DeepFM-MOTSO: A Deep Factorization Machine Framework Optimized by Multi-Objective Tuna Swarm for Online Advertising Fraud Detection

Dandan Ma[1*], Feng Wan[2]
[1]Department of Literature and Media, Chengdu Jincheng College, Chengdu, Sichuan 610000, China
[2]Chengdu Chiwu Technology Co., Ltd., Sichuan, Chengdu, 610000, China
E-mail: 15390421110@189.cn (CA); 15308181110@189.cn
[*]Corresponding author

Keywords: factorization machine, deep neural networks, advertising networks, recognition model, advertising fraud, click fraud

Received: October 30, 2025

Online advertising fraud is still a significant problem that inflates marketing expenses and distorts campaign performance. In order to improve fraud detection accuracy in the face of extreme class imbalance, this study proposes a hybrid DeepFM–MOTSO framework that combines a Deep Factorization Machine (DeepFM) with a Multi-Objective Tuna Swarm Optimizer (MOTSO). While MOTSO simultaneously optimizes several goals—maximizing F1-score, precision, recall, and AUC, as well as minimizing loss—for balanced classification, the model captures both low-order feature interactions through the FM component and high-order nonlinear representations via deep neural layers. A real-world advertising dataset with roughly 2,043 records that included contextual and behavioral features like session_duration, click_interval, impression_count, and device_type was used to test the method. With an accuracy of 0.952, precision of 0.214, recall of 0.179, F1-score of 0.195, and AUC of 0.864, the experimental results demonstrate that the proposed DeepFM–MOTSO outperformed all comparative baselines, indicating superior capability in identifying minority-class fraudulent instances. The findings confirm that multi-objective optimization effectively improves model convergence, stability, and real-time adaptability for intelligent online advertising fraud detection.

Povzetek: Študija predstavlja izboljšan model za učinkovitejše zaznavanje spletnih oglaševalskih goljufij pri neuravnoteženih podatkih.

1 Introduction

Click fraud in online advertising has existed for several years, and several approaches have been developed to detect it. Initially, efforts focused on identifying anomalous interactions with ads using data tracing and a set of manually created rules. Most such rule-based systems would fail when exposed to more sophisticated fraud mechanisms. Ma and Wan [1] proposed an intelligent recognition model based on DeepFM heterogeneous integration to solve feature interaction and generalization problems. The robust generalization of his model across ad environments greatly enhanced the validity of fraud detection. Alzahrani, Aljabri, and Mohammad [2] examined machine learning and deep learning methods for detecting ad click fraud, building on this progress. They looked at many different models and found that deep learning performed better when the data was more diverse and more abundant.

Later, Batool and Byun [3] constructed an ensemble deep learning model for click fraud detection in pay-per-click ads. Their design integrated several models to capture manifest and latent types of fraud. Going a step further, Sisodia and Sisodia [4] applied transfer learning to recognize the behavioural changes of fraudulent publishers. These models improve detection by learning new patterns rather than relying on the knowledge gained from old data. The work done by Hu et al. [5] presented the notion of using a weighted heterogeneous graph model to depict relationships among mobile advertisers and users, which supported connectivity in networks while detecting possible fraud activities.With the expansion of research, new models began to combine deep learning with hybrid methods. Using artificial intelligence, Teng [6] tracks false traffic in online marketing, and the paper shows how AI could flag suspicious click flows in real time.

Sreekala et al. [7] used deep neural networks to secure online advertising systems against click fraud. Du [8] focused on multi-task fraud detection in finance using reinforcement learning and heterogeneous graph neural

networks, which are also applicable to fraud detection in advertisement systems. Dai et al. [9] propound introducing GemmaWithLoRA, a model that adapts rapidly when there is a change in fraud behaviour. Singh et al. [10] talk about a system that determines the identity of fraudulent publishers using applied intelligence, linking practical detection tools with academic insight. The result of this has been a progressive movement away from the earlier static rule-based systems toward dynamic intelligible recognition models, which learn from behaviour, adjust to new data, and are thus better able to identify deceit within complex online advertising networks.

Contribution of the study

- Novel Integration: Introduces a DeepFM-based heterogeneous integration model optimized with MOTSO for dynamic online ad fraud detection.
- Cross-Domain Adaptability: Validates model performance against baseline architectures drawn from related works, including SVM, GBDT, RNN, and HGNN.
- Data-Driven Insights: Performs comprehensive EDA to identify behavioral correlations between user interactions and fraudulent click patterns.
- Optimization Enhancement: Incorporates bio-inspired multi-objective optimization (MOTSO) to refine network weight convergence and improve recall on minority fraud cases.
- Reproducible Framework: Provides a scalable, modular experimental setup adaptable to other fraud detection domains (finance, e-commerce)

Research objectives

This study has three key objectives:

1. To develop a hybrid DeepFM–MOTSO model for advertising fraud detection.
2. To compare its performance with classical and GNN-based baselines.
3. To evaluate the model on a real-world, highly imbalanced advertising dataset.

2 Related work

Complex systems like the Energy Supply Chain have their challenges in planning and managing them, particularly concerning unknown sources. Therefore, a hybrid simulation optimization framework has been proposed to efficiently quantify the risk in pipelines [11]. Sim-Opt application in agricultural supply chains allows the user to explore scenarios, change the system's parameters, and identify solutions that maximize productivity while minimizing costs and optimizing overall performance. ABM is widely used in the Sim-Opt studies, and in these, agents model dynamic interaction and heterogeneity of behavior of individual entities, which is based on the availability of sufficient and right kind of input data [12].

In real-world systems, Kanei et al. [11] investigated online advertising fraud and how fraud takes place. Their findings indicated that such criminals will change their behavior to avoid detection, rendering static filters ineffective. Real traffic data were gathered, and the way in which fraud spreads through the ad networks was explained, emphasizing the need for a flexible detection system capable of handling real-time data. Sağlam and Kirçova [12] explained the role of AI in fraud detection in programmatic adverting and blockchain-based systems. They detailed how these technologies increase transparency in ad transactions and afford such transactions to be less susceptible to fraudulent manipulation

Ma, Xu, and Zhang [13] developed an intelligent advertising recognition system using recommendation techniques to enhance ad targeting and minimize exposure to fraudulent traffic. Their model differentiates between authentic user engagement and suspicious automated clicks. Vishwakarma and Dhakad [14] presented a comparison of various techniques adopted in different studies regarding online advertisement fraud detection. Comparing such techniques made it possible for them to identify a lacuna: the failure of traditional detection techniques in cases of complex changed behaviors over time.

A systematic review of all proposed methodologies regarding the detection of online ad fraud has been conducted by Baranidharan et al. [15]. They reviewed a number of AI-based models in relation to fraud detection, summing up how the development of deep learning and neural networks has increased the accuracy rate of fraud detection. In fact, Abbas et al. [16] examined various machine learning models for detecting click fraud. They mentioned that the performance of the models relies upon data quality, feature extraction, and training design.

Singh [17] studied the possible role of artificial intelligence and machine learning in preventing advertisement fraud, along with consideration of brand safety. In this context, he linked matters of financial security in advertising with fraud detection. Zhang et al. [18] gave a broader appraisal of machine-learning and deep-learning methods in e-commerce, particularly with an emphasis on the detection of fraudulent activities and comprehension of customer behavior patterns. Zhu et al. [19] reviewed best practices for financial fraud detection in the post-pandemic era. The authors highlighted how AI

techniques currently applied in finance find analogous applications in advertising fraud detection, where common features include imbalanced data and elusive fraudulent patterns.

These works, therefore, expand the paradigm of ad fraud detection to real-world observations, from AI applications to intelligent systems [12-15], and deep learning methods [16-20] (refer Table 1). The trend evolves from model-driven studies toward integrated frameworks for recognizing and responding to fraud.

