IJISEM
INTERNATIONAL JOURNAL OF INNOVATIONS IN SCIENCE ENGINEERING AND MANAGEMENT

International Journal of Innovations In Science Engineering And Management

Check for updates

OPEN ACCESS

Volume: 5

Issue: 1

Month: March

Year: 2026

ISSN: 2583-7117

Published: 13.03.2026

Citation:

Sonali Sharma, Sadu Atchuta Rao "Data-Driven Analysis of Customer Return Behavior and Fraudulent Patterns in Indian E-Commerce" International Journal of Innovations in Science Engineering and Management, vol. 5, no. 1, 2026, pp. 274-280.

DOI:

10.69968/ijisem.2026v5i1274-280

This work is licensed under a Creative Commons Attribution-Share Alike 4.0 International License

Data-Driven Analysis of Customer Return Behavior and Fraudulent Patterns in Indian E-Commerce

Sonali Sharma[1], Sadu Atchuta Rao[2]

[1,2]Data Science, MVGR College of Engineering, Vizianagaram, India

Abstract

The rapid growth of the Indian e-commerce sector has brought a parallel rise in "wardrobing" and return abuse, where customers purchase items for temporary use and return them for a full refund. Traditional fraud detection systems, particularly tabular models like XGBoost, frequently fail to catch this behavior because they treat every transaction as an isolated event, completely missing the structural "guilt-by-association" patterns inherent in organized fraud. To address this gap, this paper presents a Graph Neural Network (GNN) framework integrated with geospatial hotspot analysis to detect behavioral fraud across a unified Indian market dataset. We benchmarked three distinct architectures to handle this complexity: XGBoost, GraphSAGE, and Graph Attention Networks (GAT). Our comparative analysis revealed that while the XGBoost baseline achieved high overall accuracy, it suffered from a severe bias toward the majority class, missing 61% of actual fraud cases. GraphSAGE emerged as the champion model; by utilizing inductive learning and neighborhood aggregation, it successfully mapped user-item relationships and improved the fraud detection recall rate by 21% over the baseline. Conversely, the more complex GAT architecture struggled with the dataset's extreme class imbalance and overfit the noise. Ultimately, this research demonstrates that transitioning from flat data to Geometric Deep Learning, combined with regional hotspot tracking, is essential for securing modern retail platforms against subtle behavioral abuse.

Keywords: Commerce Return Fraud, Wardrobing, Geometric Deep Learning, GraphSAGE, XGBoost, Geospatial Hotspot Analysis

INTRODUCTION

The rapid expansion of the Indian e-commerce sector has revolutionized retail but concurrently introduced a silent yet substantial financial drain: return fraud. While conventional security systems are adept at detecting "hard" fraud, such as stolen credit cards, they consistently fail to identify "soft" fraud, primarily "wardrobing" or return abuse. This behavior, where a customer purchases an item for temporary use and returns it for a full refund, creates significant logistical challenges because the returned inventory frequently cannot be resold as new.

The fundamental challenge in detecting return abuse is that these transactions appear perfectly legitimate in isolation. Traditional machine learning models, such as XGBoost, represent the industry standard for tabular data but treat every transaction as an independent row or "island," fundamentally failing to capture the historical context or network structure of user behavior. Furthermore, real-world e-commerce data suffers from severe class imbalance, where actual fraudulent activity constitutes a minuscule fraction (less than 1%) of overall traffic, causing the subtle signals of organized fraud to be drowned out by the noise of legitimate shoppers.

To address these specific blind spots, recent advancements have shifted towards Geometric Deep Learning, specifically Graph Neural Networks (GNNs). Unlike tabular models, GNNs explicitly map the complex web of relationships between users and products. This structural understanding allows systems to identify "guilt-by-association" patterns, which are crucial for uncovering coordinated fraud networks that flat data models miss.

