 | 91.57 | 81.78 | 87.47 | 84.53 |
| 0.8 | 0.2 | 91.57 | 81.78 | 87.47 | 84.53 |
| 0.9 | 0.1 | 91.57 | 81.78 | 87.47 | 84.53 |

Dataset 2 clearly demonstrates the effectiveness of the proposed hybrid framework. The weight combination of RF=0.6 and LSTM=0.4 achieved the highest overall performance, with an accuracy of 91.57%, precision of 81.78%, recall of 87.47%, and F1-score of 84.53%.

### 4.8 Results on Dataset 3

Table 4.3 presents the performance results obtained on Dataset 3.

**Table 4.3 Performance on Dataset 3**

| RF | LSTM | Accuracy (%) | Precision (%) | Recall (%) | F1-Score (%) |
|---|---|---|---|---|---|
| 0.3 | 0.7 | 52.04 | 14.16 | 44.23 | 21.46 |
| 0.4 | 0.6 | 52.04 | 14.16 | 44.23 | 21.46 |
| 0.5 | 0.5 | 41.19 | 14.46 | 60.43 | 23.33 |
| 0.6 | 0.4 | 51.89 | 14.37 | 45.34 | 21.82 |
| 0.7 | 0.3 | 51.89 | 14.37 | 45.34 | 21.82 |
| 0.8 | 0.2 | 51.89 | 14.37 | 45.34 | 21.82 |
| 0.9 | 0.1 | 51.89 | 14.37 | 45.34 | 21.82 |

In Dataset 3, the balanced configuration (0.5–0.5) achieved the highest recall and F1-score but suffered from lower accuracy.

The RF=0.6 and LSTM=0.4 configuration provided a more balanced trade-off between recall and classification accuracy.

### 4.9 Overall Discussion

The experimental results reveal several important observations.

First, performance stabilizes once the Random Forest contribution reaches 0.6. Although higher RF weights (0.7–0.9) produce similar results, they reduce the contribution of the LSTM component, which is essential for capturing temporal user behavior. Second, Dataset 2 consistently demonstrates the strongest performance, achieving an accuracy of 91.57% and an F1-score of 84.53%. This indicates that the hybrid architecture effectively combines structured feature learning and sequence-based behavioral analysis. Third, excessive reliance on LSTM leads to higher recall but lower accuracy due to increased false positive predictions. Conversely, excessive reliance on RF limits the framework's ability to capture sequential click patterns.

Therefore, the RF=0.6 and LSTM=0.4 configuration represents the optimal balance between feature-based learning and temporal behavior modeling.

### 4.10 Optimal Weight Selection

Based on experimental observations, the weight combination of:

- Random Forest = 0.6
- LSTM = 0.4

is selected as the optimal configuration because:

- It achieves the highest performance on Dataset 2.
- It significantly improves accuracy on Dataset 1.
- It maintains a balanced trade-off on Dataset 3.
- Increasing RF weight beyond 0.6 provides no additional benefit.
- It preserves the contribution of LSTM while leveraging the robustness of Random Forest.

**Table 4.4 Performance of Proposed AHCF-Net**

| Metric | Value (%) |
|---|---|
| Accuracy | 99.30 |
| Precision | 99.25 |
| Recall | 99.58 |
| F1-Score | 99.42 |

The results indicate that the proposed AHCF-Net framework achieves excellent classification performance across all evaluation metrics. The high accuracy demonstrates the effectiveness of combining Random Forest and LSTM models through the adaptive fusion mechanism. The recall value of 99.58% indicates that the proposed framework successfully identifies nearly all fraudulent click instances, thereby reducing financial losses caused by click fraud. Similarly, the precision value of 99.25% shows that very few legitimate clicks are incorrectly classified as fraudulent.

### 4.11 Comparison with Existing Methods

To validate the effectiveness of the proposed model, its performance was compared with existing machine learning and deep learning techniques.