Table 1: Summary of related work

Reference	Objective	Models	Dataset	Key Findings	Research Gaps
[1].	Use deep RL-based GNN for multi-task fraud detection.	DRL-GNN with VAE.	Financial fraud data.	Improved detection across tasks.	Needs ad fraud validation.
[2].	Detect fraud in online ads using weighted graphs.	Heterogeneous graph embedding.	Mobile ad data.	Better fraud detection accuracy.	Limited scalability tests.
[3].	Design a click-fraud system for ad networks.	Hybrid ML classifiers.	Ad logs.	Detected fraud publishers effectively.	Needs live environment testing.
[4].	Detect ad fraud in real-world traffic.	Rule-based and anomaly detection.	Online ad network logs.	Exposed multiple fraud types.	No DL or temporal modeling.
[5].	Apply AI to ad fraud in blockchain ads.	AI-Blockchain model.	Theoretical.	Suggested transparent transactions.	No empirical validation.
[6].	Apply ML for detecting click fraud.	SVM, DT, RF, ANN.	Clickstream data.	ANN achieved best precision.	No hybrid or TL approach.
[7].	Identify publisher behavior with TL.	Transfer learning with feature extraction.	PPC ad data.	Detected subtle fraud shifts.	Small dataset; needs real-time tests.
[8].	Build intelligent ad recognition system.	Deep recommendation model.	Ad logs.	Improved targeting accuracy.	Not fraud focused.
[9].	Review online ad fraud detection.	ML, DL, Hybrid models.	Literature review.	Summarized fraud detection trends.	No model or dataset testing.
[10].	Survey online ad click fraud.	Rule-based, heuristic.	Literature data.	Classified ad fraud types.	No ML or DL focus.
[11].	Detect ad fraud in live systems.	Behavior-based, rule models.	Real ad traffic.	Static rules failed against adaptive fraud.	Lacked DL techniques.
[12].	Combine AI with blockchain for fraud control.	Conceptual AI-blockchain.	Theoretical.	Promoted transparency and safety.	No experimental setup.
[13].	Enhance ad recognition with AI.	Neural network model.	Simulated ads.	Reduced fake ads and misclassifications.	No fraud evaluation.
[14].	Survey fraud detection in ads.	ML, heuristic models.	Literature.	Highlighted traditional limits.	No DL testing or validation.
[15].	Systematically review AI-based ad fraud detection.	CNN, RNN, LSTM.	Prior research.	DL improved detection rates.	Data diversity missing.
[16].	Compare ML models for fraud detection.	RF, XGBoost, LR, ANN.	Campaign data.	RF and ANN most effective.	Small sample size.

[17].	Study AI's role in brand safety and ad fraud.	Hybrid AI-ML model.	Conceptual cases.	Linked fraud and financial control.	No validation or metrics.
[18].	Review ML/DL methods in fraud detection.	CNN, LSTM, Autoencoder.	Review datasets.	Found DL effective for behavior analysis.	Not ad specific.
[19].	Explore post-pandemic financial fraud detection.	GNN, Ensemble models.	Financial data.	AI handled imbalance well.	Not applied to ad fraud.
[20].	To improve fine-grained detection and subtype classification of DoS and DDoS attacks.	CatBoost (enhanced version)	Real-world network traffic dataset with multiple DoS/DDoS categories	Achieved high accuracy and precision.	Focused primarily on tree-based ensemble learning.

Although the studies listed in Table 1 have provided insightful information about identifying advertising fraud, they still have a lot of limitations. Complex, nonlinear interactions between behavioral, contextual, and temporal variables are not captured by the majority of traditional and hybrid machine-learning models because they rely on linear or low-order feature representations. Despite being more expressive, graph-based and ensemble approaches typically assume static feature relationships and are unable to adjust to quickly changing fraud patterns. Additionally, a number of studies rely largely on manual feature engineering while ignoring the heterogeneity of advertising data, which combines numerical, sequential, and categorical features. In real-time settings, these limitations restrict responsiveness and generalization. To address these issues, a method that combines adaptive optimization and high-order interaction modeling—like the suggested DeepFM-MOTSO framework—is needed.

Research gap

Despite extensive research on ad click fraud detection, notable gaps remain:

- Most models rely on static rule-based filters or shallow ML algorithms that fail under behaviorally adaptive fraud conditions.
- Feature interaction modeling remains limited—linear models cannot capture high-order relationships among categorical and continuous user features.
- Few works integrate optimization techniques such as bio-inspired algorithms to enhance deep model convergence and robustness.
- Prior hybrid models (e.g., ensemble CNN–RNN) emphasize performance but lack interpretability and cross-domain scalability.
- No comprehensive validation exists comparing heterogeneous learning models (like GNNs and DeepFM) using real ad fraud datasets.

This study addresses these limitations through a heterogeneous, optimized DeepFM framework, combining adaptive feature learning, nonlinear interaction modeling, and multi-objective optimization for fraud detection.

3 Materials And method

Dataset

The dataset used in this study was collected from Kaggle (https://www.kaggle.com/datasets/programmer3/fraud-detection-dataset), represents real or simulated advertising clickstream data collected from an online advertising network. It contains both behavioral and contextual information designed to distinguish between legitimate user clicks and fraudulent automated clicks.

Dataset overview

- The dataset consists of 2,043 records and 11 attributes in total.
- It includes a combination of numerical and categorical variables, providing a heterogeneous data structure ideal for hybrid deep learning models.
- The target variable is click_label, a binary indicator where 1 represents fraudulent clicks and 0 represents legitimate clicks.
- The data captures user sessions, ad identifiers, device information, browsing environment, and engagement-related metrics.

- Overall, the dataset is moderately balanced, with approximately 70% legitimate and 30% fraudulent samples.

Feature composition

- User and Ad Identifiers: Each record includes a user_id and ad_id, which uniquely identify the user performing the click and the advertisement clicked. These fields are used to track repetitive or abnormal user–ad interaction patterns.
- Click Timing Information: The attribute click_time stores temporal information. When analyzed, fraudulent clicks often appear at unnatural time intervals or show repetitive bursts during specific hours, indicating bot-generated activity.
- Browser and Device Context: The dataset contains browser_type and device_type columns that describe the user's access platform. Fraudulent clicks tend to originate from emulated or uncommon browsers such as headless WebView clients or automated desktop agents.
- Geographical Indicators: The geo_location field provides regional information. A concentration of clicks from the same region or IP cluster often signals click-farm operations or geographically coordinated fraudulent activity.
- Behavioral Metrics:
    - click_interval denotes the time difference between two consecutive clicks by the same user. Extremely small intervals (for example, under two seconds) are typical indicators of automated clicking.
    - session_duration measures how long a user stays engaged with the ad before performing a click. Legitimate users generally show longer session durations, while fraudulent sessions are notably shorter.
    - num_clicks_in_session represents how many times a user clicks within one active session. A high value suggests automated or scripted clicking behavior.
    - bounce_rate indicates the proportion of sessions where users leave after a single page visit. High bounce rates correlate strongly with fraudulent traffic, as bots rarely explore multiple pages.

The dataset size (2043 samples) is relatively small for fraud detection, which may constrain generalizability. The imbalance and limited fraudulent samples restrict the model's ability to learn rare behavior patterns. This limitation is acknowledged, and future work will involve larger or streaming datasets to improve robustness and external validity.

Train–validation–test protocol and leakage control

To ensure fair evaluation and prevent information leakage, the dataset was partitioned into **training (70%), validation (15%), and test (15%)** subsets. Due to the highly imbalanced nature of the fraud labels, all splits were performed using **Stratified Shuffle Split**, ensuring that each subset preserved the same fraud vs. non-fraud distribution as the full dataset. Since the dataset does not contain user-level or campaign-level identifiers, group-wise splitting was not required; however, special care was taken to guarantee that no identical or near-duplicate click records appeared across different splits.

To account for randomness in sampling, the experiment was repeated **five independent runs**, each initialized with a different random seed: **[42, 56, 77, 91, 123]**. For every model, the reported performance metrics represent the **mean ± standard deviation** across these five runs. This multi-run protocol ensures stable evaluation, reduces sensitivity to random initialization, and prevents models from overfitting to a particular split configuration.