Motivated by these limitations, this paper proposes a comprehensive, data-driven framework for e-commerce return fraud detection. By engineering specific behavioral signals, such as a "Wardrobing Indicator" and a "User Global Return Rate," alongside synthesized geospatial mapping, the proposed system is designed to identify regional fraud hotspots and abusive patterns. Furthermore, this study rigorously benchmarks a traditional tabular baseline (XGBoost) against advanced graph architectures specifically GraphSAGE and Graph Attention Networks (GAT) to determine the most effective and scalable method for applying "network intelligence" to modern e-commerce platforms

LITERATURE SURVEY

The evolution of fraud detection in e-commerce has historically mirrored the increasing sophistication of fraudulent behaviors. This section reviews the transition from traditional tabular machine learning to advanced Geometric Deep Learning, specifically focusing on the financial impact of return fraud and the application of Graph Neural Networks (GNNs).

The Financial Impact of Wardrobing and Return Fraud

Return fraud, particularly "wardrobing" (where a customer purchases an item, uses it temporarily, and returns it for a full refund), has become a critical vulnerability for digital retailers. Ekata (2024) reported that return fraud cost U.S. merchants up to $101 billion in 2023, accounting for nearly 13.7% of all returns [15]. A recent study on Amazon's flexible return policies by Zengler et al. (2025) highlighted that while lenient return windows drive customer loyalty, they simultaneously create massive attack vectors for policy abuse [16]. To mitigate this, industry leaders are increasingly relying on behavioral analytics. Mangopay (2024) noted that traditional rule-based internal policies (such as blanket return bans) severely damage the user experience, whereas adaptive machine learning can prevent up to 60% of return fraud in real-time without alienating legitimate customers [14].

Limitations of Traditional Machine Learning (XGBoost and Isolation Forests)

Historically, supervised machine learning models like Random Forests and XGBoost have been the industry standard for anomaly detection. Tax et al. (2021) and Reddy et al. (2024) extensively mapped out the success of these models in detecting "hard" anomalies, such as stolen credit card usage and address manipulation [3], [1]. However, as Alexander et al. (2023) demonstrated, tabular models treat every transaction as an isolated event [2]. They fail to capture the multi-node, coordinated nature of modern fraud rings. Furthermore, when dealing with geospatial or temporal data, standard decision trees often experience data leakage or fail to properly index the chronological sequence of fraudulent behavior [8].

The Shift to Graph Neural Networks (GNNs)

To overcome the limitations of isolated transaction analysis, researchers have pivoted to modeling e-commerce environments as heterogeneous graphs. Mendonca et al. (2025) showcased that by representing users, merchants, and transactions as interconnected nodes, GNNs can effectively map "guilt-by-association" patterns [4]. This structural capability allows systems to uncover hidden fraud rings and coordinated review manipulations. Further validating this, Wang et al. (2024) and Zhang et al. (2024) implemented multi-layer Graph Convolutional Networks (GCNs) on real-world e-commerce datasets, proving that capturing structural transaction dependencies drastically lowers false-positive rates compared to traditional logistic regression [6], [5].

Scalability and Inductive Learning via GraphSAGE

A primary challenge with deploying standard GNNs in active e-commerce platforms is their transductive nature—they require complete retraining whenever a new user or transaction enters the network. To solve this, the GraphSAGE (Graph Sample and Aggregate) architecture was introduced. Chen et al. (2025) applied GraphSAGE to dynamic social networks for financial fraud detection, proving its superiority in handling continuous streams of new data [10]. Because GraphSAGE learns an inductive aggregation function rather than fixed node embeddings, it can evaluate unseen transactions in real-time [9]. Recent advancements, such as the HMOA-GNN framework proposed by Li et al. (2025), have further optimized GraphSAGE using adversarial training to prevent the model from over-smoothing in deep layers, making it highly effective for credit card and return fraud [7].