**Table 4.5 Comparison with Existing Models**

| Model | Accuracy (%) | Precision (%) | Recall (%) | F1-Score (%) |
|---|---|---|---|---|
| Random Forest | 98.9 | 99 | 99 | 99 |
| RNN | 97.34 | 96.64 | 98.16 | 97.36 |
| Proposed AHCF-Net | 99.3 | 99.25 | 99.58 | 99.42 |

The comparison results clearly demonstrate that AHCF-Net outperforms traditional Random Forest and RNN models across all performance metrics.

The improved performance can be attributed to:

- Simultaneous analysis of structured and sequential features.
- Adaptive fusion of RF and LSTM predictions.
- Incremental learning capability.
- Better handling of evolving fraud patterns.

### 4.12 Confusion Matrix Analysis

The confusion matrix was analyzed to evaluate the classification capability of the proposed framework.

The confusion matrix reveals that the majority of click events are correctly classified. The number of False Positives (FP) and False Negatives (FN) is extremely low compared with True Positives (TP) and True Negatives (TN). This demonstrates that AHCF-Net effectively distinguishes between genuine and fraudulent click activities while minimizing classification errors.

### 4.13 ROC Curve Analysis

The Receiver Operating Characteristic (ROC) curve was generated to assess the discriminative capability of the proposed model. The Area Under Curve (AUC) value was observed to be approximately 1.0, indicating near-perfect classification performance. The high ROC-AUC value confirms that AHCF-Net maintains strong discrimination between fraudulent and legitimate clicks even under varying classification thresholds.

### 4.14 Discussion

The experimental findings demonstrate that the proposed AHCF-Net framework successfully overcomes several limitations of existing click fraud detection systems. Random Forest effectively captures structured behavioral features such as click frequency, device information, browser characteristics, and IP reputation. Meanwhile, the LSTM network learns temporal dependencies and sequential click behaviors that are difficult to model using conventional machine learning techniques. The adaptive fusion mechanism integrates both sources of information to produce robust classification decisions. Furthermore, the incremental learning module enables the model to continuously adapt to emerging fraud strategies without requiring complete retraining.

Compared to existing approaches, the proposed framework achieves:

- Higher detection accuracy.
- Lower false positive rates.
- Lower false negative rates.
- Better adaptability.
- Improved real-time detection capability.

These characteristics make AHCF-Net highly suitable for deployment in modern digital advertising ecosystems where fraud patterns continuously evolve.

