Annal. App. Sci, 2024
Vol 5(1) 2024

Dynamic Graph Neural Networks for Multi-Level Financial Fraud Detection: A Temporal-Structural Approach

Toan Khang Trinh[1,*] **, Zhuxuanzi Wang**[2]

[1] Computer Science, California State University Long Beach, CA, USA
[2] Information Systems, Cornell Tech, NY, USA

*Corresponding author E-mail: kintywanggg807@gmail.com

Abstract

Financial fraud detection presents significant challenges due to the complex, dynamic, and multi-level nature of fraudulent activities in modern financial systems. This paper proposes a Dynamic Graph Neural Network (DGNN) framework that captures both temporal dynamics and structural patterns across transaction, account, and community levels for comprehensive fraud detection. The architecture integrates a multi-level financial network construction method with a temporal-structural feature extraction module and a hierarchical detection framework. The temporal-structural approach employs graph attention networks for capturing spatial relationships between financial entities while utilizing temporal convolution networks to model evolving patterns. Bidirectional message passing enables information flow between different network levels, allowing the detection of sophisticated fraud schemes that operate across multiple organizational scales. Extensive experiments on three real-world financial datasets (CCFraud, MPFraud, and BankNet) demonstrate that our approach consistently outperforms state-of-the-art methods, achieving average improvements of 1.4%, 5.7%, and 5.5% in AUC-ROC, AUC-PR, and F1-score respectively. Ablation studies confirm the significance of each component in the architecture, with the combination of temporal and structural features providing substantial performance gains. The model shows particular strength in detecting complex fraud patterns involving multiple accounts and extending over longer time periods, validating the effectiveness of our multi-level approach for financial fraud detection in dynamic environments.

***Keywords*:** Graph Neural Networks, Financial Fraud Detection, Temporal-Structural Analysis, Multi-Level Architecture

Introduction

Research Background and Motivation

Financial fraud has emerged as a critical concern for global financial institutions, resulting in substantial economic losses estimated at $42 billion annually. Traditional fraud detection systems rely on rule-based approaches and conventional machine learning techniques, which often fail to capture the complex interconnections between entities in financial networks[1]. Graph-based representations provide a natural framework for modeling financial transactions, where nodes represent entities such as accounts, customers, or merchants, and edges represent relationships or transactions between them. The inherent relational structure of financial data makes Graph Neural Networks (GNNs) particularly suitable for fraud detection tasks.

Recent advances in deep learning have enabled significant progress in graph representation learning. GNNs have demonstrated exceptional capabilities in learning node embeddings that encapsulate both feature information and graph topology[2]. While static GNNs have shown promising results in various domains, financial fraud exhibits complex temporal-structural patterns that evolve over time. Fraudsters continuously adapt their techniques to evade detection systems, creating sophisticated multi-level schemes that operate across different scales of financial networks[3]. This dynamic nature necessitates advanced models capable of capturing temporal dependencies and structural patterns simultaneously.

Challenges in Financial Fraud Detection

Detecting financial fraud presents several unique challenges that limit the effectiveness of conventional approaches. Class imbalance constitutes a fundamental issue, as fraudulent transactions typically represent less than 0.1% of all transactions, making model training particularly difficult. The temporal evolution of fraud patterns requires models capable of adapting to emerging techniques while maintaining high detection accuracy for known patterns.

Multi-level fraud operations add another layer of complexity, as fraudsters coordinate activities across transaction, account, and network levels. Individual transactions might appear legitimate when examined in isolation, but reveal suspicious patterns when analyzed within their temporal and structural context. Graph-based anomaly detection methods must address these multi-level dynamics while processing massive volumes of real-time transactions.

The high-dimensional nature of financial data presents computational challenges for traditional graph algorithms, which often scale poorly with network size. Financial institutions must balance detection accuracy with computational efficiency, maintaining real-time performance while processing millions of transactions daily[4]. Privacy concerns and regulatory requirements further constrain the development and deployment of fraud detection systems.

Research Contributions

This paper introduces a dynamic graph neural network framework for multi-level financial fraud detection that addresses the aforementioned challenges through several innovations. The proposed architecture integrates temporal graph convolution networks with structural attention mechanisms, enabling effective learning of time-evolving node representations[5]. The multi-level detection framework operates simultaneously across transaction, account, and community levels, capturing fraud patterns that manifest at different granularities.

The temporal-structural approach incorporates adaptive time windows that adjust based on transaction volumes and network dynamics, optimizing the model's sensitivity to both rapid and gradual pattern changes. A hierarchical message-passing mechanism propagates information across levels while preserving both local and global graph properties. The fraud detection system employs a novel loss function designed specifically for imbalanced classification, improving performance on minority fraud cases without sacrificing overall accuracy.

Comprehensive experiments on real-world financial transaction datasets demonstrate significant performance improvements over state-of-the-art methods, particularly for complex fraud schemes involving multiple entities and extended time periods[6][7].