Addressing Imbalance with Graph Attention Networks (GAT)

Real-world e-commerce data suffers from extreme class imbalance, where fraudulent returns represent less than 1% of total traffic. Cherif et al. (2024) noted that in standard graph aggregations, the subtle signals of a fraudulent node are easily drowned out by the overwhelming noise of legitimate neighboring transactions [13]. To combat this, Graph Attention Networks (GAT) incorporate an attention mechanism that assigns dynamic weights to different

neighbors. Wang et al. (2025) implemented a GAT based on feature-difference weight assignments for telecom fraud, successfully isolating anomalous nodes from highly similar legitimate users [11]. Similarly, Mienye and Sun (2025) applied GATs to credit card transaction graphs, demonstrating that relational attention mechanisms significantly outperform tabular baseline models like Random Forest when predicting minority-class fraud [12].

METHODOLOGY

To effectively detect organized return abuse, the proposed system diverges from standard tabular machine learning by treating e-commerce transactions as interconnected networks.

System Architecture

The proposed fraud detection framework is designed as a multi-layered, end-to-end pipeline capable of processing disparate e-commerce data streams. The architecture consists of four primary modules:

1. **Data Ingestion Layer:** Aggregates and standardizes raw transaction records from multiple e-commerce platforms (Amazon, Flipkart, Meesho, Ajio, and Online Retail II) into a unified schema.
2. **Graph Construction Layer**: Converts the flat tabular data into a bipartite topological structure, mapping users and products as distinct nodes connected by transaction edges.
3. **Representation Learning Layer:** Utilizes Graph Neural Networks (specifically GraphSAGE) to perform neighborhood sampling and feature aggregation, generating low-dimensional embeddings for each user node.
4. **Classification & Analytics Layer:** Passes the generated embeddings through a dense classification network to output a final fraud probability score. Simultaneously, it maps the geographic coordinates of high-risk nodes to generate a spatial hotspot dashboard.

Data Unification and Bipartite Graph Construction

Standard models analyze datasets row-by-row, which inherently obscures coordinated multi-actor fraud rings [2]. To capture structural "guilt-by-association" relationships, the unified tabular data was converted into a bipartite graph [4]. The graph architecture defined two distinct node sets: Users (U) and Items (I). Edges (E) were generated to represent purchase interactions between these nodes. To stabilize the neural network's learning process and prevent high-magnitude numerical values from dominating the gradient updates, continuous attributes—such as Total Price and Product Return Rate—were standardized using Z-Score Normalization before being assigned as node features.

Behavioral Feature Engineering

Because explicit "fraud" labels are rarely provided in raw commercial datasets, targeted behavioral signals were engineered to define malicious intent. First, a specific "Wardrobing Indicator" was developed to flag transactions where a single user purchased the identical product in multiple sizes simultaneously, a strong precursor to return abuse [15]. Second, a "User Global Return Rate" metric was calculated to quantify the historical percentage of items a user returned over their entire lifecycle. Utilizing these signals, a synthetic ground-truth target variable was formulated: transactions were officially classified as fraudulent (Label 1) if the associated user possessed a global return rate exceeding 70% or explicitly triggered the wardrobing logic.

Multi-Architecture Implementation

To determine the optimal mathematical approach for handling behavioral complexity, the study benchmarked three distinct architectures:

1. ***Extreme Gradient Boosting (XGBoost):*** Deployed as the robust tabular baseline to establish the absolute limit of what non-structural data processing could achieve [1].
2. ***GraphSAGE:*** Implemented utilizing a mean aggregation function to enable inductive learning [10]. Unlike transductive methods that require full retraining for new data, GraphSAGE generates localized embeddings dynamically, making it scalable for real-time platforms where new users constantly join [9].
3. ***Graph Attention Networks (GAT):*** Constructed to test whether assigning dynamic attention weights to structural edges could effectively teach the model to ignore safe purchasing "noise" and focus strictly on high-risk relationships in highly imbalanced environments [13].

Algorithm

The core processing logic for the champion model (GraphSAGE) is detailed below. The algorithm demonstrates how the system processes a new, unseen user by sampling their transaction history and aggregating risk features to compute a final fraud probability.