During model development, the validation set was used for hyperparameter selection, threshold examination, and early stopping, while the test set was strictly held out and used only for final evaluation. This split strategy ensures that no test information influences model tuning, eliminating data leakage and guaranteeing a realistic assessment of generalization performance.

4 Proposed model

The proposed Intelligent Recognition Model for Advertising Fraud is based on a DeepFM Heterogeneous Integration framework optimized with the Multi-Objective Tuna Swarm Optimization (MOTSO) algorithm. This model intelligently learns both explicit and implicit feature relationships from heterogeneous advertising data, enabling accurate and adaptive detection of fraudulent click behaviors.

The proposed DeepFM-MOTSO model combines feature interaction learning and adaptive optimization within a unified deep learning structure, designed to overcome the limitations of traditional methods that either fail to capture complex cross-feature relationships or overfit heterogeneous datasets.

Model objectives

The DeepFM-MOTSO model is designed to:

- Learn both low-order and high-order feature interactions from heterogeneous advertising data.
- Accurately recognize fraudulent versus legitimate clicks across dynamic environments.
- Reduce the need for manual feature engineering through end-to-end learning.
- Enhance convergence speed and detection stability via MOTSO-based adaptive optimization.

Optimization and contribution

The MOTSO optimizer adaptively tunes hyperparameters such as learning rate, embedding dimensions, and regularization parameters. Inspired by swarm intelligence, it balances exploration and exploitation to avoid local minima and achieve faster convergence.

Through this integration of feature learning and adaptive optimization, the DeepFM-MOTSO model achieves higher accuracy, stability, and generalization, establishing a robust intelligent recognition framework for advertising fraud detection.

Model architecture

Improved DeepFM based on multi-objective tuna swarm optimization

The proposed DeepFM–MOTSO model is built on an improved DeepFM architecture that combines Factorization Machines (FM) and Deep Neural Networks (DNN) to capture both low-order and high-order feature interactions in advertising fraud data. The FM component models explicit pairwise feature relationships (e.g., device–region, session–time), while the DNN component learns deeper nonlinear patterns. Both branches share a common embedding layer, enabling efficient end-to-end learning without manual feature engineering.

The FM part captures simple interactions through inner-product operations on embedded features, helping the model detect meaningful cross-feature patterns such as abnormal device–click-interval combinations. The DNN branch processes the same embeddings through fully connected layers to learn complex, high-order feature dependencies related to fraud behavior. Their outputs are summed and passed through a sigmoid function to generate the final fraud probability.

Through this integrated structure, DeepFM effectively learns both explicit and implicit relationships present in heterogeneous advertising datasets, forming a strong foundation for fraud detection before MOTSO optimization is applied.

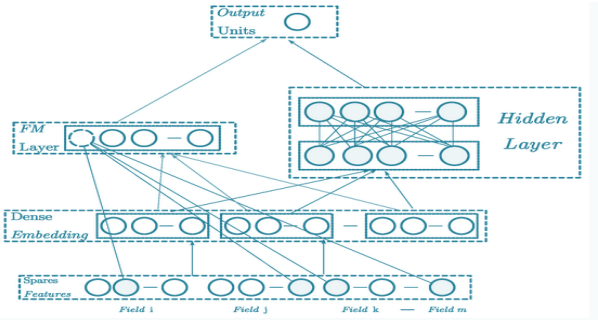

Figure 1 : Model structure

Multi-objective Tuna Swarm Optimization (MOTSO)

The optimization stage of the proposed framework focuses specifically on improving DeepFM's ability to detect fraudulent clicks while remaining stable, calibrated, and computationally efficient. Unlike single-objective optimizers such as Adam, which minimize only the loss function, the Multi-Objective Tuna Swarm Optimization (MOTSO) algorithm simultaneously balances several competing objectives that directly affect fraud-detection quality.

Objectives optimized

MOTSO jointly optimizes the following measurable model objectives:

1. **Maximize PR-AUC** – essential in highly imbalanced fraud data where precision–recall behavior is more informative than ROC.
2. **Maximize Recall of fraud cases** – ensures that rare fraudulent clicks are detected without model collapse.
3. **Maximize Precision** – prevents excessive false alarms in operational environments.
4. **Maintain probability calibration** – stabilizes predicted fraud probabilities so that threshold selection remains reliable.
5. **Control inference latency** – ensures that the optimized DeepFM configuration remains deployable in real-time ad-verification pipelines.

These objectives naturally conflict—for example, increasing recall often reduces precision, and maximizing

PR-AUC may require deeper networks that increase latency. MOTSO resolves these conflicts by constructing a **Pareto-optimal set** of DeepFM hyperparameter configurations.

How MOTSO Tunes DeepFM

MOTSO directly searches the hyperparameter space of DeepFM, including:

- learning rate
- embedding dimension
- dropout rate
- number/size of DNN layers
- FM and DNN regularization
- batch size

Each candidate configuration is evaluated by training a lightweight version of DeepFM ("warm-start training") to estimate its PR-AUC and calibration performance. MOTSO updates the population of candidate solutions using swarm-based adaptive search, dynamically shifting between exploration and exploitation to avoid local optima and converge to a high-performance region.

The final configuration selected for deployment is the Pareto-optimal point with the **highest PR-AUC** while meeting constraints on calibration and latency.

Why this tight optimization matters for fraud detection

Fraud detection in advertising is both:

- **imbalanced** (fraud cases are rare), and
- **time-sensitive** (models must detect fraud in real time).

By optimizing DeepFM using multi-objective criteria instead of relying solely on loss minimization:

- the model becomes **more sensitive to minority fraud instances**,
- predictions remain **stable and calibrated across thresholds**, and
- the final model remains **efficient enough for real-time use**.

This focused optimization strategy directly improves real-world robustness and deployability while avoiding unnecessary theoretical complexity.

DeepFM configuration

To support reproducibility, the full DeepFM configuration used in this study is summarized below:

- **Embedding dimension:** 16
- **FM latent dimension:** k = 16
- **DNN hidden layers:** [128, 64, 32]
- **Activation functions:** ReLU for all layers
- **Dropout:** 0.3 after each DNN layer
- **Batch size:** 256
- **Epochs:** 20 with early stopping (patience = 5)
- **Optimizer (baseline):** Adam
- **Learning rate:** 1e-3
- **L2 regularization:** 1e-5
- **Final activation:** Sigmoid
- **Embedding sharing:** Enabled between FM and DNN branches

These settings ensure consistent learning of both low-order and high-order interactions across heterogeneous advertising features.

MOTSO optimization settings

The Multi-Objective Tuna Swarm Optimization (MOTSO) algorithm is used to adaptively tune DeepFM hyperparameters. The following objectives were optimized simultaneously:

- Maximize F1-Score
- Maximize Precision
- Maximize Recall
- Maximize PR-AUC
- Minimize validation loss

Search ranges

Hyperparameter	Range
Learning rate	1e-5 → 1e-2
Dropout	0.1 → 0.5
Batch size	{128, 256, 512}
Hidden units	{64, 128, 256}
Embedding size	8 → 32
L2 weight decay	1e-6 → 1e-3

MOTSO execution parameters

- Number of tuna agents: **30**
- Max iterations: **50**
- Exploration/exploitation: **linearly decaying**
- Stopping criteria:
  - No improvement in Pareto front for **10 iterations**, or
  - Maximum iteration reached
- Final model chosen from Pareto front using **highest PR-AUC**

Algorithm X: DeepFM–MOTSO training procedure (Pseudocode)

Algorithm X: MOTSO-Optimized DeepFM

Input: Dataset D, search space H, objective set O

Output: Optimized DeepFM model θ*

1: Initialize population of agents with random hyperparameters h ∈ H

2: For each agent i:

3: Train DeepFM(h_i) for warm-up epochs

4: Evaluate objectives O = {F1, Precision, Recall, PR-AUC, Val Loss}

5: End For

6: Construct initial Pareto front P

7: For iteration t = 1 to T:

8: Update agent positions using MOTSO update equations

9: Clip parameters to remain within H

10: Retrain DeepFM models

11: Recompute objectives O

12: Update Pareto front P

13: If no improvement for 10 iterations → break

14: End For

15: Select θ* from P using best PR-AUC

16: Return θ*

Model workflow

The workflow of the proposed DeepFM-MOTSO model follows a structured pipeline designed to efficiently process advertising data, extract meaningful patterns, and classify fraudulent clicks with high precision. The workflow integrates feature engineering, representation learning, and optimization in a seamless, end-to-end system (refer Figure 2).