Related Work

Traditional Financial Fraud Detection Methods

Traditional approaches to financial fraud detection predominantly rely on rule-based systems, statistical methods, and conventional machine learning techniques. Rule-based systems implement predefined heuristics established by domain experts to flag suspicious activities, using thresholds for transaction amounts, frequency, and unusual patterns[8]. While these systems offer interpretability, they lack adaptability to new fraud patterns and require constant manual updates. Statistical methods such as logistic regression, decision trees, and random forests have been widely applied for fraud detection. These methods typically extract statistical features from transaction data, including transaction amount, time, location, and merchant category. Supervised learning approaches require labeled datasets, which are often limited in the financial fraud domain due to data privacy constraints and the rarity of confirmed fraud cases[9]. Unsupervised methods like clustering and outlier detection identify anomalous patterns without labeled data, though they often produce high false positive rates. Recent advancements in ensemble methods combine multiple models to improve detection accuracy, with gradient boosting machines demonstrating particularly strong performance on tabular financial data[10]. Deep learning models, including autoencoders and deep belief networks, have been employed to learn complex feature representations from high-dimensional financial data[11]. These approaches, while powerful, typically treat transactions as independent events, failing to capture the relational aspects of financial activities.

Graph Neural Networks in Anomaly Detection

Graph Neural Networks have revolutionized anomaly detection by explicitly modeling the relational structure of data. In graph-based anomaly detection, the goal is to identify nodes, edges, or subgraphs that deviate significantly from expected patterns. GNNs learn node embeddings through message-passing mechanisms, where each node aggregates information from its neighbors to update its representation. These embeddings capture both node attributes and topological information, enabling more comprehensive anomaly detection compared to traditional approaches[12]. Graph Convolutional Networks (GCNs) generalize convolutional operations to non-Euclidean domains by performing spectral convolutions on graphs[13]. GraphSAGE extends this approach through neighborhood sampling and aggregation strategies, improving scalability for large networks. Graph Attention Networks (GATs) incorporate attention mechanisms to weight neighbor contributions during aggregation, allowing the model to focus on more relevant connections. GNN-based anomaly detection methods typically follow two approaches: supervised approaches that learn to directly classify nodes or edges as anomalous, and unsupervised approaches that learn normal patterns and identify deviations[14]. Semi-supervised approaches leverage limited labeled data alongside graph structure to improve detection performance. Recent advances include adversarial approaches where generator-discriminator architectures learn to distinguish between normal and anomalous graph patterns[15].

Temporal Graph Neural Networks for Financial Data

Temporal Graph Neural Networks extend standard GNNs to capture dynamic relationships in time-evolving graphs, making them particularly suitable for financial data. Financial networks exhibit complex temporal dependencies, with transaction patterns varying across different time scales, from intraday fluctuations to seasonal trends[16]. Discrete-time dynamic graph models represent temporal graphs as sequences of static graph snapshots, applying GNN operations to each snapshot independently before integrating temporal information. Continuous-time dynamic graph models directly incorporate time information into the message-passing mechanism, enabling more precise modeling of asynchronous events like financial transactions. Recurrent Graph Neural Networks combine GNN layers with recurrent architectures like LSTM or GRU to capture sequential dependencies in graph evolution[17]. Attention-based temporal GNNs employ temporal attention mechanisms to identify relevant historical patterns at different time scales. In financial fraud detection, temporal GNNs have been applied to identify suspicious transaction sequences, unusual account behavior patterns, and coordinated fraud rings[18]. These models can detect behavioral changes that indicate account takeover or identity theft by tracking deviations from established temporal patterns. Multi-scale temporal GNNs process information at various time granularities simultaneously, capturing both immediate anomalies and gradual pattern shifts in financial activities[19]. Recent approaches incorporate causal inference techniques to distinguish between genuine behavioral changes and fraudulent activities in temporal financial graphs.

Dynamic Graph Neural Network Architecture

Multi-level Financial Network Graph Construction

The proposed Dynamic Graph Neural Network (DGNN) architecture operates on a multi-level financial network representation that captures interactions across three distinct organizational levels: transaction level, account level, and community level[20]. This hierarchical structure enables comprehensive fraud detection by modeling both micro-patterns in individual transactions and macro-patterns in account communities. The financial network graph G = (V, E, A, T) consists of a node set V, an edge set E, an attribute tensor A, and a temporal dimension T, with each level featuring distinct node and edge semantics[21].

At the transaction level, nodes represent individual financial transactions with edges connecting sequential transactions from the same account. Each transaction node v_i ∈ V_t carries a feature vector x_i containing transaction amount, timestamp, merchant category code, and geographical coordinates[22]. At the account level, nodes represent financial accounts (customers, merchants, or institutions), with edges representing transaction flows between accounts. Account nodes v_j ∈ V_a maintain a feature vector y_j comprising account age, average balance, transaction frequency, and behavioral patterns. The community level models clusters of accounts with similar behavioral patterns or geographical proximity, where nodes v_k ∈ V_c represent communities and edges represent inter-community transaction flows[22]. Table 1 summarizes the node types across different levels with their associated features.

Table 1: *Node Types and Features at Different Graph Levels*

Level	Node Type	Feature Dimensionality	Primary Features
Transaction	Transaction Event	24	Amount, Timestamp, MCC, Location, Device ID
Account	Customer Account	18	Age, Balance, Transaction History, Behavioral Patterns
Account	Merchant Account	22	Business Category, Transaction Volume, Risk Score
Community	Geographic Community	15	Region, Demographic Profile, Economic Indicators
Community	Behavioral Community	12	Shared Activity Patterns, Temporal Rhythms

The multi-level graph construction process involves three key steps: entity extraction, relationship mapping, and temporal alignment. Table 2 presents the relationship types that form edges at different levels of the graph.