Algorithm 1: Inductive Return Fraud Detection via GraphSAGE

Input: Graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ are User/Item nodes and $\mathcal{E}$ are transactions.

- Input features $x_v, \forall v \in \mathcal{V}$
- Depth K (number of search layers)
- Neighborhood sampling function $\mathcal{N}(v)$
- Weight matrices W^k and non-linear activation σ

Output: Vector representations z_v and Fraud Probability $P(y_u = 1)$

1. Initialize: $h_v^0 \leftarrow x_v, \forall v \in \mathcal{V}$
2. For $k = 1$ to K do:
3. For each node $v \in \mathcal{V}$ do:
4. Sample a fixed-size neighborhood: $\mathcal{N}_{sample}(v) \leftarrow \text{RandomSample}\big(\mathcal{N}(v)\big)$
5. Aggregate neighbor features:
$$h_{\mathcal{N}(v)}^k \leftarrow \text{MEAN}\big(\{h_u^{k-1}, \forall u \in \mathcal{N}_{sample}(v)\}\big)$$
6. Update node state:
$$h_v^k \leftarrow \sigma\left(W^k \cdot \text{CONCAT}\big(h_v^{k-1}, h_{\mathcal{N}(v)}^k\big)\right)$$
7. Normalize: $h_v^k \leftarrow h_v^k / |h_v^k|_2$
8. End For
9. End For
10. Extract Embeddings: $z_v \leftarrow h_v^K, \forall v \in \mathcal{V}$
11. Classification: For Target User u: $P(y_u = 1) \leftarrow \text{Sigmoid}\big(W_{clf} \cdot z_u + b\big)$
12. Return: $P(y_u = 1)$

ENVIRONMENTAL SETUP

The proposed e-commerce return fraud detection framework was developed and evaluated using a robust Python-based ecosystem, specifically Python version 3.8 or above. The advanced Geometric Deep Learning architectures, including GraphSAGE and Graph Attention Networks (GAT), were implemented utilizing the PyTorch framework (v1.12+) to efficiently handle dynamic computational graphs and complex message-passing logic. The tabular baseline model was constructed using the xgboost library, alongside scikit-learn for unbiased data splitting and performance metric evaluation (such as AUC-ROC). For large-scale data manipulation, schema standardization, and geospatial feature engineering, Pandas and NumPy were utilized extensively, while Matplotlib and Seaborn were employed for exploratory data analysis and spatial hotspot mapping.

Due to the computationally intensive nature of processing a bipartite graph with nearly 800,000 interactions, the hardware environment requires a multi-core processor (such as an Intel Core i7 or AMD Ryzen 7) and a minimum of 16 GB of system RAM to parallelize graph construction. Furthermore, training the GAT architecture demands significant memory to compute and store attention matrices across thousands of nodes; therefore, a CUDA-enabled NVIDIA GPU with at least 6 GB to 8 GB of VRAM is highly recommended to support large batch sizes (e.g., 1024) without encountering out-of-memory errors. The system is cross-platform compatible with both Windows 10/11 and Linux (Ubuntu 20.04+) operating systems. It requires approximately 20 GB of solid-state drive (SSD) storage to house the merged e-commerce datasets, synthesized geospatial records, and saved .pt model checkpoints.

EVALUATION METRICS USED

Since our proposed system deals with real-world e-commerce transaction data, it faces an extreme class imbalance where actual fraudulent behavior (Label 1) constitutes only approximately 5.23% of the dataset. In such environments, traditional metrics like overall Accuracy are highly deceptive, as a model could achieve near 95% accuracy simply by guessing that every transaction is "Safe". Therefore, we have used task-specific metrics that accurately measure the model's ability to distinguish between classes and its effectiveness at capturing the minority fraud class without being overwhelmed by the noise of legitimate shoppers.