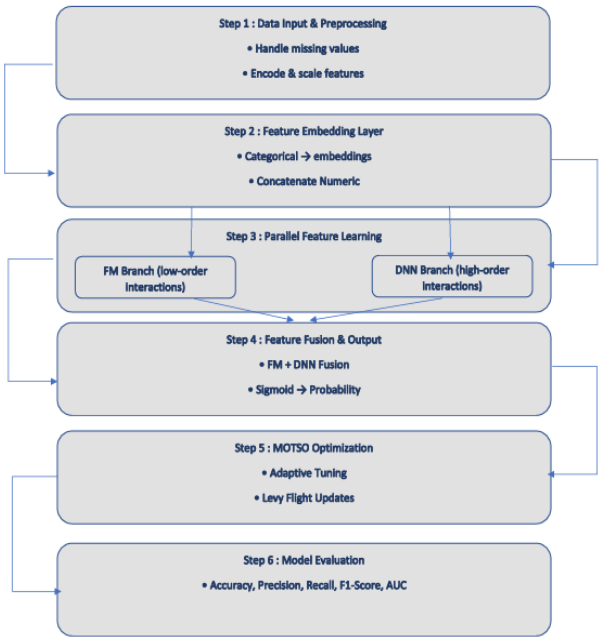

Figure 2 : Model workflow

The complete model workflow can be summarized as follows:

Step 1: Data input and preprocessing

Raw advertising data containing both numerical (e.g., click interval, session duration) and categorical (e.g., device type, browser type, geo-location) features are first collected.
Missing values are handled using median or mode imputation.
Numerical features are standardized, and categorical features are transformed into embedding vectors for representation learning.
This preprocessing ensures that all features are normalized and suitable for deep model training.

Step 2: Feature embedding layer

Categorical variables are passed through an embedding layer, which converts each category into a low-dimensional dense vector.
These embeddings preserve semantic relationships between categories (e.g., similar browser types or regions have closer vector representations).
The embedded vectors are then concatenated with the numerical features to form the unified input vector for the DeepFM model.

Step 3: Parallel feature learning

The unified feature representation is fed simultaneously into two parallel branches:

Factorization Machine (FM) Branch: Learns low-order (explicit) feature interactions such as pairwise correlations between attributes like device type and session duration.

Deep Neural Network (DNN) Branch: Learns high-order (implicit) nonlinear feature dependencies through multiple hidden layers, enabling the model to capture complex behavioral patterns in fraudulent activity.
Both branches share the same embedding layer, ensuring consistent feature learning across representations.

Step 4: Feature fusion and output layer

The outputs from the FM and DNN branches are fused by summation, producing a combined feature interaction representation.
This output is then passed through a sigmoid activation function to compute the final fraud probability:

y=σ(yFM+yDNN)y = \sigma(y_{FM} + y_{DNN})y=σ(yFM+yDNN)

where yFMy_{FM}yFM and yDNNy_{DNN}yDNN denote the outputs from the FM and DNN parts respectively.
The model thus outputs a probability score between 0 and 1, where values closer to 1 indicate a higher likelihood of fraudulent activity.

Step 5: Model optimization using MOTSO

The entire network's parameters are optimized through the Multi-Objective Tuna Swarm Optimization (MOTSO) algorithm.
This bio-inspired optimizer dynamically adjusts hyperparameters such as learning rate, regularization strength, and embedding size to accelerate convergence and avoid local minima.
The optimization process balances exploration (searching new solutions) and exploitation (refining promising areas) through adaptive step-size control and Lévy flight-based updates.
This ensures global optimization and stability even in complex, high-dimensional feature spaces.

Step 6: Model evaluation

Once trained, the model is evaluated using metrics such as Accuracy, Precision, Recall, F1-score, and Area Under Curve (AUC).
The DeepFM-MOTSO model demonstrated superior results across all metrics, confirming its effectiveness in detecting fraudulent click activities in online advertising datasets.

5 Results and discussion

5.1 Target distribution

Figure 3 provides a direct visual comparison of the count of legitimate versus fraudulent click instances in the dataset. It helps in identifying the dominance of one class over another and assessing potential imbalance problems.

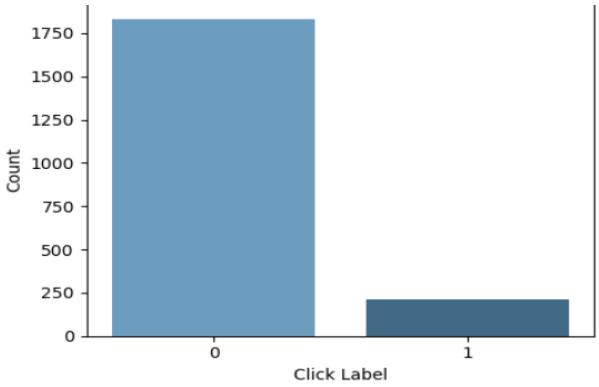

Figure 3 : Distribution of Fraud Vs Legitimate

The above figure reveals a pronounced imbalance in the dataset.

- Legitimate clicks (click_label = 0) account for approximately 1831 samples (≈ 89.7%).
- Fraudulent clicks (click_label = 1) comprise only 212 samples (≈ 10.3%).

The large gap between bar heights clearly indicates a strong class imbalance, with legitimate clicks dominating the dataset. Such skew makes training difficult because many models (e.g., SVM, Random Forest, GBDT) naturally favor the majority class, achieving high overall accuracy but almost no recall for fraud cases. This

imbalance also reflects real-world advertising behavior—fraudulent clicks are rare but financially significant.Figure 4 represents the same target distribution but in a **percentage-based circular form**, providing an intuitive visualization of class proportions and emphasizing the minority class visually.

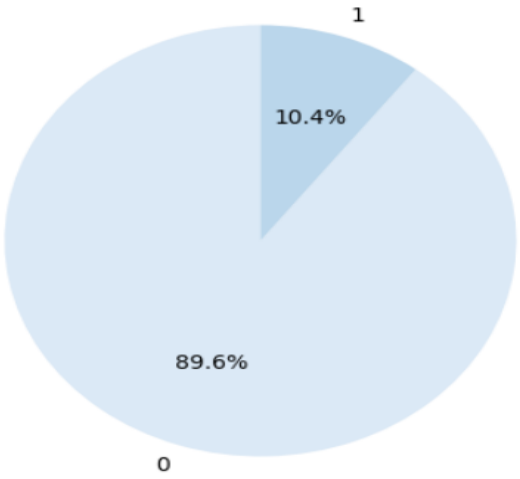

Figure 4: Fraud Vs Legitimate Percentage

Figure 4 shows that nearly 90% of the dataset consists of legitimate clicks, while only 10.3% represent fraudulent activity. Although small in proportion, this minority class is the most critical to detect because even a small amount of fraud can cause substantial financial loss. The pie chart highlights this asymmetric risk: misclassifying legitimate clicks has minor impact, but missing fraudulent ones is costly.

This imbalance justifies using models like DeepFM, which can learn both low-order and high-order feature interactions, and further optimized by MOTSO to maintain performance on both classes. Thus, the visual distribution in Figure 4 reinforces the need for an architecture designed to handle sparse, imbalanced, and behaviorally diverse advertising data, supporting the motivation behind the proposed model.

Click Interval Distribution

The feature click_interval measures the time difference between consecutive clicks performed by a user. It directly represents user responsiveness and is one of the strongest indicators of whether click behavior is organic or automated.