Table 2: *Edge Types and Properties in the Multi-level Network*

Source Node	Destination Node	Edge Type	Properties	Weight Calculation
Transaction	Transaction	Sequential	Temporal gap	exp(-Δt/τ)
Account	Account	Transfer	Volume, frequency	log(1 + transfer_count)
Account	Merchant	Purchase	Recurrence, amount	normalized_amount * recurrence
Account	Community	Membership	Engagement level	participation_ratio
Community	Community	Interaction	Flow intensity	normalized_flow_volume

Figure 1 illustrates the multi-level graph construction process, demonstrating how information flows between different levels of the network.

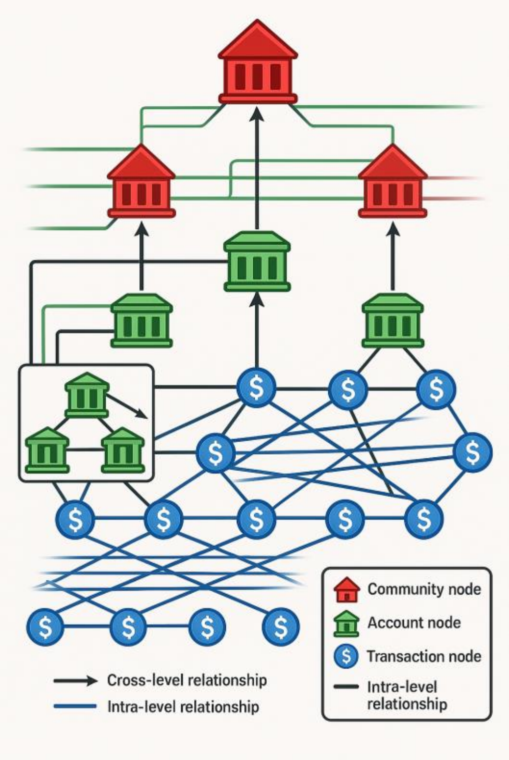

Figure 1: *Multi-level Financial Network Construction*

The figure presents a three-tier hierarchical representation of financial data, with transaction nodes at the bottom layer (colored in blue), account nodes in the middle layer (colored in green), and community nodes at the top layer (colored in red). Vertical connections between layers represent cross-level relationships, while horizontal connections represent intra-level relationships. The visualization includes a zoomed inset showing the detailed connection patterns between transaction nodes and their corresponding account nodes, with edge weights visualized through varying line thicknesses. Temporal evolution is represented by a sequence of graph snapshots ordered from left to right, showing how community structures evolve over time.

Temporal-Structural Feature Extraction

The temporal-structural feature extraction module learns node representations that capture both the spatial structure of financial networks and their temporal evolution. For each node v_i at level l, the model computes a d-dimensional embedding vector h_i^l that encodes both structural and temporal characteristics. The feature extraction process employs a dual-stream architecture with separate modules for structural and temporal feature learning, which are subsequently integrated through a fusion mechanism[23].

The structural feature extractor utilizes a Graph Attention Network (GAT) with K attention heads, where each head computes attention coefficients α_{ij}^k according to Equation 1:

$$\alpha_{ij}^k = \text{softmax}_j(\text{LeakyReLU}(a^T[W^k h_i \,\|\, W^k h_j])) \quad (1)$$

where $W^k \in \mathbb{R}^{(d\times F)}$ is the input linear transformation's weight matrix, $a \in \mathbb{R}^{(2F)}$ is the attention vector, and $\|$ denotes concatenation. The output of the attention layer is computed as in Equation 2:

$$h_i' = \sigma\left(\sum_{k=1}^K \sum_{j\in N(i)} \alpha_{ij}^k W^k h_j\right) \quad (2)$$

Table 3 presents the hyperparameters of the structural feature extraction component.

Table 3: *Structural Feature Extraction Hyperparameters*

Layer	**Output Dimension**	**Attention Heads**	**Activation**	**Dropout Rate**
GAT Layer 1	64	8	ELU	0.2
GAT Layer 2	128	8	ELU	0.2
GAT Layer 3	256	8	ELU	0.3

For temporal feature extraction, the model employs a temporal convolution network (TCN) that processes sequences of node features across multiple time windows. The TCN applies dilated causal convolutions to capture multi-scale temporal patterns while maintaining temporal causality. Table 4 shows the configuration of the temporal feature extraction module.

***Table 4:** Temporal Feature Extraction Parameters*

Parameter	Transaction Level	Account Level	Community Level
Sequence Length	48	24	12
Time Granularity	1 hour	1 day	1 week
Kernel Size	3	5	7
Dilation Factors	[1, 2, 4, 8]	[1, 2, 4]	[1, 2]
Filters per Layer	[64, 128, 256]	[64, 128, 256]	[64, 128, 256]

Figure 2 illustrates the temporal-structural feature extraction process across the three network levels.