Area Under the Receiver Operating Characteristic Curve (AUC-ROC): Measures the model's ability to distinguish between the fraudulent and legitimate classes across all possible classification thresholds. It represents the probability that the model ranks a randomly chosen positive instance higher than a randomly chosen negative one.

Recall (Sensitivity): Measures the proportion of actual fraudulent transactions correctly identified by the system. This is our most critical operational metric, as failing to detect an abusive return (False Negative) result in direct financial loss.

$$Recall = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

Precision: Measures the accuracy of the model's positive fraud predictions, effectively tracking the rate of false

alarms (False Positives) which could inconvenience legitimate customers.

$$Precision = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

F1-Score: The harmonic mean of Precision and Recall. It provides a single, balanced performance score specifically for the minority (fraud) class, ensuring that the model does not optimize for one metric at the extreme expense of the other.

$$F1\text{-}Score = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

IMPLEMENTATION LOGIC

The implementation of the proposed geometric deep learning framework is carried out through a modular data processing and graph learning pipeline. The following snippet represents the central logic responsible for data integration, structural graph construction, and inductive risk classification

```
def detect_return_fraud(transaction_data_paths):
    # Phase 1: Data Integration & Feature Engineering
    df = merge_ecommerce_datasets(transaction_data_paths)
    # Calculate User Global Return Rate & Wardrobing Indicator
    df = engineer_behavioral_features(df, return_threshold=0.70)
    # Phase 2: Graph Construction
    # Map transactions to User (U) and Item (I) nodes
    node_features, edge_index, labels = build_bipartite_graph(df)

    # Phase 3: Model Initialization (GraphSAGE Champion Model)
    model = GraphSAGE_Model(in_channels=node_features.shape[1],
                            hidden_channels=64,
                            out_channels=2)
    optimizer = torch.optim.Adam(model.parameters(), lr=0.005)
    # Phase 4: Inductive Training
    model.train()
    for epoch in range(10):  # Rapid convergence observed by Epoch 10
        optimizer.zero_grad()
        out = model(node_features, edge_index)
        loss = compute_loss(out, labels, class_weights='balanced')
        loss.backward()
        optimizer.step()
    # Phase 5: Risk Classification & Hotspot Mapping
    model.eval()
    with torch.no_grad():
        risk_probabilities = torch.softmax(model(node_features, edge_index), dim=1)
        predictions = (risk_probabilities[:, 1] > 0.5).int()
    hotspot_report = generate_geospatial_dashboard(df, predictions)
    return predictions, hotspot_report
```

The above code represents the core execution logic of the system. The ingested multi-platform transaction data is first merged and processed to engineer behavioral markers, specifically the "User Global Return Rate" and the "Wardrobing Indicator". The flat tabular data is then converted into a bipartite graph using PyTorch Geometric, enabling the GraphSAGE model to aggregate neighborhood features via inductive message-passing. Based on the structural graph embeddings, a final risk probability is assigned to detect fraudulent behavior, which is subsequently mapped to geographic nodes to highlight regional fraud hotspots.

RESULTS & DISCUSSIONS

We experimented with the unified Indian E-Commerce dataset across three distinct architectures to evaluate their

ability to detect subtle "wardrobing" behaviors. The observed values for key classification metrics—including ROC AUC, Precision, Recall, and F1-Score—are presented below in Table 1.

Table 1: Performance Benchmarking of Proposed Architectures

Metric	XGBoost (Baseline)	GraphSAGE (Champion)	GAT (Experimental)
ROC AUC Score	0.9763	0.9761	0.8638
Precision (Fraud)	0.65	0.64	0.47
Recall (Fraud)	0.39	0.60	0.26
F1-Score (Fraud)	0.49	0.62	0.34
True Positives	3,293	5,043	1,864
False Negatives	5,169	3,419	6,598
Model Type	Tabular/Tree	Graph/Inductive	Graph/Attention

The above table indicates that while the XGBoost baseline achieved a high ROC AUC score (0.9763), it suffered from a severe bias toward the majority "safe" class. This is reflected in its low Recall (0.39), meaning it missed 61% of the actual fraud cases (5,169 False Negatives). Conversely, GraphSAGE significantly improved the detection rate, achieving a Recall of 0.60 and identifying 5,043 True Positives. By simply aggregating neighborhood information, it caught nearly 2,000 more fraud cases than the baseline. The Graph Attention Network (GAT) struggled the most, recording the lowest F1-Score (0.34) and missing 6,598 fraud cases.

