European Journal of AI, Computing & Informatics

Vol. 2 No. 1 2026

Article Open Access

Graph Neural Network-Based Governance of Fraudulent Traffic: Detecting and Suppressing Fake Impressions and Clicks in Digital Platforms

Wenwen Liu [1,*]

[1] University of Washington, Seattle, WA, USA

* Correspondence: Wenwen Liu, University of Washington, Seattle, WA, USA

Received: 21 December 2025
Revised: 02 February 2026
Accepted: 16 February 2026
Published: 21 February 2026

Abstract: Digital advertising platforms rely heavily on impression and click metrics to quantify user engagement, but fraudulent traffic-including fake impressions generated by bot farms and click farming-causes over $68 billion in annual losses for global advertisers. Traditional anti-fraud methods, such as rule-based engines and isolated machine learning (ML) models, fail to capture the complex relational patterns among users, advertisements, and traffic sources, leading to high false positive rates and poor adaptability to evolving fraud tactics. To address these gaps, this study proposes a Dynamic Graph Neural Network-based Fraud Traffic Governance Framework (DGNN-FTG) that integrates real-time detection and adaptive suppression in a unified pipeline. First, we construct a User-Advertisement-Medium (UAM) dynamic interaction graph that encodes spatio-temporal behavioural features and relational dependencies, overcoming the limitations of static, single-node modelling. Second, we design a Spatio-Temporal Dynamic GNN (ST-DGNN) for fraud detection, which incorporates temporal attention to track behaviour evolution and spatial attention to identify anomalous relational clusters. Third, we develop an adaptive suppression module that adjusts traffic filtering thresholds based on detection confidence, combined with a false-positive compensation mechanism to minimize disruption to legitimate users. Validated on a real-world dataset from a leading e-commerce advertising platform, DGNN-FTG achieves a fraud detection F1-score of 92.8%, outperforming traditional XGBoost and static GNN models by 18.3% and 11.5%, respectively. The suppression module reduces fraudulent traffic by 89.2% while maintaining a false positive rate of only 2.1%, balancing anti-fraud efficacy and user experience. This framework provides a scalable, real-time solution for digital platforms to combat fraudulent traffic and safeguard the integrity of the advertising ecosystem. Notably, DGNN-FTG exhibits strong generalization across platform types, achieving an F1-score of over 90% when deployed on both e-commerce and social media advertising systems, with minimal retraining required for cross-platform adaptation. For small and medium-sized advertisers, the framework's low false positive rate translates to a 22.3% reduction in wasted ad expenditure, directly improving their return on investment (ROI) in digital marketing campaigns. Additionally, the dynamic graph update mechanism enables DGNN-FTG to detect emerging fraud tactics-such as AI-generated fake user behaviour-within 48 hours of their appearance, far faster than the 7-10 day response time of traditional anti-fraud systems.

Keywords: Fraudulent Traffic Governance; Graph Neural Network; Fake Impressions; Click Fraud; Spatio-Temporal Dynamic Graph; Adaptive Suppression

1. Introduction

The global digital advertising market is projected to reach $856 billion by 2026, with impression and click-through rates serving as core metrics for ad performance evaluation. However, the proliferation of fraudulent traffic-driven by automated bots, click farms, and compromised devices-has become a systemic threat to the ecosystem [1]. Fake impressions, generated by non-human traffic that mimics user browsing behaviour, and fake clicks, initiated by coordinated bot networks to inflate ad engagement, distort performance data and erode advertiser trust. According to the 2024 Global Ad Fraud Report, 22.3% of all digital ad impressions are fraudulent, with e-commerce platforms suffering the highest fraud rates due to the direct link between clicks and sales commissions. A 2025 case study of a top Chinese fast-fashion e-commerce platform illustrates the severity of the problem: the platform lost over $12 million in a single quarter due to fake clicks targeting its affiliate marketing program, with fraudsters using a network of 50,000 compromised mobile devices to generate 1.2 billion fake clicks [2]. These fraudulent interactions not only wasted advertiser budgets but also skewed the platform's algorithmic recommendation system, leading to a 15.7% drop in legitimate user engagement over three months. For small and medium-sized enterprises (SMEs) that rely on digital advertising for customer acquisition, fraudulent traffic can be catastrophic-over 30% of SMEs surveyed in the 2024 Digital Advertising Trust Report reported halting all digital ad spending after discovering that over 50% of their clicks were fake [3].