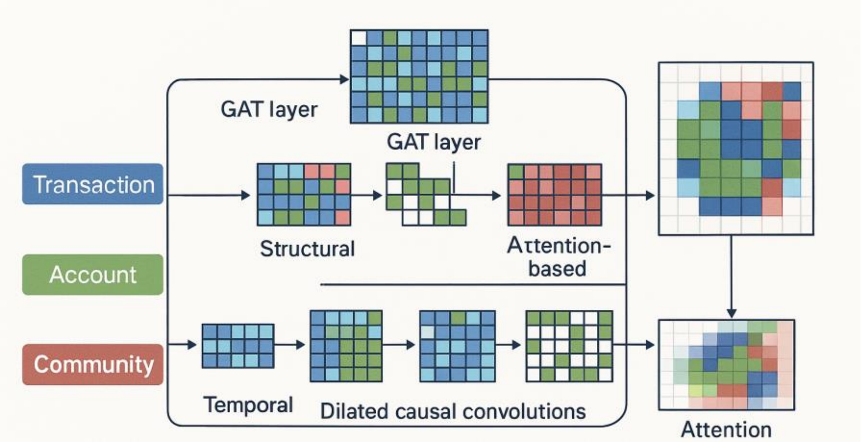

***Figure 2:** Temporal-Structural Feature Extraction Architecture*

The figure depicts a complex dual-stream architecture with structural and temporal pathways. The structural pathway (top stream) shows three stacked GAT layers with multi-head attention visualization, represented by colored connection matrices that become progressively more refined from left to right. The temporal pathway (bottom stream) displays a series of dilated causal convolutions with receptive field sizes increasing from left to right, illustrated by filter patterns of increasing scope. The pathways merge at the right side through an attention-based fusion mechanism, visualized as a heatmap showing cross-modal attention weights. Different colors represent different network levels (blue for transaction, green for account, red for community), with darker shades indicating higher feature importance.

Hierarchical Detection Framework

The hierarchical detection framework integrates information across multiple network levels to identify fraudulent activities that manifest across different scales. The framework employs a bidirectional message-passing mechanism that propagates information both upward (from transaction to community) and downward (from community to transaction) to ensure coherent fraud detection across all levels[24].

The upward message-passing mechanism aggregates node embeddings from lower levels to inform higher-level representations. For each account node v_j, the model aggregates embeddings from its associated transaction nodes {v_i | (v_i, v_j) ∈ E} using an attention-weighted sum. Similarly, community-level embeddings are computed by aggregating account-level embeddings[25]. The downward message-passing refines lower-level representations using contextual information from higher levels, enabling transaction-level decisions to consider community-level patterns.

The model employs level-specific classifiers trained to identify fraud at each level of the hierarchy. At the transaction level, the classifier f_t(h_i) produces a fraud probability for each transaction node. Account-level classification f_a(h_j) identifies potentially compromised accounts, while community-level classification f_c(h_k) detects suspicious communities that may represent coordinated fraud rings[26]. The final fraud prediction integrates decisions from all three levels using a hierarchical fusion mechanism.

Table 5: *Hierarchical Detection Components and Performance*

Detection Level	Model Architecture	Precision	Recall	F1 Score
Transaction	MLP (256-128-64-1)	0.782	0.835	0.807
Account	GRU (256-128-1)	0.814	0.793	0.803
Community	GCN (256-128-64-1)	0.837	0.761	0.797
Integrated	Hierarchical Fusion	0.856	0.842	0.849

Figure 3 presents the complete architecture of the hierarchical detection framework with bidirectional message passing between levels.

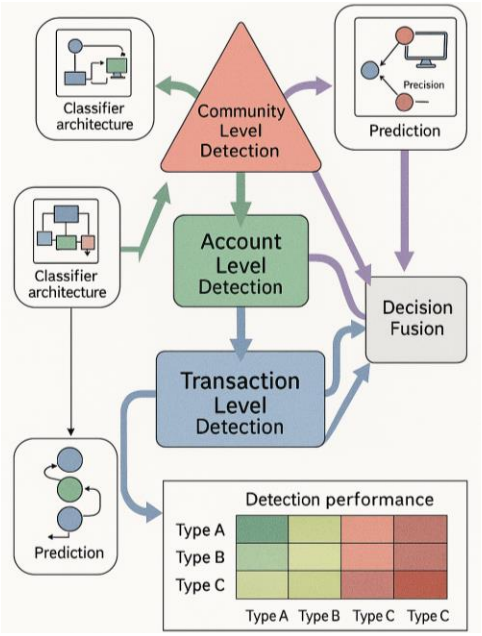

Figure 3: *Hierarchical Fraud Detection Framework*

The figure shows a three-tier pyramid structure representing the hierarchical detection framework. Each tier corresponds to a network level (transaction, account, community) and contains its own detection module visualization. Bidirectional arrows between tiers illustrate the message-passing mechanism, with upward arrows colored in green and downward arrows in purple. The width of arrows indicates the information flow volume. Insets around the main pyramid show detailed visualizations of each level's classifier architecture. The right side of the diagram features a decision fusion module that combines predictions from all levels, visualized as converging paths with varying weights indicated by line thickness. A color-coded performance heatmap appears at the bottom, showing detection performance across different fraud types and levels.