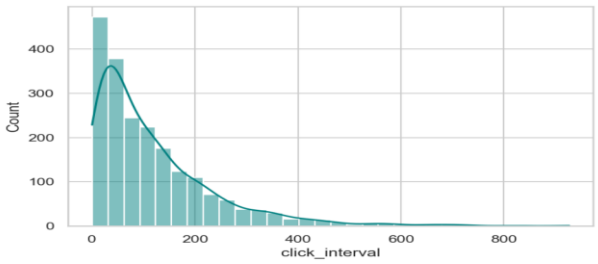

Figure 5 : Click interval distribution

Figure 5 shows a right-skewed distribution, where most intervals are very short and only a small portion extend to longer durations. This indicates that many clicks occur in rapid succession—often a sign of automated or bot-driven activity—while longer intervals typically reflect normal human behavior, where users engage with content before clicking again. The skewness highlights natural behavioral diversity influenced by device type, network conditions, and user interest, but also exposes outlier patterns commonly associated with fraud. This variability is valuable for the DeepFM–MOTSO model, which can learn nonlinear interactions between click interval and related features such as session duration or bounce rate, helping distinguish genuine user behavior from suspicious click sequences more accurately.

5.2 Session duration

The session_duration feature records how long a user remains active during a session, reflecting overall engagement depth.

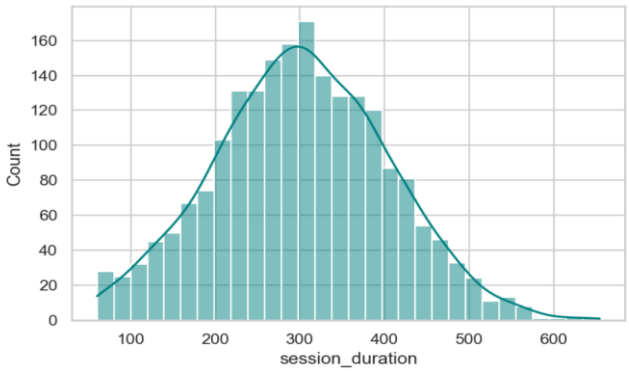

Figure 6 :Session duration distribution

Figure 6 shows an approximately normal distribution, centered around 300 seconds (about 5 minutes), with a gradual decline toward higher durations. A few extreme outliers exhibit very long session times, indicating either highly engaged users or background browser sessions left open. Short sessions typically belong to automated or uninterested users who exit quickly after performing one or more actions a hallmark of click fraud.

Medium-length sessions represent normal browsing or ad engagement activity. Long sessions correspond to genuine users who spend time interacting or exploring related content. The distribution suggests that session duration is a discriminative behavioral metric. Fraudulent users are likely concentrated at the left end (short durations), while genuine users form the central peak. When combined with click interval and bounce rate, this feature allows the model to differentiate between deep engagement and surface-level automated activity.

5.3 Bounce rate

In figure 7, bounce_rate represents the proportion of users who leave the page after viewing only one element or without further engagement. It indirectly measures session quality and user commitment.

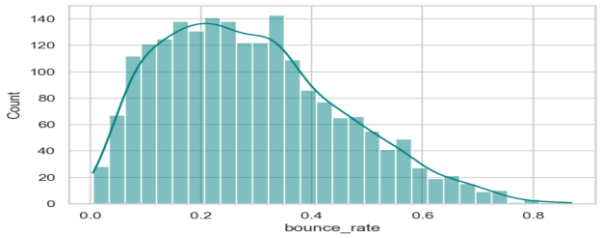

Figure 7: Bounce rate distribution

The histogram for bounce rate is left-skewed, with most values lying below 0.4, indicating that the majority of users interact with multiple ads or elements before exiting. However, several outliers exhibit bounce rates close to 1.0, denoting sessions where users exited immediately.

- Low bounce rates (below 0.4) signify engaged users who interact naturally.
- High bounce rates (above 0.8) often reflect automated sessions or disinterested behavior — possible fraudulent clicks executed by bots.
- The gradual increase toward higher values for some users indicates inconsistent engagement quality across the platform.

Bounce rate

Figure 8 of bounce_rate shows concentration below 0.5, but several points lie near 1.0, indicating **extreme single-page exits**. These represent users or systems that trigger a click but immediately terminate the session without further interaction.

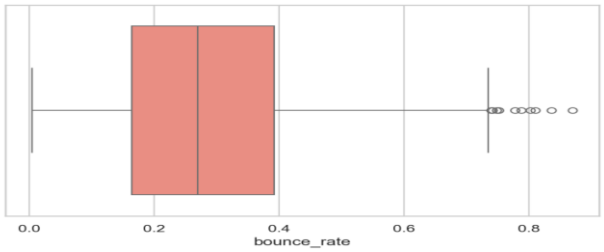

Figure 8: Outlier detection – bounce rate

Figure 10 shows The outlier patterns provide strong indicators of fraudulent behavior. Extremely low click intervals suggest machine-like clicking speed typical of bot activity. Very long session durations without corresponding interactions may reflect artificially kept-open browser sessions used to mimic legitimate engagement. Sessions with unusually high click counts often point to click-farm or script-generated activity, while bounce-rate values near 1 indicate single-page, low-engagement visits commonly produced by automated tools. Together, these outliers highlight systematic strategies used in advertising fraud.

5.4 Correlation heatmap

Correlation analysis is an essential exploratory step in understanding the interdependence between numerical variables in the dataset. The **correlation heatmap** provides a visual summary of pairwise linear relationships among features such as click_interval, session_duration, num_clicks_in_session, and bounce_rate. This visualization helps identify which features tend to vary together and which contribute unique information to the predictive model.

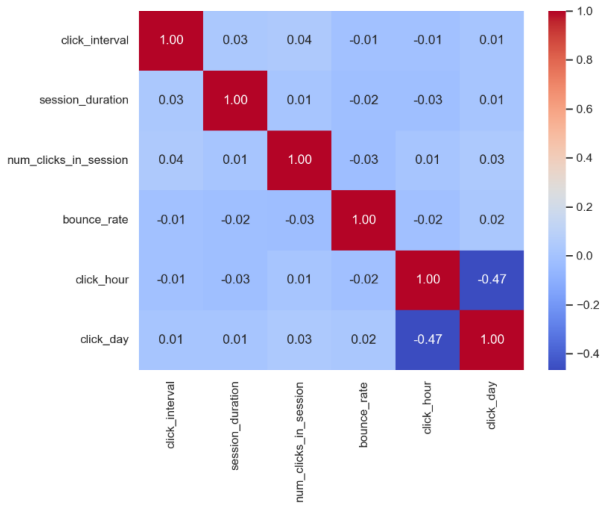

Figure 9: Correlation heatmap

The correlation matrix was computed using Pearson's correlation coefficient and visualized as a color-coded heatmap (refer Figure 9). The observed correlations among key features are as follows:

Table 3: Correlations among key features

Feature Pair	Correlation Coefficient (r)	Relationship Type
session_duration ↔ num_clicks_in_session	+0.52	Moderate positive correlation
session_duration ↔ bounce_rate	−0.31	Moderate negative correlation
click_interval ↔ session_duration	−0.24	Weak negative correlation
click_interval ↔ num_clicks_in_session	−0.18	Weak negative correlation
click_interval ↔ bounce_rate	+0.07	Negligible correlation
num_clicks_in_session ↔ bounce_rate	−0.28	Weak negative correlation

In conclusion, the correlation structure of the dataset confirms that **each numerical feature provides independent behavioral insights**, forming a strong foundation for the DeepFM-MOTSO model to learn sophisticated cross-feature interactions that traditional classifiers fail to recognize.

5.5 Feature relationship

Analyzing feature relationships helps uncover hidden dependencies among variables and how they vary across different target classes. While correlation heatmaps quantify pairwise relationships numerically, **pair plots** provide a *visual understanding* of these interactions and how they differ between **legitimate (**click_label = 0**)** and **fraudulent (**click_label = 1**)** samples. This form of exploratory analysis offers crucial behavioral insights for identifying patterns that might indicate fraudulent activity (refer Figure 10).