Existing anti-fraud solutions face three critical challenges:

1. Relational Blindness: Rule-based engines (e.g., blocking IPs with abnormal click frequencies) and isolated ML models (e.g., logistic regression on user attributes) treat each user or click as an independent entity, ignoring the relational patterns that define fraud-such as bot clusters sharing identical browsing paths or targeting the same set of ads.
2. Static Modeling Limitation: Most GNN-based fraud detection models use static graphs, failing to capture the temporal evolution of fraud tactics (e.g., bots switching from fixed IPs to dynamic proxy pools over time).
3. Detection-Suppression Decoupling: Traditional systems separate detection and suppression, applying rigid filtering rules (e.g., blocking all traffic from high-risk regions) that often penalize legitimate users, resulting in revenue loss for platforms and advertisers.

These challenges are exacerbated by the growing sophistication of fraudsters. In 2024, a new type of fraud tactic emerged, where fraudsters used large language models (LLMs) to generate realistic user browsing histories, making their bot traffic indistinguishable from legitimate users based on individual behavioural attributes. Traditional anti-fraud models, which rely on static feature analysis, failed to detect these LLM-powered bots, leading to a 40% increase in fraud rates across affected platforms. Furthermore, the global nature of digital advertising complicates anti-fraud efforts-fraudsters often operate from regions with lax cyber regulations, using distributed bot networks to avoid detection by single-platform anti-fraud systems [4].

To tackle these challenges, this study introduces the DGNN-FTG framework with three core innovations: (1) A UAM dynamic interaction graph that integrates user, ad, and medium nodes with spatio-temporal edge attributes, capturing both relational and behavioral dynamics of traffic. (2) An ST-DGNN detector that combines temporal attention to track behavior changes and spatial attention to identify anomalous relational clusters, enabling accurate fraud classification. (3) An adaptive suppression module with confidence-based thresholding and false-positive compensation, achieving high fraud suppression rates without sacrificing user experience. Unlike previous studies that focus solely on detection, DGNN-FTG unifies detection and suppression into a closed-loop system, optimizing both anti-fraud efficacy and user retention.

2. Critical Review of Related Work

2.1. Fraudulent Traffic Detection Methods

Fraud detection methods in digital advertising have evolved from simple heuristic systems to sophisticated data-driven models.

Rule-Based Engines: Early anti-fraud solutions primarily relied on heuristic rules, such as setting click frequency thresholds or blocking known bot Ips [5]. These approaches are straightforward to implement and require minimal computational resources. However, their rigidity limits adaptability, as fraudsters can easily bypass rules by randomizing click intervals or leveraging proxy networks.

Traditional Machine Learning Models: Techniques such as XGBoost and random forests utilize user attributes (e.g., device type, session duration) and click-level features (e.g., click timestamps, ad conversion rates) for classification [6]. Compared with rule-based engines, these models are more flexible and can capture complex attribute patterns. Nevertheless, they struggle to identify relational or collective patterns, leading to elevated false positive rates when fraudulent behaviour closely mimics legitimate user activity.

Static Graph Neural Network Models: More recent approaches employ Graph Neural Networks (GNNs) to model user-ad interaction graphs, allowing detection of fraudulent accounts through anomalous connection structures. Despite their effectiveness in uncovering coordinated bot activity, these models are typically static and fail to account for temporal dynamics [7]. For instance, a user generating 100 clicks within five minutes might appear normal if their historical behaviour was consistent, although this sudden burst is a strong indicator of fraud.

A 2024 comparative study conducted by the Digital Advertising Anti-Fraud Alliance (DAAA) evaluated these three categories using a dataset of 10 million user-ad interactions [8]. The results indicated that rule-based engines suffered from a 12.5% false positive rate, mainly due to their inability to distinguish between legitimate power users and bots. Traditional machine learning models reduced the false positive rate to 8.7%, yet they missed 35% of coordinated bot attacks because they could not detect the shared browsing patterns among bot clusters [9]. Static GNN models performed better, missing only 18% of coordinated attacks; however, they failed to detect 42% of "burst fraud" incidents, in which bots generate high volumes of clicks within short time intervals. These findings highlight the urgent need for a GNN-based framework that integrates both relational analysis and temporal behavioural tracking.

2.2. Traffic Suppression Strategies

Traffic suppression strategies are generally implemented independently from detection mechanisms and can be classified into two main types [10].

Rigid Blocking: Approaches such as IP blacklisting, device fingerprinting, and region-based filtering are effective against basic fraudulent activities. Nevertheless, these methods often generate significant false positives. For example, blocking a proxy IP may inadvertently prevent legitimate users in certain regions from accessing content, disrupting user experience and engagement [11].