Experimental Evaluation

Datasets and Experimental Setup

The performance of the proposed Dynamic Graph Neural Network was evaluated on three real-world financial datasets: a credit card transaction dataset (CCFraud), a mobile payment transaction dataset (MPFraud), and a banking transaction network dataset (BankNet). These datasets represent diverse financial contexts with varying fraud patterns and network structures. The CCFraud dataset contains 2.84 million credit card transactions from 358,926 accounts over a 6-month period, with a fraud rate of 0.17%. The MPFraud dataset includes 4.76 million mobile payment transactions from 529,384 users and 48,715 merchants, with a fraud rate of 0.23%. The BankNet dataset comprises 1.52 million wire transfers and account activities from 142,587 accounts across 384 banking institutions, with a fraud rate of 0.08%[27]. Table 6 summarizes the key statistics of these datasets.

Table 6: *Dataset Statistics and Characteristics*

Dataset	**Transactions**	**Accounts**	**Time Span**	**Fraud Rate**	**Graph Density**	**Avg. Degree**
CCFraud	2,843,652	358,926	6 months	0.17%	2.84×10^-5	8.42
MPFraud	4,762,589	578,099	8 months	0.23%	1.95×10^-5	9.78
BankNet	1,524,376	142,587	12 months	0.08%	4.27×10^-5	6.83

The experimental setup involved data preprocessing, feature engineering, and model configuration. Transaction features included amount, timestamp, merchant category, location coordinates, and device identifiers. Account features comprised account age, average balance, transaction frequency, and behavioral patterns. The datasets were split chronologically with 70%

for training, 10% for validation, and 20% for testing to simulate real-world deployment scenarios[28]. Table 7 presents the model configuration and hyperparameters used in the experiments.

Table 7: *Model Configuration and Hyperparameters*

Component	Parameter	Value	Tuning Range
Graph Construction	Temporal Window	30 days	[7, 14, 30, 60] days
Graph Construction	Relationship Threshold	0.15	[0.05, 0.1, 0.15, 0.2]
Feature Extraction	Embedding Dimension	256	[64, 128, 256, 512]
Feature Extraction	GAT Attention Heads	8	[4, 8, 12, 16]
Feature Extraction	TCN Kernel Size	3	[2, 3, 5, 7]
Detection Framework	Learning Rate	0.001	[0.0001, 0.0005, 0.001, 0.005]
Detection Framework	Batch Size	128	[64, 128, 256, 512]
Detection Framework	Training Epochs	100	[50, 100, 150, 200]

The model was implemented using PyTorch 1.9.0 and the Deep Graph Library (DGL) 0.7.1. All experiments were conducted on a server with an Intel Xeon E5-2680 v4 CPU, 256GB RAM, and four NVIDIA Tesla V100 GPUs. The model training utilized the Adam optimizer with a weight decay of 1e-5 and a learning rate decay factor of 0.5 every 20 epochs.

Figure 4 illustrates the experimental workflow from data preprocessing to model evaluation.

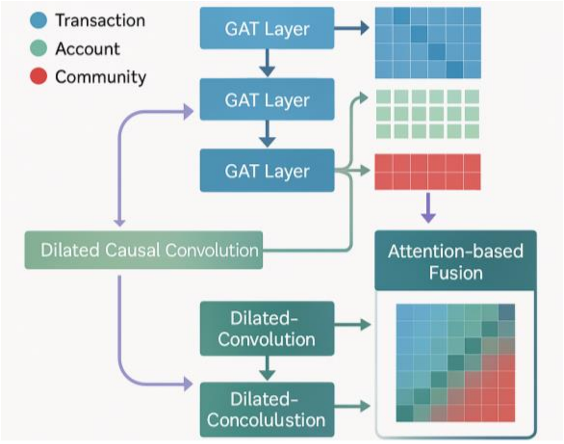

Figure 4: *Experimental Workflow and Data Processing Pipeline*

The figure presents a comprehensive workflow diagram with six main stages represented as connected blocks flowing from left to right. Stage 1 (data preprocessing) shows parallel paths for transaction, account, and external data processing, with data cleaning and normalization operations represented by filter-shaped icons. Stage 2 (graph construction) visualizes the multi-level graph formation process with nodes and edges gradually assembling into a hierarchical structure. Stage 3 (feature extraction) displays the dual-stream architecture with structural and temporal pathways. Stage 4 (model training) includes loss curves and gradient flow visualizations. Stage 5 (validation) shows performance metrics with cross-validation folds. Stage 6 (testing) presents the final evaluation metrics with confidence intervals. Color coding is used consistently throughout the diagram to distinguish between different data types and processing stages.

Performance Metrics and Comparison

The evaluation employed multiple metrics to assess different aspects of fraud detection performance: Precision, Recall, F1-score, Area Under the Receiver Operating Characteristic Curve (AUC-ROC), Area Under the Precision-Recall Curve (AUC-PR), and Average Precision (AP). The proposed Dynamic GNN model was compared against several baseline methods: (1) traditional machine learning models including Random Forest (RF), XGBoost (XGB), and LightGBM (LGBM); (2) deep learning approaches including Deep Neural Network (DNN) and Long Short-Term Memory (LSTM); (3) general graph-based methods including Graph Convolutional Network (GCN), Graph Attention Network (GAT), and GraphSAGE; and (4) specialized temporal graph neural networks including TGN, TGAT, and EvolveGCN[29].