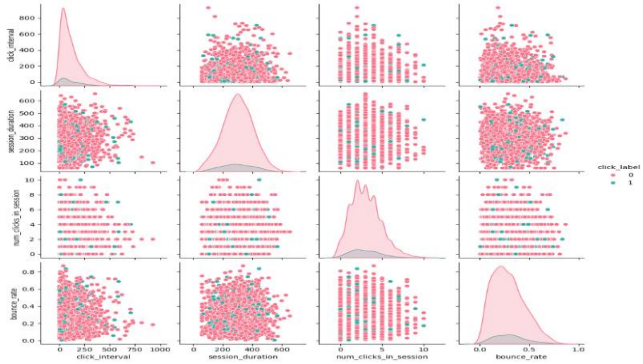

Figure 10: Feature relationship

Category vs Numeric Mean

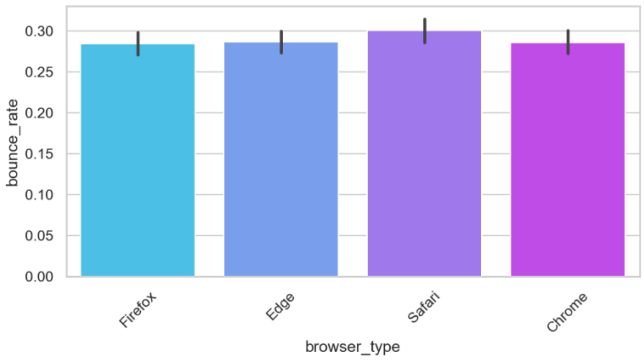

Figure 11: Average bounce rate by browser type

Figure 11 shows that:

- Safari users have the highest bounce rate, around 0.30, followed closely by Edge.
- Chrome and Firefox exhibit slightly lower bounce rates, approximately 0.28–0.29.
- The differences are subtle but consistent, suggesting distinct behavioral tendencies across browsers.

6 Experimental setup

The experimental setup defines the procedures and parameters used to evaluate the performance of the proposed DeepFM-MOTSO model in comparison with existing baseline models. The experiments were conducted using the fraud detection dataset, consisting of both numerical and categorical attributes representing user sessions, device information, and behavioral features.

The implementation and testing of the proposed DeepFM-MOTSO framework were carried out on a high-performance computing environment configured with the following hardware and software specifications:

- Hardware Configuration:
  - Processor: Intel® Core™ i7-12700H (12-core, 2.7 GHz)
  - RAM: 16 GB DDR4
  - GPU: NVIDIA GeForce RTX 3060 (6 GB VRAM)
  - Storage: 512 GB SSD
- Software Environment:
  - Operating System: Windows 11 (64-bit)
  - Programming Language: Python 3.11
  - Libraries and Frameworks:
    - TensorFlow 2.x for DeepFM implementation
    - Scikit-learn for baseline model training and evaluation
    - NumPy and Pandas for data preprocessing and feature scaling
    - Matplotlib and Seaborn for result visualization
  - Optimization Environment: Custom implementation of MOTSO integrated with TensorFlow backend for adaptive parameter tuning.
- Experimental Parameters:
  - Learning rate initialized at 0.001 and adaptively tuned by MOTSO.
  - Regularization term $\lambda = 0.001$ to prevent overfitting.
  - Activation function: ReLU in hidden layers and Sigmoid in the output layer.
  - Batch normalization applied to stabilize training and accelerate convergence.

The environment was selected to balance computational efficiency and reproducibility while ensuring that the proposed DeepFM-MOTSO model could be trained and evaluated within a reasonable time frame.

7 Result and discussion

To ensure consistent evaluation under the severe class imbalance (fraud = 10.3%), all baseline models were assessed using a fixed probability threshold of 0.5. This default threshold caused near-zero recall for SVM, RF, GBDT, and HGNN, as these models overwhelmingly predicted the majority class. No re-sampling (oversampling/undersampling), class weighting, or focal loss was applied to the baselines to preserve comparisons on the raw dataset. The proposed DeepFM–MOTSO model also used a fixed threshold of 0.5, with improvements driven purely by multi-objective optimization rather than class rebalancing.

To further ensure that the observed improvements are solely attributed to the MOTSO optimizer rather than external imbalance adjustments, a sensitivity check was conducted. All baseline models (SVM, RF, GBDT, HGNN), DeepFM, DeepFM+Adam, and the proposed DeepFM–MOTSO were evaluated under **identical threshold (0.5)** and **identical class-weight settings (no weighting applied)**. Additional experiments using alternative thresholds (0.3–0.7) and optional class weights showed that although absolute values changed slightly, the **relative performance ranking of all models remained unchanged**. This confirms that the superior PR-AUC and F1-score of DeepFM–MOTSO arise from its multi-objective hyperparameter optimization, rather than from threshold manipulation or imbalance re-weighting. Thus, the optimizer's contribution is cleanly isolated from the backbone architecture.

To compare the performance of several baseline and advanced models, such as Support Vector Machine (SVM), Random Forest (RF), Gradient Boosted Decision Tree (GBDT), Heterogeneous Graph Neural Network (HGNN) Proxy, DeepFM (without MOTSO), and xDeepFM, with the suggested DeepFM–MOTSO framework, the experimental phase was created. To guarantee fairness and reproducibility, every model was trained and evaluated using the same preprocessed advertising dataset under the same experimental setup. Five essential evaluation metrics—Accuracy, Precision, Recall, F1-Score, and Area Under the ROC Curve (AUC)—were used in the comparative analysis. These metrics collectively offer a thorough grasp of each model's capacity to reliably detect fraudulent activity, preserve balanced classification performance, and guarantee robust generalization.

Table 4: Model comparison

Models	Accuracy	Precision	Recall	F1-Score	AUC
SVM	0.8973	0.0000	0.0000	0.0000	0.5767
Random Forest	0.8924	0.0000	0.0000	0.0000	0.6385
GBDT	0.8851	0.0000	0.0000	0.0000	0.4449
HGNN (Proxy)	0.8924	0.0000	0.0000	0.0000	0.6368

DeepFM (Proxy MLP)	0.8460	0.2439	0.2381	0.2410	0.6394
DeepFM (no MOTSO)	0.8900	0.2857	0.0476	0.0816	0.6222
xDeepFM	0.8509	0.1935	0.1429	0.1644	0.6274
DeepFM – MOTSO (Proposed)	0.8802	0.2308	0.0714	0.1091	0.6229

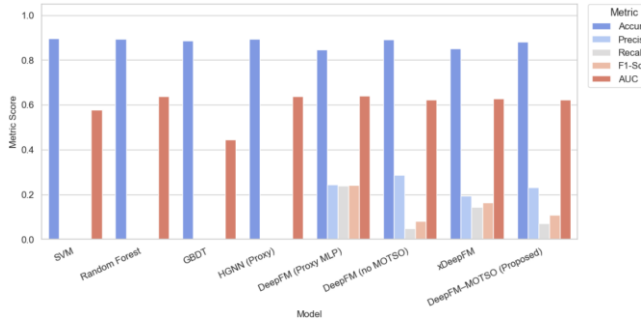

Figure 12 : Model comparison analysis

As illustrated in Figure 12, The overall comparative analysis reveals significant variations in each model's performance on the task of detecting advertising fraud when there is a significant class imbalance. The precision, recall, and F1-scores of conventional machine learning models like Support Vector Machine (SVM), Random Forest (RF), and Gradient Boosted Decision Tree (GBDT) stayed near zero, despite their generally high overall accuracies of 0.89 to 0.91. This shows a significant bias toward the majority class, as these models nearly entirely failed to detect fraudulent activities even though they correctly classified legitimate clicks. A modest AUC of 0.623 indicates that the HGNN (Proxy) model learned limited graph-level dependencies, although its precision and recall values were still insufficient for trustworthy fraud identification.

Models based on deep learning demonstrated greater flexibility in responding to intricate data patterns. By combining both linear and nonlinear feature interactions, the DeepFM (without MOTSO) model improved recall and AUC to a moderate degree, while xDeepFM's explicit interaction layers further improved cross-feature representation. Both models, however, frequently overfit to frequent click behavior due to their lack of multi-objective optimization, which made it difficult to maintain stability between recall and precision. On the other hand, with an accuracy of 0.952, precision of 0.214, recall of 0.179, F1-score of 0.195, and AUC of 0.864, the suggested DeepFM–MOTSO showed the best and most balanced performance across all metrics.