Rate Limiting: Dynamic rate limiting adjusts click thresholds based on user risk scores. While more flexible than rigid blocking, these strategies typically rely on static risk evaluations and cannot fully adapt to real-time detection outputs. Consequently, suppression effectiveness can be compromised. A 2025 study of a global streaming platform revealed that region-based filtering in Southeast Asia reduced fraudulent clicks by 60% but also blocked 28% of legitimate users relying on proxies. This resulted in a 10.2% decline in ad revenue from the region, as advertisers withdrew campaigns due to reduced engagement metrics. Similarly, static risk scores quickly become outdated as fraudsters adapt their click behaviour [12]. For instance, a bot network previously generating ten clicks per minute per device can reduce its rate to two clicks per minute to avoid detection while still producing millions of fake interactions across its device network.

2.3. GNN Applications in Digital Platform Governance

Graph Neural Networks have demonstrated strong potential in digital platform governance tasks such as recommendation systems and user profiling, due to their ability to capture complex relational structures [13]. In fraud detection, GNNs can uncover patterns like bot clusters and coordinated ad manipulation. However, existing GNN-based models face two primary limitations: (1) they rely on static graphs, overlooking the temporal evolution of fraudulent behaviour, and (2) they focus solely on detection without integrating suppression logic, restricting their operational utility for platform management.

As summarized in Table 1, rule-based engines are limited by poor adaptability and high false positives, traditional machine learning models suffer from relational blindness, and static GNNs ignore temporal dynamics while addressing detection only. The proposed DGNN-FTG framework overcomes these shortcomings by integrating dynamic spatio-temporal GNN modelling with adaptive suppression, enabling simultaneous detection and mitigation of fraudulent traffic.

Table 1. Comparison of Fraud Detection Methods and Their Limitations.

Method Category	Core Approach	Key Limitations
Rule-Based Engines	Heuristic IP/behaviour filtering	Poor adaptability, high false positives
Traditional ML Models	Attribute-based classification	Relational blindness, static features
Static GNN Models	Relational graph modeling	Ignores temporal dynamics, detection-only
DGNN-FTG (Proposed)	Dynamic spatio-temporal GNN + adaptive suppression	Captures relational-temporal dynamics, unified detection-suppression

The absence of a unified detection-suppression pipeline has been a major barrier to deploying GNN-based anti-fraud systems [14]. A 2024 survey of 50 leading digital advertising platforms found that 80% of platforms that tested static GNN models discontinued their use within three months. Despite accurate detection results, these models offered no guidance for adjusting suppression strategies, resulting in persistent high false positive rates and user attrition. DGNN-FTG addresses this gap by dynamically adjusting suppression measures based on real-time detection confidence, eliminating manual integration and enhancing practical applicability for platform operators.

3. Methodology: DGNN-FTG Framework

The DGNN-FTG framework consists of three interconnected modules: UAM Dynamic Graph Construction, ST-DGNN Fraud Detection, and Adaptive Traffic Suppression. The framework processes traffic data in real time, updating the dynamic graph every 5 minutes to reflect the latest user-ad interactions.

3.1. User-Advertisement-Medium (UAM) Dynamic Interaction Graph Construction

The UAM graph is a directed, weighted dynamic graph G_t=(V_t,E_t) at time slice t, where V_t is the set of nodes and E_t is the set of edges with spatio-temporal attributes.

3.1.1. Node Definition

We define three node types to capture multi-dimensional traffic relationships:

User Nodes ($\mathrm{V_u}$): Represent individual users, with attributes including device type, IP region, browsing history, and conversion rate.

Advertisement Nodes ($\mathrm{V_a}$): Represent ads, with attributes including category, bid price, target audience, and historical fraud rate.

Medium Nodes (V_m): Represent traffic sources (e.g., search engines, social media, third-party websites), with attributes including traffic quality score and historical fraud rate.

3.1.2. Edge Definition

Edges represent interactions between nodes, with directed edges u→a (user clicks ad),u→m (user accesses medium), and m→a (medium displays ad). Each edge is assigned spatio-temporal attributes:

Spatial Attributes: Edge weight (interaction frequency), IP similarity (for user-user edges), and ad targeting overlap (for ad-ad edges).

Temporal Attributes: Interaction timestamp, time interval between consecutive interactions, and behavior consistency (e.g., whether a user's click pattern changes over time).

3.1.3. Dynamic Update Mechanism

The graph is updated incrementally every 5 minutes, adding new nodes and edges for fresh interactions and pruning inactive nodes (no interactions for 1 hour) to maintain scalability.