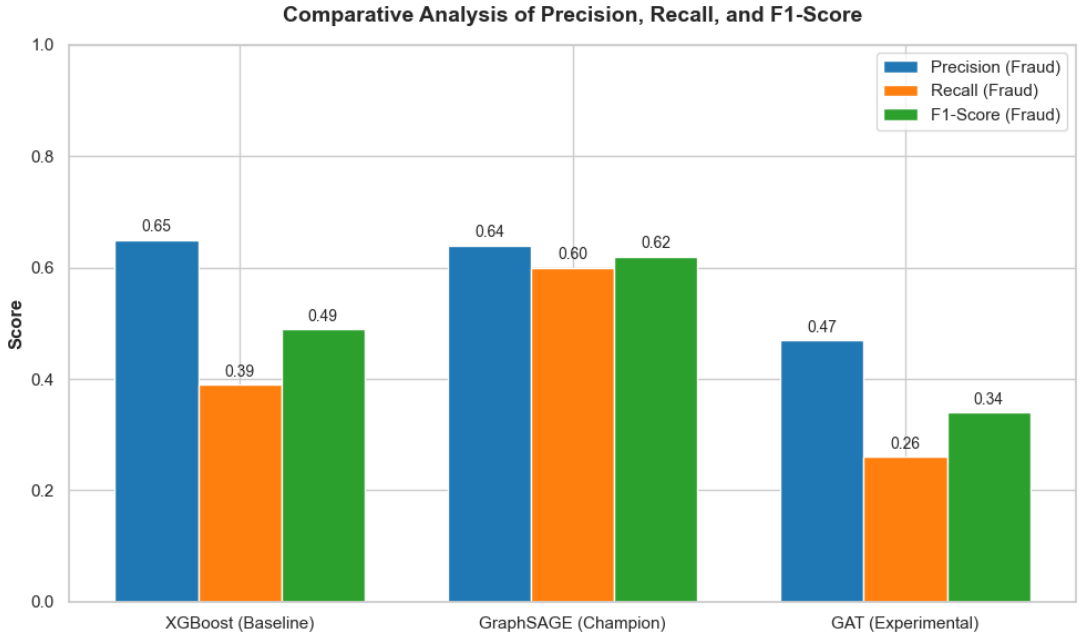


Figure 1: Bar Chart Comparative Analysis of Precision, Recall, and F1-Score Across Models

The comparative analysis illustrates the critical trade-offs between model complexity and structural awareness in highly imbalanced e-commerce environments. Through this study, it is observed that the tabular baseline (XGBoost) fails to capture the "guilt-by-association" networks inherent in organized return abuse. Furthermore, the "complexity trap" of GAT reveals that highly sensitive attention mechanisms tend to overfit to noise when fraudulent signals are extremely sparse. Overall, the GraphSAGE framework exhibits the most robust performance, proving that inductive neighborhood aggregation provides the optimal balance of accuracy, scalability, and structural fraud detection for modern e-commerce platforms.

CONCLUSION AND FUTURE SCOPE

Conclusion

This study successfully developed a robust data-driven foundation for analyzing "wardrobing" and return abuse within the Indian e-commerce sector. By unifying disparate datasets and engineering specific behavioral markers—such as the Wardrobing Indicator and User Global Return Rate—the research quantified the specific footprint of organized

fraud. The Exploratory Data Analysis proved that return fraud is highly concentrated among a small percentile of users (validating the Pareto principle) and clusters significantly in specific geographic hotspots. Furthermore, the extreme class imbalance identified (with fraud representing only ~5.23% of data) dictates that standard rule-based systems are insufficient for securing modern retail platforms.