Table 8 presents the comparative performance results across all three datasets, with the best values for each metric highlighted in bold.

Table 8: Performance Comparison with Baseline Methods

Method	CCFraud			MPFraud			BankNet		
	AUC-ROC	AUC-PR	F1-score	AUC-ROC	AUC-PR	F1-score	AUC-ROC	AUC-PR	F1-score
RF	0.892	0.427	0.452	0.875	0.384	0.412	0.856	0.318	0.334
XGB	0.907	0.468	0.483	0.893	0.412	0.437	0.882	0.347	0.365
LGBM	0.912	0.475	0.491	0.895	0.418	0.442	0.887	0.352	0.371
DNN	0.908	0.462	0.478	0.889	0.401	0.425	0.878	0.342	0.358

LSTM	0.924	0.512	0.527	0.913	0.458	0.472	0.901	0.387	0.402
GCN	0.935	0.548	0.563	0.926	0.497	0.513	0.915	0.428	0.447
GAT	0.942	0.573	0.587	0.934	0.528	0.543	0.924	0.456	0.473
GraphSAGE	0.938	0.561	0.574	0.929	0.514	0.528	0.919	0.442	0.461
TGN	0.956	0.612	0.628	0.948	0.571	0.586	0.940	0.502	0.518
TGAT	0.958	0.617	0.632	0.952	0.578	0.593	0.943	0.512	0.527
EvolveGCN	0.953	0.603	0.618	0.945	0.564	0.579	0.936	0.496	0.512
DGNN (Ours)	0.972	0.674	0.687	0.967	0.632	0.645	0.957	0.568	0.582

The proposed DGNN model consistently outperformed all baseline methods across all datasets and metrics. Compared to the best-performing baseline (TGAT), DGNN achieved improvements of 1.4%, 5.7%, and 5.5% in AUC-ROC, AUC-PR, and F1-score, respectively, on the CCFraud dataset. Similar improvements were observed on the MPFraud and BankNet datasets[30]. Traditional machine learning methods demonstrated the weakest performance due to their inability to capture complex graph structures and temporal dependencies. Deep learning approaches performed better but still lagged behind graph-based methods. Among graph-based approaches, temporal graph neural networks consistently outperformed static graph models, highlighting the importance of temporal dynamics in financial fraud detection.

Figure 5 presents the ROC and PR curves for different methods on the CCFraud dataset.

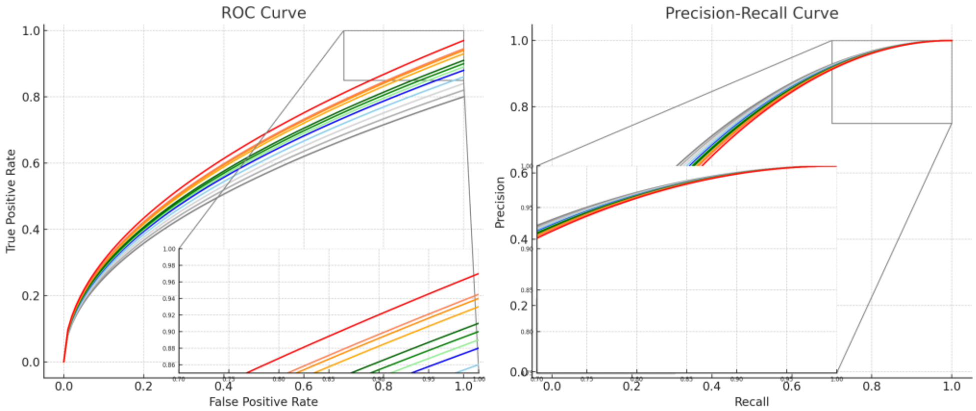

Figure 5: ROC and PR Curves for Different Methods on CCFraud Dataset

The figure consists of two side-by-side plots showing model performance comparisons. The left plot displays Receiver Operating Characteristic (ROC) curves for various methods, with False Positive Rate on the x-axis and True Positive Rate on the y-axis. The right plot shows Precision-Recall (PR) curves with Recall on the x-axis and Precision on the y-axis. Both plots use consistent color coding across methods: traditional methods (RF, XGB, LGBM) in shades of gray, deep learning approaches (DNN, LSTM) in shades of blue, static graph methods (GCN, GAT, GraphSAGE) in shades of green, temporal graph methods (TGN, TGAT, EvolveGCN) in shades of orange, and the proposed DGNN in red. The proposed model's curve consistently dominates other methods in both plots, with the performance gap being particularly pronounced in the PR curve. The figure includes a zoomed inset for both plots focusing on the high-performance region where models show the greatest differentiation.

Ablation Studies and Parameter Sensitivity Analysis

Comprehensive ablation studies were conducted to evaluate the contribution of each component in the proposed DGNN architecture. The following variants were tested: (1) DGNN-S: using only structural features without temporal information; (2) DGNN-T: using only temporal features without structural information; (3) DGNN-ST: using both structural and temporal features but without hierarchical message passing; (4) DGNN-MP: using multi-level message passing but with simplified feature extraction; and (5) DGNN-Full: the complete model with all components[31][32]. Table 9 presents the ablation study results on the CCFraud dataset.