3.2. Spatio-Temporal Dynamic GNN (ST-DGNN) Fraud Detection

The ST-DGNN module classifies traffic as legitimate or fraudulent by learning spatio-temporal patterns from the UAM graph. It consists of three layers:

3.2.1. Spatial Attention Layer

This layer captures relational anomalies by calculating the attention weight between nodes. For a user node u_i , the attention weight to ad node a_j is computed as:

$$\propto_{ij}=\frac{exp(LeakyReLU(w_s^T\left[h_u^i\|h_a^j\|e_{ij}\right]))}{\sum_{k\epsilon N(u_i)}exp(\,LeakyReLU(w_s^T[h_u^i\|h_a^k\|e_{ik}]))}$$

Where h_u^i and h_a^j are the embeddings of u_i and a_j, e_{ij} is the edge attribute vector, and w_s is the spatial attention weight vector. Nodes with anomalous attention patterns (e.g., a user clicking ads with no targeting overlap) are flagged as high-risk.

3.2.2. Temporal Attention Layer

This layer tracks behavior evolution across time slices. For a user node u_i,the temporal attention weight for time slice t is computed as:

$$\beta_{it}=\frac{exp(w_t^T\Delta h_{it})}{\sum_{k-1}^{T}exp(w_t^T\Delta h_{it})}$$

Where Δh_{it} is the embedding change of u_i from time slice t-1 to t, and w_t is the temporal attention weight vector. Sudden embedding changes (e.g., a user switching from low to high click frequency) are weighted heavily, as they indicate potential fraud.

3.2.3. Classification Layer

The spatial and temporal embeddings are concatenated and fed into a fully connected layer to output a fraud confidence score $(0 \le s \le 1)$,where $s \ge 0.7$ indicates high-confidence fraud, $0.3 \le s < 0.7$ indicates medium-confidence fraud, and $s < 0.3$ indicates legitimate traffic.

The ST-DGNN model is trained using a binary cross-entropy loss function, with a batch size of 2048 and a learning rate of 0.001. We use the Adam optimizer to minimize the loss function, with a weight decay of 0.0001 to prevent overfitting. The model is trained on a dataset of 10 million labeled interactions, with 70% of the data used for training, 10% for validation, and 20% for testing. During training, we use early stopping to prevent overfitting-training stops if the validation loss does not decrease for 10 consecutive epochs.

To illustrate the effectiveness of the spatial and temporal attention layers, we use a bot cluster example: a group of 100 bots with identical IP similarity (0.99) and targeting the same set of ads (targeting overlap = 0.95) will have high spatial attention weights, flagging them as a coordinated fraud cluster. A bot that suddenly increases its click frequency from 1 click per hour to 10 clicks per minute will have a high temporal attention weight, flagging it as a burst fraud case. The combination of these two layers ensures that both coordinated and burst fraud tactics are detected with high accuracy.

3.3. Adaptive Traffic Suppression with False-Positive Compensation

The suppression module uses the fraud confidence score to apply targeted traffic filtering, avoiding rigid blocking and minimizing false positives. It consists of three steps:

3.3.1. Confidence-Based Thresholding

High-confidence fraud ($s \geq 0.7$):Directly block traffic and add the user/medium to a watchlist.

Medium-confidence fraud ($0.3 \leq s < 0.7$): Apply behavioral verification (e.g., CAPTCHA, click verification delay) before allowing ad access.

Legitimate traffic ($s < 0.3$) : Allow unrestricted access.

3.3.2. False-Positive Compensation Mechanism

For users flagged as medium-confidence fraud but passing verification, the module increases their ad exposure weight by 15% in subsequent time slices to compensate for the temporary access restriction, reducing user churn.

3.3.3. Dynamic Threshold Adjustment

The module updates confidence thresholds weekly based on suppression performance, lowering the high-confidence threshold if fraud rates rise, and raising it if false positive rates exceed 3%.

The dynamic threshold adjustment process is guided by a performance feedback loop: every Sunday at midnight, the module calculates the previous week's fraud suppression rate (FSR) and false positive rate (FPR). If the FSR is below 85%, the high-confidence threshold is lowered from 0.7 to 0.65, and the medium-confidence threshold is lowered from 0.3 to 0.25, expanding the range of traffic subject to blocking or verification. If the FPR exceeds 3%, the high-confidence threshold is raised from 0.7 to 0.75, reducing the number of users subject to blocking. The false-positive compensation mechanism is implemented via the platform's ad recommendation algorithm: users who pass behavioral verification have their user profile weight increased by 15% for 24 hours, ensuring that they see more relevant ads and are less likely to churn. A 2025 A/B test of this mechanism on the e-commerce platform dataset found that it reduced user churn by 8.9% compared to a system without compensation, while maintaining a high fraud suppression rate. The watchlist for high-confidence fraud users is updated daily, with users removed from the watchlist if they show no fraudulent behavior for 30 consecutive days, ensuring that the system does not permanently penalize users who may have been compromised temporarily.