Future Scope

The insights generated from this comprehensive EDA phase provide the exact parameters required for the next phase of this research: predictive modeling.

1. **Baseline Modeling:** The immediate next step involves implementing an Extreme Gradient Boosting (XGBoost) classifier to establish a high-performance tabular baseline.
2. **Geometric Deep Learning:** Because the EDA confirmed that fraud is a structural, network-based phenomenon, future work will transition to modeling the data as a bipartite graph. We intend to implement and evaluate GraphSAGE (Graph Sample and Aggregate) to leverage inductive neighborhood learning. This will allow the system to detect complex "guilt-by-association" relationships and multi-node fraud rings that traditional tabular models overlook.
3. **Imbalance Handling:** Finally, to counteract the 5% minority class constraint discovered during the EDA, future neural network training will explore attention mechanisms (such as Graph Attention Networks) and synthetic over-sampling techniques (SMOTE) to ensure high recall rates for fraudulent transactions.

REFERENCES

[1]. Reddy, S. R. B., Kanagala, P., Ravichandran, P., et al. "Effective fraud detection in e-commerce: Leveraging machine learning and big data analytics." Measurement: Sensors, vol. 33, 101138, 2024.

[2]. Alexander, I., Lai, C., & Yang, H. C. "Deep Learning Based Behavior Anomaly Detection within the Context of Electronic Commerce." 2023 IEEE International Conference on Intelligence and Security Informatics (ISI), pp. 1-6, 2023.

[3]. Tax, N., de Vries, K. J., de Jong, M., et al. "Machine learning for fraud detection in e-Commerce: A research agenda." Deployable Machine Learning for Security Defense, Springer, pp. 30-54, 2021.

[4]. Mendonça, R., Salazar, A., & Martinez, E. "Graph-Based Deep Learning for E-Commerce Fraud Detection." ResearchGate, March 2025.

[5]. Zhang, G., Li, Z., Huang, J., et al. "EFraudCom: an e-commerce fraud detection system via competitive graph neural networks." Macquarie University Research Outputs, 2024.

[6]. Wang, X., & Zhang, L. "Graph Neural Networks for Fraud Detection in E-commerce Transactions." 2024 International Conference on Computing, Sciences and Communications (ICCSC), IEEE Xplore, 2024.

[7]. Li, Y., et al. "HMOA-GNN: adaptive adversarial GraphSAGE with hierarchical hybrid sampling and metric-optimized graph construction for credit card fraud detection." PMC, Dec. 2025.

[8]. NVIDIA. "Supercharging Fraud Detection in Financial Services with Graph Neural Networks." NVIDIA Technical Blog, 2024.

[9]. Jabri, J. "Fraud detection with GNN." Kaggle, Jan. 2023.

[10]. Chen, Z., et al. "Application of GraphSAGE in Financial Fraud Detection within Dynamic Social Networks." IEEE Xplore, July 2025.

[11]. Wang, Y., et al. "GDFGAT: Graph attention network based on feature difference weight assignment for telecom fraud detection." PLOS One, 2025.

[12]. Mienye, I. D., & Sun, Y. "Graph attention networks for credit card fraud detection: A relational learning approach." World Journal of Advanced Research and Reviews, 26(03), pp. 2580-2585, 2025.

[13]. Cherif, et al. "Detecting Imbalanced Credit Card Fraud via Hybrid Graph Attention and Variational Autoencoder Ensembles." MDPI, 2024.

[14]. Mangopay. "How to prevent return fraud with machine learning models." Mangopay Blog, May 2024.

[15]. Ekata. "Return fraud: The $100 billion problem facing retailers." Ekata, Feb. 2024.

[16]. Zengler, T., et al. "The Role of AI in Preventing Return Fraud: A Study of Amazon's Flexible Return Policy and Consumer Behavior." ResearchGate, Sept. 2024