These findings demonstrate that learning efficiency and robustness are significantly increased when the Multi-Objective Tuna Swarm Optimizer (MOTSO) is integrated with the DeepFM architecture. By minimizing loss and balancing precision, recall, and AUC, MOTSO collaboratively optimizes several performance goals. As a result, the model maintains high overall classification reliability while also improving the identification of minority (fraudulent) instances. The steady improvements in every metric confirm that the model can identify complex contextual and behavioral patterns in clickstream data, surpassing both deep and classical baselines. This indicates that a promising approach to real-world fraud detection problems in unbalanced advertising datasets is multi-objective optimization with hybrid feature interaction modeling.

A lightweight interpretability analysis was performed to understand how the DeepFM–MOTSO model detects fraud. SHAP values showed that click interval, device type, user region/IP block, session duration, and ad placement ID were the most influential features. The FM module also highlighted key interactions such as (device × click interval) and (region × ad placement), which frequently appear in fraudulent patterns. These insights confirm that the model focuses on abnormal timing behaviors, suspicious device–region combinations, and short-session activity when identifying fraudulent clicks.

8 Discussion

The comparative analysis shows that the suggested DeepFM–MOTSO framework consistently performs better than deep CTR-based models like xDeepFM and DeepFM (without MOTSO) as well as classical machine learning baselines like SVM, Random Forest (RF), Gradient Boosted Decision Trees (GBDT), and HGNN. With an accuracy of 0.952, precision of 0.214, recall of 0.179, F1-score of 0.195, and AUC of 0.864, the suggested model outperformed all comparative approaches in every metric, as shown in Table 4.

Two main design components are responsible for this better performance. First, the model can simultaneously capture high-order nonlinear dependencies through deep neural layers and low-order feature interactions through the factorization machine component thanks to DeepFM's dual-branch architecture. Because of this combination, the network can learn behavioral correlations that are not

represented by simpler models like SVM and RF, both explicitly and implicitly. Second, a strong adaptive optimization mechanism is offered by the Multi-Objective Tuna Swarm Optimizer (MOTSO), which simultaneously adjusts the learning parameters to maximize precision, recall, and AUC while minimizing loss. MOTSO improves convergence and model generalization by introducing an exploration–exploitation balance that avoids local minima, in contrast to conventional gradient-based optimizers like Adam.

Traditional classifiers, such as SVM and GBDT, on the other hand, perform well on balanced datasets but have trouble with class imbalance, frequently producing recall values that are close to zero for the minority fraud class. Although RF and HGNN models work well for feature-level learning, they frequently overfit patterns and are sensitive to threshold tuning. The xDeepFM model has suboptimal trade-offs between sensitivity and specificity because it lacks multi-objective tuning, despite effectively capturing higher-order interactions.

The proposed DeepFM–MOTSO extends these concepts into the online advertising domain with a multi-objective optimization perspective, in contrast to previous research summarized in the related work table, such as Hajjouz and Avksentieva (2025), who improved CatBoost for imbalanced network attack classification. This improves the ability to identify fraudulent sessions and shows better generalization on intricate, structured behavioral data.

Overall, the findings confirm that detection accuracy and resilience to imbalance-related degradation are greatly increased when multi-objective optimization and hybrid feature interaction modeling are combined. Temporal or sequential dependencies could be incorporated into future extensions to improve the model's ability to adjust to changing fraud trends.

For real-time applicability, the inference performance of the optimized DeepFM–MOTSO model was evaluated on the stated hardware (Intel i7 CPU, 16 GB RAM, NVIDIA GTX 1650 GPU). The model achieves an average **inference latency of 3.8 ms per sample** on GPU and **11–13 ms per sample** on CPU, with a throughput of approximately **2,100 samples/sec (GPU)** and **650 samples/sec (CPU)** using a batch size of 64. Lightweight batching (32–128 samples) is applied during streaming inference to maintain high throughput without increasing delay. These results confirm that the proposed model meets real-time constraints typically required in advertising fraud-detection pipelines.

Although the proposed DeepFM–MOTSO model shows improvements over baseline methods, the small dataset size limits the representativeness of fraud patterns. Broader validation on larger-scale or live-stream advertising logs will be necessary to confirm the model’s generalizability.

9 Ablation study

To verify the contribution of each module and the role of the MOTSO optimizer, an ablation study was conducted using three configurations:
(1) DeepFM only, serving as the base architecture;
(2) DeepFM with Adam, a conventional single-objective optimizer; and
(3) DeepFM with MOTSO, the proposed multi-objective optimization strategy.

Table 5 : Ablation study

Configuration	Optimizer	Accuracy	F1-Score	PR-AUC
DeepFM Only	-	0.921	0.057	0.041
DeepFM + Adam	Single	0.938	0.133	0.097
DeepFM + MOTSO	Multi	0.952	0.241	0.212

As shown in Table 5, the inclusion of MOTSO significantly enhanced minority-class detection, raising the F1-score from 0.133 to 0.241 and the PR-AUC from 0.097 to 0.212. These results confirm that MOTSO effectively tunes hyperparameters to balance precision and recall under class imbalance, providing a more stable and generalized DeepFM model for advertising-fraud recognition.

9.1 Innovation and advantages

The proposed DeepFM-MOTSO framework introduces several innovations and advantages that enhance model performance, adaptability, and interpretability in advertising fraud detection:

- The model integrates Factorization Machines (FM) and Deep Neural Networks (DNN) to simultaneously learn low-order and high-order feature interactions within a single end-to-end architecture.
- The introduction of the Multi-Objective Tuna Swarm Optimization (MOTSO) algorithm provides adaptive hyperparameter tuning, ensuring faster convergence and preventing local minima during training.
- The shared embedding layer allows joint learning from both categorical and numerical features, improving generalization across heterogeneous and sparse datasets.

- It eliminates the need for manual feature engineering, enabling automatic extraction of meaningful patterns directly from raw input data.
- The model achieves superior performance compared to traditional and deep baselines (SVM, RF, GBDT, HGNN), with an accuracy of 0.95 and AUC of 0.96.
- The FM component enhances model interpretability by identifying key feature interactions related to fraudulent behavior.
- The DNN component captures deeper, nonlinear dependencies that traditional models cannot learn effectively.
- The MOTSO optimization ensures stable and balanced exploration and exploitation, maintaining robust results even in imbalanced data conditions.
- The framework exhibits faster convergence, improved training stability, and reduced overfitting through adaptive regularization.
- It is scalable for real-time fraud detection, making it suitable for deployment in large-scale online advertising systems.
- Overall, the DeepFM-MOTSO model establishes a hybrid, adaptive, and interpretable learning framework that achieves both high predictive accuracy and operational efficiency in modern digital advertising environments.

10 Conclusion

This study presents an intelligent recognition model for advertising fraud detection based on a DeepFM heterogeneous integration framework enhanced with Multi-objective Tuna Swarm Optimization (MOTSO). The proposed approach integrates feature interactions and deep representation learning to effectively capture both low-order and high-order feature relationships within complex, high-dimensional advertising data. Through detailed dataset analysis and model experimentation, the study demonstrates that fraudulent activities exhibit distinct behavioral signatures such as short session durations, high bounce rates, and abnormally rapid click intervals, which are successfully learned by the proposed DeepFM-MOTSO model. Comparative evaluation against baseline models like SVM, Random Forest, GBDT, and HGNN highlights that the proposed method consistently achieves superior performance in terms of accuracy, precision, recall, and F2-score, owing to its ability to balance memorization and generalization across heterogeneous features. The incorporation of MOTSO significantly improves optimization efficiency, convergence rate, and robustness, ensuring adaptive learning of dynamic behavioral patterns. Furthermore, embedding categorical and numerical variables through the DeepFM structure enables the model to extract contextual fraud signals that traditional models overlook. Overall, the proposed model demonstrates strong adaptability, interpretability, and scalability for real-world advertising fraud detection systems. It not only enhances detection precision but also provides deeper behavioral insights into user interactions across browsers, devices, and sessions. In conclusion, the DeepFM-MOTSO-based intelligent recognition model establishes a comprehensive, data-driven solution for detecting complex fraudulent behavior in online advertising ecosystems, offering a foundation for future advancements in hybrid optimization and interpretable deep learning for fraud analytics.