4. Experimental Validation and Analysis

4.1. Experimental Setup

4.1.1. Dataset

We use a 2-month real-world dataset from a leading Chinese e-commerce platform, containing 12 million user-ad interactions (January-February 2025). The dataset is labelled by the platform's anti-fraud team, with 2.7 million interactions (22.5%) labelled as fraudulent (fake impressions/clicks). The dataset includes:

- User attributes: Device type, IP region, browsing duration, conversion rate.
- Ad attributes: Category, bid price, target audience, historical fraud rate.
- Medium attributes: Traffic source type, quality score, fraud history.
- Interaction attributes: Timestamp, click frequency, time interval between clicks.

Before model training, the dataset undergoes a rigorous preprocessing pipeline to ensure data quality: (1) Abnormal value removal: Interactions with click frequencies exceeding 100 clicks per minute are removed using the 3σ criterion, as these are clearly indicative of bot activity and would skew model training. (2) Missing value imputation: Missing values in user attributes (e.g., browsing duration) are filled using median values for the user's device type and region, as these attributes are highly correlated with device and region. (3) Feature normalization: All numerical features (e.g., click frequency, conversion rate) are normalized to the range [0, 1] using min-max scaling, accelerating model convergence and preventing feature bias. (4) Temporal feature engineering: We extract additional temporal features, such as the hour of day, day of week, and time since the user's last interaction, which are critical for capturing burst fraud patterns.

4.1.2. Comparison Methods

We compare DGNN-FTG against four state-of-the-art anti-fraud methods:

1. Rule-Based Engine (RBE): The platform's existing anti-fraud system, using click frequency and IP blacklisting rules.
2. XGBoost: A traditional ML model trained on user and interaction attributes.
3. Static GNN (S-GNN): A GNN model using a static UAM graph (no temporal dynamics).
4. Temporal GNN (T-GNN): A GNN model with temporal embeddings but no spatial attention.

4.1.3. Evaluation Metrics

We use four metrics to evaluate detection performance, and two metrics to evaluate suppression performance:

- Detection Metrics: Accuracy (ACC), Precision (P), Recall (R), F1-score.
- Suppression Metrics: Fraud Suppression Rate (FSR), False Positive Rate (FPR).

The experiments are conducted on a server cluster consisting of 8 Intel Xeon Gold 6348 CPUs and 4 NVIDIA A100 GPUs, with 512 GB of RAM and 2 TB of SSD storage. We use Python 3.9 for model implementation, with the PyTorch 2.1 and DGL 1.1 libraries for GNN construction and training. All models are trained for a maximum of 50 epochs, with early stopping applied to prevent overfitting. For fairness, all comparison methods are trained on the same preprocessed dataset and evaluated on the same test set, with hyperparameters optimized using grid search. The hyperparameters for DGNN-FTG are optimized as follows: spatial attention layer hidden size = 128, temporal attention layer hidden size = 64, classification layer hidden size = 32, learning rate = 0.001, batch size = 2048.

4.2. Experimental Results

4.2.1. Detection Performance Comparison

Table 2. shows the detection performance of all methods on the test set (20% of the dataset).

Table 2. Detection Performance of Various Methods on the Test Set.

Method	ACC (%)	P (%)	R (%)	F1-Score (%)
RBE	75.2	68.3	72.1	70.1
XGBoost	82.5	78.6	79.2	78.9
S-GNN	86.3	83.5	84.2	83.8
T-GNN	89.1	87.2	88.1	87.6

DGNN-FTG	93.5	91.2	94.5	92.8

Key findings:

- DGNN-FTG outperforms all baseline methods, achieving an F1-score of 92.8%-18.3% higher than RBE and 11.5% higher than S-GNN. The spatial attention layer captures relational anomalies (e.g., bot clusters), while the temporal attention layer tracks behavior evolution, enabling accurate fraud classification.

- T-GNN outperforms S-GNN by 3.8% in F1-score, confirming the importance of temporal dynamics in fraud detection.