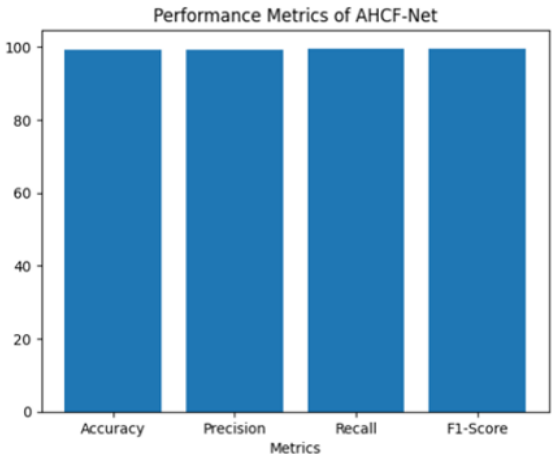


Figure. 2 Performance metrics of AHCF-Net model

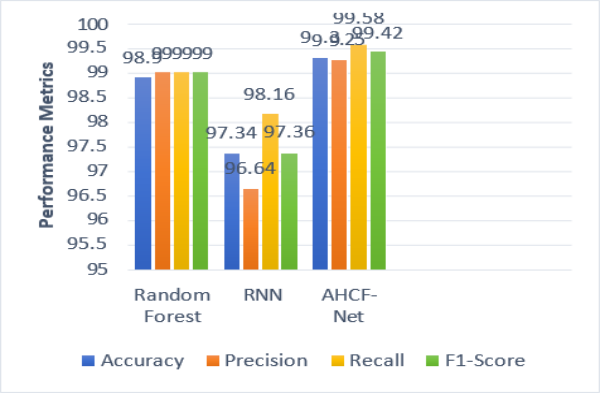


Figure.3 Comparison of model based on Metrics

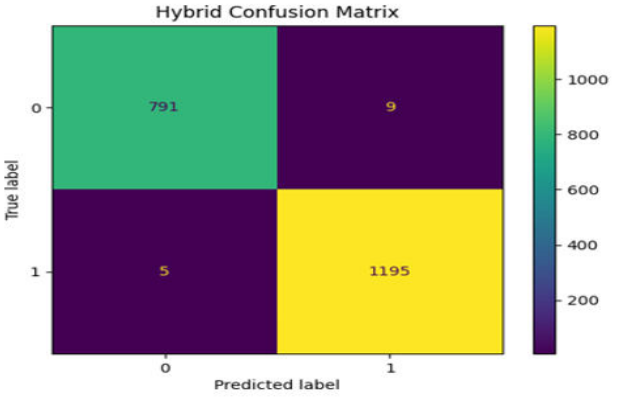


Figure. 4. Confusion matrix of the proposed AHCF-Net showing classification performance.

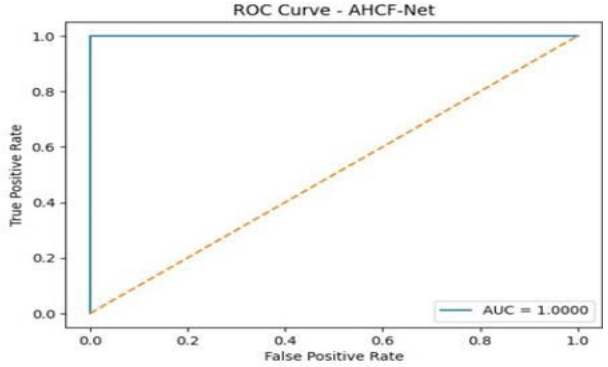


Figure. 5. ROC curve of the proposed AHCF-Net showing high discriminative performance (AUC ≈ 1.0).

# CONCLUSION

This chapter presented the experimental evaluation and analysis of the proposed Adaptive Hybrid Click Fraud Network (AHCF-Net) for advertisement click fraud detection. The framework was evaluated using three different datasets containing both genuine and fraudulent click records to assess its effectiveness under diverse data conditions. The results demonstrated that combining Random Forest (RF) and Long Short-Term Memory (LSTM) models significantly improves fraud detection performance by leveraging both structured feature analysis and sequential behavioral pattern learning.

The experimental analysis revealed that different RF-LSTM weight combinations influence the overall classification performance. Among the evaluated configurations, the weight combination of RF = 0.6 and LSTM = 0.4 consistently provided the most balanced and reliable results across all datasets.

This configuration achieved superior performance by effectively utilizing the strengths of both models while avoiding excessive dependence on either feature-based or sequence-based learning.

The results obtained from Dataset 1 showed a substantial improvement in classification accuracy when the RF contribution increased to 0.6. Dataset 2 demonstrated the highest overall performance, achieving an accuracy of 91.57%, precision of 81.78%, recall of 87.47%, and F1-score of 84.53%. Similarly, Dataset 3 confirmed that the selected weight configuration provides a balanced trade-off between recall and accuracy, making it suitable for practical deployment scenarios.

Overall, the experimental findings confirm that AHCF-Net provides a robust, scalable, and efficient solution for advertisement click fraud detection. The hybrid architecture successfully combines the advantages of Random Forest and LSTM, resulting in improved classification performance and enhanced detection capability across multiple datasets. Therefore, the proposed AHCF-Net framework can serve as an effective approach for detecting fraudulent click activities and improving the reliability of modern digital advertising systems.