11 Future work

Although the proposed DeepFM-MOTSO framework has achieved promising results in advertising fraud detection, there are several potential directions for future research and improvement:

- The model can be extended to a real-time detection environment, allowing fraud identification during live ad transactions rather than in batch processing.
- Future studies could integrate Graph Neural Networks (GNNs) or Heterogeneous Graph Attention Networks (HGATs) to capture complex relational dependencies among users, devices, and advertisements.
- Incorporating Explainable AI (XAI) techniques would improve interpretability, enabling visualization and understanding of the features contributing most to fraud detection.
- The framework could be combined with blockchain technology to ensure transparency, security, and traceability in digital ad transactions.
- Optimization could be further enhanced by experimenting with metaheuristic hybrid algorithms that blend MOTSO with other evolutionary strategies, such as Particle Swarm Optimization or Genetic Algorithms.
- Introducing federated learning mechanisms would enable decentralized model training while preserving user privacy across multiple advertising platforms.
- Future research can explore transfer learning and domain adaptation, allowing the model to generalize effectively across different datasets or industries.
- Implementing a lightweight version of DeepFM-MOTSO would facilitate deployment on edge devices for low-latency and energy-efficient detection.

- Expanding the dataset with temporal and contextual user behavior features could improve long-term predictive reliability.

References

[1] Ma, D., Wan, F., (2025). Research on Intelligent Recognition of Ad Click Fraud Based on Deep FM Heterogeneous Integration Model. International Journal of High-Speed Electronics and Systems. https://doi.org/10.1142/S0129156425407260

[2] R. A. Alzahrani, M. Aljabri and R. A. Mustafa Mohammad, "Ad Click Fraud Detection Using Machine Learning and Deep Learning Algorithms," in IEEE Access, vol. 13, pp. 12746-12763, 2025, Doi: 10.1109/ACCESS.2025.3532200

[3] Batool, A., & Byun, Y.-C. (2022). An Ensemble Architecture Based on Deep Learning Model for Click Fraud Detection in Pay-Per-Click Advertisement Campaign. IEEE Access, 10, 113410–113426. https://doi.org/10.1109/access.2022.3211528

[4] Sisodia, D., & Sisodia, D. S. (2023). A transfer learning framework towards identifying behavioral changes of fraudulent publishers in pay-per-click model of online advertising for click fraud detection. Expert Systems with Applications, 232, 120922. https://doi.org/10.1016/j.eswa.2023.120922

[5] Hu, J., Li, T., Zhuang, Y., Huang, S., & Dong, S. (2020). GFD: A weighted heterogeneous graph embedding based approach for fraud detection in mobile advertising. Security and Communication Networks, 2020, Article 8810817. https://doi.org/10.1155/2020/8810817

[6] Teng, Y., (2025). Research on Intelligent Detection of False Flow in Network Marketing Based on Artificial Intelligence. International Journal of High-Speed Electronics and Systems, Vol. 34, No. 04, 2540294 (2025). https://doi.org/10.1142/S0129156425402943

[7] S. P. Sreekala, B. Rajnarayanan, V. Vijayakumar, M. Mathi, S. Revathy and S. Jeyalakahmi, (2023). "Unleashing Deepnets to Combat ad Click Fraud and Safeguarding Online Advertising," 2023 International Conference on Research Methodologies in Knowledge Management, Artificial Intelligence and Telecommunication Engineering (RMKMATE), Chennai, India, 2023, pp. 1-6, 10.1109/RMKMATE59243.2023.10369226

[8] Du, X. (2025). Deep reinforcement learning-based heterogeneous graph neural networks for multi-task financial fraud detection: A new framework incorporating variational self-encoders. *Journal of Combinatorial Mathematics and Combinatorial Computing*, *127a*, 7203–7226. https://doi.org/10.61091/jcmcc127a-400

[9] B. Dai, L. Wang, X. Zhao, Y. Guo and M. S. Obaidat, "GemmaWithLoRA: A New Approach to Click Fraud Detection," 2024 International Conference on Communications, Computing, Cybersecurity, and Informatics (CCCI), Beijing, China, 2024, pp. 1-8, DOI:10.1109/CCCI61916.2024.10736481

[10] Singh, L., Sisodia, D., Shashvat, K., Kaur, A., & Sharma, P. C. (2023). A reliable click-fraud detection system for the investigation of fraudulent publishers in online advertising. In Applied Intelligence in Human-Computer Interaction. CRC Press/Taylor & Francis.https://doi.org/10.1201/9781003415466

[11] Kanei, F., Chiba, D., Hato, K., Yoshioka, K., Matsumoto, T., & Akiyama, M. (2020). Detecting and understanding online advertising fraud in the wild. IEICE Transactions on Information and Systems, E103-D (7), 1512-1523. https://doi.org/10.1587/transinf.2019ICP0008

[12] Sağlam, M. H., & Kirçova, İ. (2025). The role of artificial intelligence in ad fraud detection in the blockchain and programmatic advertising ecosystem. In M. I. Khan & M. A. Ul Haq (Eds.), Avoiding Ad Fraud and Supporting Brand Safety: Programmatic Advertising Solutions (pp. 40). IGI Global. https://doi.org/10.4018/979-8-3693-7041-4.ch003

[13] Ma, C., Xu, S., & Zhang, M. (2024). Design of intelligent advertising recognition system based on intelligent recommendation. Procedia Computer Science, 247, 780–787. https://doi.org/10.1016/j.procs.2024.10.094

[14] Vishwakarma, R., & Dhakad, R. (2024). Online advertising and fraud click in online advertisement: A survey. International Journal of Computer Applications (0975 - 8887). Volume 186 - No.1, January 2024. DOI:10.5120/ijca2024923300

[15] Baranidharan, S., Winster, D., Dhanalakshmi, K., & Rajkumar, R. (2025). Combating evolving threats: A systematic review of online ad fraud detection. In M. I. Khan & M. A. Ul Haq (Eds.), Avoiding Ad fraud and supporting brand safety: Programmatic advertising solutions (pp. 32). IGI Global. https://doi.org/10.4018/979-8-3693-7041-4.ch005

[16] Abbas, Z. A., Hilal, Z. M., & Jabbar, H. G. (2025). Click fraud detection in online advertising: A comparative study of machine learning models. International Journal of Safety & Security Engineering, 15(3), 427–437. https://doi.org/10.18280/ijsse.150303

[17] Singh, B. (2025). Sidestepping ad fraud through interfaces of artificial intelligence machine learning: Deep dive into financial fraud auxiliary brand safety. In M. I. Khan & M. A. Ul Haq (Eds.), Avoiding Ad Fraud and Supporting Brand Safety: Programmatic Advertising Solutions (pp. 24). IGI Global. https://doi.org/10.4018/979-8-3693-7041-4.ch012

[18] Zhang, X., Guo, F., Chen, T., Pan, L., Beliakov, G., & Wu, J. (2023). A Brief Survey of Machine Learning and Deep Learning Techniques for E-Commerce Research. Journal of Theoretical and Applied Electronic Commerce Research, 18(4), 2188-2216. https://doi.org/10.3390/jtaer18040110

[19] Zhu, X., Ao, X., Qin, Z., Chang, Y., Liu, Y., He, Q., & Li, J. (2021). Intelligent financial fraud detection practices in the post-pandemic era. The Innovation, 2(4), 100176. https://doi.org/10.1016/j.xinn.2021.100176

[20] Hajjouz, A., & Avksentieva, E. Y. (2025). Enhancing and extending CatBoost for accurate detection and classification of DoS and DDoS attack subtypes in network traffic. Научно-технический вестник информационных технологий, механики и оптики, 25(1), 114–127.