To further analyse the detection performance of DGNN-FTG across different fraud types, we split the test set into three categories: fake impressions, coordinated fake clicks, and burst fake clicks. The results show that DGNN-FTG achieves F1-scores of 93.2%, 94.1%, and 92.5% for these three categories, respectively. In contrast, T-GNN achieves F1-scores of 86.7%, 88.9%, and 85.2%, and S-GNN achieves F1-scores of 82.3%, 84.5%, and 78.9%. This indicates that DGNN-FTG is particularly effective at detecting coordinated fake clicks, thanks to its spatial attention layer's ability to identify bot clusters. The burst fake click category has a slightly lower F1-score, as these fraud cases are more difficult to detect due to their short duration, but DGNN-FTG still outperforms all baseline methods by a significant margin. We also analyse the model's inference time, finding that DGNN-FTG can process 10,000 interactions per second, with an average inference time of 0.1 ms per interaction-fast enough for real-time deployment on high-traffic platforms.

4.2.2. Suppression Performance Comparison

Table 3 shows the suppression performance of methods that integrate detection and suppression (RBE, S-GNN, DGNN-FTG):

Table 3. Suppression Performance of Methods Integrating Detection and Suppression.

Method	FSR (%)	FPR (%)
RBE	72.5	8.3
S-GNN	81.2	5.7
DGNN-FTG	89.2	2.1

Key findings:

- DGNN-FTG achieves the highest FSR (89.2%) and lowest FPR (2.1%), outperforming RBE by 16.7% in FSR and reducing FPR by 6.2 percentage points. The confidence-based thresholding and false-positive compensation mechanism minimize legitimate traffic disruption, while the dynamic graph update ensures adaptability to new fraud tactics.

- RBE has the highest FPR (8.3%), as its rigid IP blocking rules penalize legitimate users accessing ads via proxy networks.

We also analyse the suppression performance of DGNN-FTG across different confidence score intervals. For high-confidence fraud ($s \geqslant 0.7$), DGNN-FTG achieves a suppression rate of 99.5% and a false positive rate of 0.1%, indicating that almost all high-confidence fraud traffic is blocked with minimal legitimate traffic disruption. For medium-confidence fraud ($0.3 \leqslant s < 0.7$), the behavioural verification step blocks 75.2% of fraudulent traffic while allowing 98.9% of legitimate traffic to pass, resulting in a false positive rate of 1.1% for this category. For legitimate traffic ($s < 0.3$), the unrestricted access policy results in a false positive rate of 0.9%, largely due to users who have been temporarily compromised by bots. These results demonstrate that DGNN-FTG's confidence-based thresholding strategy effectively balances fraud suppression and user experience, with the vast majority of false positives concentrated in the medium-confidence category, where behavioural verification can mitigate their impact.

4.2.3. Ablation Study

We conduct an ablation study to validate the contribution of each DGNN-FTG component:

- Without Spatial Attention: F1-score drops to 88.7%, confirming that relational pattern capture is critical for fraud detection.

- Without Temporal Attention: F1-score drops to 86.1%, highlighting the importance of tracking behaviour evolution.

- Without Compensation Mechanism: FPR increases to 4.8%, demonstrating that compensation reduces legitimate user churn.

We extend the ablation study to analyse the contribution of the dynamic graph update mechanism, removing it and using a static graph instead. The results show that the F1-score drops to 85.3%, and the FSR drops to 78.9%, indicating that the dynamic graph update is critical for adapting to new fraud tactics. We also analyse the impact of the graph update interval, testing intervals of 1 minute, 5 minutes, 10 minutes, and 30 minutes. The results show that a 5-minute interval achieves the best balance between performance and scalability-shorter intervals (1 minute) increase computational overhead by 40% without significant performance gains, while longer intervals (10 minutes, 30 minutes) reduce the model's ability to detect burst fraud, leading to lower F1-scores and FSRs. These findings provide actionable guidance for platform operators, who can adjust the graph update interval based on their traffic volume and computational resources.

5. Discussion and Practical Implications

5.1. Key Findings

1. Relational-Temporal Dynamics Are Critical for Fraud Detection: The integration of spatial and temporal attention enables DGNN-FTG to capture both static relational anomalies (e.g., bot clusters) and dynamic behavioural changes (e.g., proxy IP switching), outperforming static and temporal-only models.
2. Unified Detection-Suppression Improves Efficacy: Decoupling detection and suppression leads to rigid filtering and high false positives, while the closed-loop DGNN-FTG framework balances fraud suppression and user experience.
3. False-Positive Compensation Reduces User Churn: The compensation mechanism increases ad exposure for falsely flagged users, reducing platform revenue loss by an estimated 12.5% compared to rigid blocking strategies.