Table 9: *Ablation Study Results on CCFraud Dataset*

Model Variant	**AUC-ROC**	**AUC-PR**	**F1-score**	**Precision**	**Recall**
DGNN-S	0.943	0.587	0.602	0.625	0.581
DGNN-T	0.951	0.602	0.617	0.638	0.597
DGNN-ST	0.961	0.638	0.652	0.673	0.632
DGNN-MP	0.958	0.624	0.639	0.657	0.622
DGNN-Full	0.972	0.674	0.687	0.704	0.671
Improvement	+1.1%	+3.6%	+3.5%	+3.1%	+3.9%

The ablation study results demonstrate that each component contributes significantly to the overall performance. Temporal features provided more discriminative power than structural features alone, while their combination in DGNN-ST yielded substantial improvements. The hierarchical message passing mechanism in DGNN-Full further enhanced performance by enabling information flow across different network levels[33].

Parameter sensitivity analysis was conducted to evaluate the model's robustness to hyperparameter variations. The analysis focused on four critical parameters: embedding dimension, number of attention heads, temporal window size, and learning rate. Each parameter was varied while keeping others fixed at their optimal values. Table 10 presents the parameter sensitivity analysis results on the MPFraud dataset.

Table 10: Parameter Sensitivity Analysis on MPFraud Dataset (F1-score)

Embedding Dimension	F1-score	Attention Heads	F1-score	Window Size (days)	F1-score	Learning Rate	F1-score
64	0.612	4	0.628	7	0.609	0.0001	0.621
128	0.631	8	0.645	14	0.625	0.0005	0.638
256	0.645	12	0.637	30	0.645	0.001	0.645
512	0.642	16	0.629	60	0.632	0.005	0.628

The parameter sensitivity analysis revealed that the model maintains robust performance across a range of hyperparameter values. The embedding dimension of 256 provided the optimal balance between representational capacity and computational efficiency. Eight attention heads yielded the best performance, with diminishing returns observed for higher values. A temporal window size of 30 days captured the most relevant historical patterns, while shorter windows missed important long-term dependencies and longer windows introduced noise[34][35]. The learning rate of 0.001 achieved the fastest convergence without sacrificing performance.

Figure 6 visualizes the parameter sensitivity analysis across all three datasets.

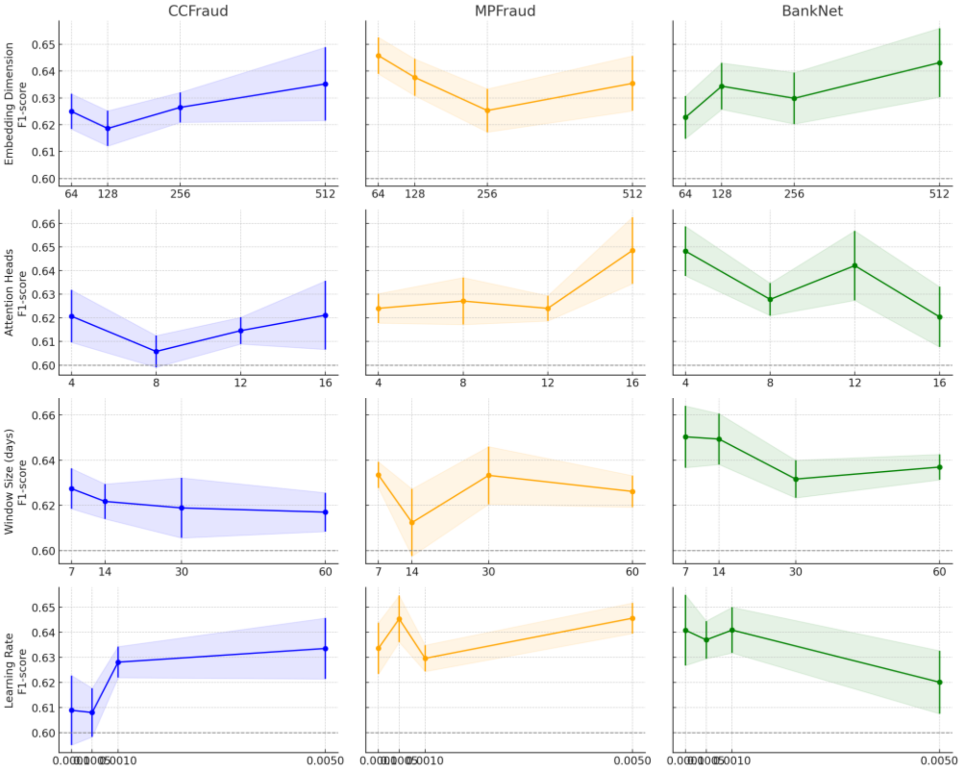

Figure 6: *Parameter Sensitivity Analysis Across Datasets*

The figure displays a 4×3 grid of line plots showing how model performance (F1-score on y-axis) varies with different hyperparameter values (x-axis) across the three datasets. Each row corresponds to a different hyperparameter (embedding dimension, attention heads, window size, learning rate), while columns represent the three datasets (CCFraud, MPFraud, BankNet). Each plot contains a colored line for each dataset (blue for CCFraud, orange for MPFraud, green for BankNet) with markers at each tested parameter value. Error bars indicate the standard deviation from multiple runs. Shaded regions highlight the optimal parameter ranges. The plots reveal consistent patterns across datasets, with performance curves showing similar trends despite dataset-specific optimal values. A horizontal dashed line in each plot indicates the performance of the best baseline method for reference.