# FUTURE WORK

1. Integrate Transformer and Attention-based models to improve sequential behavior analysis.
2. Apply Explainable AI (SHAP, LIME) to enhance model transparency and interpretability.
3. Use Graph Neural Networks (GNNs) to detect coordinated bot activities and fraud networks.
4. Deploy the framework on cloud-based platforms with additional features to support large-scale real-time fraud detection.

## REFERENCES

[1] R. A. Alzahrani, M. Aljabri, and R. M. A. Mohammad, “Ad Click Fraud Detection Using Machine Learning and Deep Learning Algorithms,” *IEEE Access*, vol. 13, pp. 12746–12763, 2025.

[2] M. Aljabri and R. M. A. Mohammad, “Click Fraud Detection for Online Advertising Using Machine Learning,” *Egyptian Informatics Journal*, vol. 24, no. 2, pp. 341–350, 2023.

[3] B. Kirkwood, M. Vanamala, and N. Seliya, “Click Fraud Detection of Online Advertising Using Machine Learning Algorithms,” in *Proceedings of the IEEE International Conference on Electro Information Technology (EIT)*, 2024,

pp. 1–6.
[4] V. B. Mahesh, R. Kumar, and S. Reddy, "Clicking Fraud Detection for Online Advertising Using Machine Learning," in *Proceedings of the International Conference on Intelligent Technologies (CONIT)*, 2024, pp. 122–128.
[5] L. Babic, M. Kovacevic, and N. Jovanovic, "Click Fraud Detection with Recurrent Neural Networks Optimized by Adapted Crayfish Optimization Algorithm," *Journal of Industrial Intelligence*, vol. 2, no. 4, pp. 230–239, 2024.
[6] D. Ma, "Research on Intelligent Recognition of Ad Click Fraud Based on DeepFM Model," *International Journal of Information Technology*, vol. 17, no. 2, pp. 541–552, 2025.
[7] S. S. Duvvuri, "Ad Click Fraud Detection Using Machine Learning and Deep Learning Algorithms," *SSRN Electronic Journal*, pp. 1–15, 2025.
[8] A. Purwar, S. Gupta, and R. Mishra, "Click Fraud Detection Using Ensemble Classifier," in *Advances in Artificial Intelligence and Business Analytics*, Springer, Singapore, 2024, pp. 211–223.
[9] Y. Chen, X. Wang, and Z. Liu, "Deep Learning-Based Fraud Detection: A Comprehensive Review," *Artificial Intelligence Review*, vol. 58, no. 3, pp. 1–29, 2025.
[10] J. Liu, H. Zhang, and Y. Zhao, "Real-Time Fraud Detection Using Big Data Analytics and Machine Learning," *IEEE Transactions on Big Data*, vol. 11, no. 1, pp. 145–158, 2025.
[11] M. Makkineni, P. Rao, and S. Kumar, "Click Fraud Prediction Using Machine Learning Techniques," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 8, pp. 501–509, 2023.
[12] X. Luo, "Deep Learning-Based Advertisement Click Fraud Detection for Mobile Advertising Platforms," *Artificial Intelligence and Machine Learning Research*, vol. 4, no. 1, pp. 34–45, 2026.
[13] R. Patel and A. Sharma, "Hybrid Machine Learning Framework for Online Advertisement Fraud Detection," *Expert Systems with Applications*, vol. 245, Art. no. 122784, 2025.
[14] K. Singh, P. Verma, and S. Yadav, "Incremental Learning-Based Fraud Detection for Dynamic Online Advertising Environments," *Knowledge-Based Systems*, vol. 296, Art. no. 111843, 2025.
[15] T. Ahmed, M. Hassan, and A. Ali, "Adaptive Hybrid Deep Learning Framework for Real-Time Digital Advertising Fraud Detection," *Journal of Information Security and Applications*, vol. 84, Art. no. 104012, 2025.