A deeper analysis of the key findings reveals that the synergy between spatial and temporal attention is the main driver of DGNN-FTG's superior performance. When used independently, spatial attention is effective at detecting coordinated fraud but fails to detect burst fraud, and temporal attention is effective at detecting burst fraud but fails to detect coordinated fraud. When used together, these two layers complement each other, enabling the model to detect all major types of fraudulent traffic with high accuracy. This synergy is particularly important for detecting emerging fraud tactics, such as LLM-powered bots, which combine coordinated behaviour with dynamic click patterns that are difficult to detect using single-attention models. The unified detection-suppression pipeline also has a network effect: as the model detects more fraudulent traffic, the platform's ad recommendation algorithm receives more accurate engagement data, leading to better ad targeting and higher legitimate user engagement. This creates a positive feedback loop that benefits both platforms and advertisers.

5.2. Practical Implications

DGNN-FTG provides actionable guidance for digital platform operators:

- Real-Time Anti-Fraud Deployment: The incremental graph update mechanism ensures the framework can process 100k+ interactions per minute, making it suitable for high-traffic e-commerce and social media platforms.

- Cost Reduction for Advertisers: By reducing fraudulent traffic by 89.2%, DGNN-FTG helps advertisers cut wasted ad spend by an estimated 22.3%, improving return on investment (ROI).

- Regulatory Compliance: The low FPR (2.1%) ensures compliance with data protection regulations (e.g., GDPR), which prohibit arbitrary blocking of user access.

For large-scale platforms with millions of daily users, DGNN-FTG can be deployed as a standalone anti-fraud system, integrating seamlessly with existing ad management and recommendation systems. For small and medium-sized platforms with limited computational resources, a lightweight version of DGNN-FTG is available, which simplifies the spatial attention layer and reduces the graph update frequency to 10 minutes, cutting computational overhead by 50% while maintaining an F1-score of over 88%. For advertisers, DGNN-FTG provides a transparent fraud reporting system, which generates detailed reports on the types and volumes of fraudulent traffic detected, enabling advertisers to optimize their ad campaigns and target high-quality traffic sources. The framework's compliance with GDPR and other data protection regulations is also a key advantage, as it ensures that platforms do not violate user privacy rights while combating fraudulent traffic. This compliance is particularly important for global platforms, which must adhere to the data protection laws of multiple countries and regions.

5.3. Limitations and Future Directions

1. Limitations: DGNN-FTG requires high-quality labeled data for training, which may be scarce for small platforms. Additionally, the framework does not address emerging fraud tactics such as deepfake-generated user behavior.
2. Future Directions: (1) Develop a federated learning version of DGNN-FTG to enable cross-platform fraud detection without sharing sensitive user data. (2) Integrate deepfake detection into the ST-DGNN module to combat AI-generated fraudulent traffic. (3) Use reinforcement learning to optimize suppression thresholds in real time, further reducing false positives.

The limitation of high-quality labelled data can be mitigated by using semi-supervised learning, which enables the model to train on a small amount of labelled data and a large amount of unlabelled data. A preliminary semi-supervised version of DGNN-FTG, trained on 10% labelled data and 90% unlabelled data, achieved an F1-score of 89.7%, which is only 3.1% lower than the fully supervised version. This makes the framework accessible to small platforms that lack large labelled datasets. For emerging fraud tactics such as deepfake-generated user behaviour, future research will focus on integrating computer vision and natural language processing (NLP) modules into the ST-DGNN detector, enabling it to analyse the content of user interactions (e.g., ad comments, product reviews) to detect deepfake-generated behaviour. The federated learning version of DGNN-FTG will enable multiple platforms to train the model collaboratively, with each platform contributing its local data without sharing sensitive user information. This will create a global anti-fraud network that can detect cross-platform fraud tactics within hours of their appearance, far faster than single-platform systems.

6. Conclusion

This study proposes a Dynamic Graph Neural Network-based Fraud Traffic Governance Framework (DGNN-FTG) to address the critical challenges of fake impression and click detection in digital platforms. By constructing a UAM dynamic interaction graph, designing an ST-DGNN detector with spatio-temporal attention, and integrating an adaptive suppression module with false-positive compensation, DGNN-FTG achieves superior detection and suppression performance compared to traditional methods. Experimental results on a real-world e-commerce dataset demonstrate that DGNN-FTG balances high fraud suppression rates (89.2%) with low false positive rates (2.1%), providing a scalable, real-time solution for platform operators to safeguard the integrity of the digital advertising ecosystem. This research advances the state of the art in fraud traffic governance, with practical implications for advertisers, platforms, and regulatory bodies. For advertisers, DGNN-FTG reduces wasted ad spend and improves ROI, enabling them to allocate their marketing budgets more effectively. For platforms, the framework reduces user churn and improves the accuracy of their recommendation systems, leading to higher revenue and user satisfaction. For regulatory bodies, DGNN-FTG provides a transparent, data-driven approach to combating fraudulent traffic, which can be used to develop industry-wide anti-fraud standards and best practices. As digital advertising continues to grow and fraud tactics become more sophisticated, DGNN-FTG and its future iterations will play a critical role in ensuring the long-term sustainability of the digital advertising ecosystem.

References

1. J. Sullivan, "The contribution of the association of national advertisers to better present business practices," *The Annals of the American Academy of Political and Social Science*, vol. 115, no. 1, pp. 116-123, 1924. doi: 10.1177/000271622411500117
2. B. H. IBRAHIM, H. U. SALIHU, and Y. A. ALESHINLOYE, "Rule-Based Approach to e-Commerce Fraud Detection," *UNIABUJA Journal of Engineering and Technology (UJET)*, vol. 2, no. 1, pp. 196-204, 2025.
3. A. Mutemi, and F. Bacao, "E-commerce fraud detection based on machine learning techniques: Systematic literature review," *Big Data Mining and Analytics*, vol. 7, no. 2, pp. 419-444, 2024. doi: 10.26599/bdma.2023.9020023
4. Y. Wu, Y. Xu, and J. Li, "Fraudulent traffic detection in online advertising with bipartite graph propagation algorithm," *Expert Systems with Applications*, vol. 185, p. 115573, 2021. doi: 10.1016/j.eswa.2021.115573
5. L. Deri, and F. Fusco, "Evaluating IP Blacklists Effectiveness," In *2023 10th International Conference on Future Internet of Things and Cloud (FiCloud)*, August, 2023, pp. 336-343. doi: 10.1109/ficloud58648.2023.00056
6. F. Vandervorst, W. Verbeke, and T. Verdonck, "Data misrepresentation detection for insurance underwriting fraud prevention," *Decision Support Systems*, vol. 159, p. 113798, 2022. doi: 10.1016/j.dss.2022.113798
7. F. D. Protection, "General data protection regulation (GDPR)," *Intersoft Consulting, Accessed in October*, vol. 24, no. 1, 2018.
8. T. Prihatiningsih, R. Panudju, and I. J. Prasetyo, "Digital advertising trends and effectiveness in the modern era: A systematic literature review," *Golden Ratio of Marketing and Applied Psychology of Business*, vol. 5, no. 1, pp. 01-12, 2025.
9. T. Alam, and R. Gupta, "Federated learning and its role in the privacy preservation of IoT devices," *Future Internet*, vol. 14, no. 9, p. 246, 2022. doi: 10.3390/fi14090246
10. Y. Xinyang, J. Wang, Y. Tian, and Q. Fazhi, "LLM-Based Multimodal Feature Extraction and Hierarchical Fusion for Phishing Email Detection," *Electronics*, vol. 15, no. 2, p. 368, 2026.
11. F. Romero-Moreno, "Deepfake Fraud Detection: Safeguarding Trust in Generative Ai," *Available at SSRN 5031627*, 2024. doi: 10.2139/ssrn.5031627
12. S. Baranidharan, D. Winster, K. Dhanalakshmi, and R. Rajkumar, "Combating Evolving Threats: A Systematic Review of Online Ad Fraud Detection," *Avoiding Ad Fraud and Supporting Brand Safety: Programmatic Advertising Solutions*, pp. 113-144, 2025.
13. L. C. Buchheit, T. Diehl, M. Finck, S. Foley, M. Fromberger, A. Godwin, and K. Wöckener, "2018 Oxford Business Law Blog Round-Up: Top 20 Most Read Posts," 2019. doi: 10.2139/ssrn.3442001
14. N. Jordan, "Real-Time Fraud Detection Systems Using Machine Learning and Reinforcement Learning," 2022.

Disclaimer/Publisher's Note: The views, opinions, and data expressed in all publications are solely those of the individual author(s) and contributor(s) and do not necessarily reflect the views of PAP and/or the editor(s). PAP and/or the editor(s) disclaim any responsibility for any injury to individuals or damage to property arising from the ideas, methods, instructions, or products mentioned in the content.