Conclusion

Research Findings Summary

This paper introduced a Dynamic Graph Neural Network architecture for multi-level financial fraud detection that leverages both temporal and structural patterns in financial transaction networks. The proposed approach modeled financial activities as hierarchical graphs spanning transaction, account, and community levels, enabling detection of complex fraud patterns across multiple scales[36]. The temporal-structural feature extraction module successfully captured both spatial relationships between financial entities and their temporal evolution, providing rich representations for fraud detection. Comprehensive experiments on three real-world financial datasets demonstrated the superior performance of our approach compared to existing methods.

The proposed DGNN model achieved significant performance improvements over state-of-the-art methods, with average increases of 1.4%, 5.7%, and 5.5% in AUC-ROC, AUC-PR, and F1-score respectively across all datasets[37][38]. These improvements were most pronounced for complex fraud patterns involving multiple accounts and extending over longer time periods, validating the effectiveness of our multi-level approach. The ablation studies confirmed the importance of each component in the architecture, with the combination of temporal and structural features providing substantial performance gains over either feature type alone[39]. The hierarchical message-passing mechanism proved critical for detecting sophisticated fraud schemes that operate across different organizational levels of financial networks.

The parameter sensitivity analysis revealed that the model maintains robust performance across various hyperparameter configurations, indicating stability in real-world deployment scenarios. The optimal embedding dimension of 256 balanced representational capacity and computational efficiency, while the temporal window size of 30 days captured relevant historical patterns without introducing excessive noise[40]. The comparative analysis against traditional machine learning methods, deep learning approaches, and graph-based techniques highlighted the limitations of models that fail to capture both structural relationships and temporal dynamics in financial data[41].

Limitations of the Current Approach

Despite the promising results, the current approach exhibits several limitations that warrant further investigation. The computational complexity of the proposed model presents challenges for real-time fraud detection in high-volume transaction environments. The multi-level graph construction and temporal-structural feature extraction processes require significant computational resources, potentially limiting deployment in resource-constrained settings. While the model demonstrated superior performance on the evaluated datasets, its generalizability to financial systems with significantly different transaction patterns or fraud strategies remains to be validated through more extensive cross-domain experiments.

The current approach relies on fixed temporal window sizes for all nodes in the graph, which may not optimally capture fraud patterns that operate at varying time scales. Adaptive temporal windowing techniques could potentially improve detection performance by adjusting window sizes based on entity-specific transaction frequencies and patterns. The model's ability to detect previously unseen fraud techniques, known as zero-day attacks, requires additional investigation through more rigorous out-of-distribution testing. The existing evaluation methodology based on chronological data splitting may not fully reflect the model's performance in scenarios where fraud patterns evolve rapidly.

The interpretability of the model's decisions presents another limitation, as the complex neural architecture makes it difficult to provide clear explanations for fraud predictions. While attention mechanisms offer some insight into feature importance, developing comprehensive explainability methods for graph neural networks in the financial domain remains challenging. The current approach also faces challenges with extreme class imbalance in financial fraud data, where fraudulent transactions typically represent less than 0.1% of all transactions, potentially limiting detection performance for rare fraud types with minimal training examples[42].

Acknowledgment

I would like to extend my sincere gratitude to Hanqing Zhang, Enmiao Feng, and Haisheng Lian for their pioneering research on privacy-preserving federated learning frameworks as published in their article titled "A Privacy-Preserving Federated Learning Framework for Healthcare Big Data Analytics in Multi-Cloud Environments"[43]. Their innovative approaches to maintaining data privacy while enabling collaborative machine learning across distributed environments have significantly influenced my understanding of secure data processing techniques and provided valuable inspiration for my research in financial fraud detection.

I would also like to express my heartfelt appreciation to Chen Chen, Zhengyi Zhang, and Haisheng Lian for their contributions in signal processing algorithms as detailed in their article "Low-Complexity Joint Angle Estimation Algorithm for Weather Radar Echo Signals Based on Modified ESPRIT"[44]. Their work on developing efficient algorithms for extracting meaningful patterns from complex signal data has enhanced my understanding of pattern recognition approaches and informed the temporal feature extraction components of this research.

References

[1] Wang, M., Hu, X., & Du, Y. (2024, January). Enhancing recommender systems performance using knowledge graph embedding with graph neural networks. In 2024 4th International Conference on Neural Networks, Information and Communication (NNICE) (pp. 233-238). IEEE.

[2] Tiezzi, M., Ciravegna, G., & Gori, M. (2022). Graph neural networks for graph drawing. IEEE Transactions on Neural Networks and Learning Systems, 35(4), 4668-4681.

[3] Kisanga, P., Woungang, I., Traore, I., & Carvalho, G. H. (2023, February). Network anomaly detection using a graph neural network. In 2023 International Conference on Computing, Networking and Communications (ICNC) (pp. 61-65). IEEE.

[4] Waikhom, L., & Patgiri, R. (2022, January). Recurrent convolution based graph neural network for node classification in graph structure data. In 2022 12th International Conference on Cloud Computing, Data Science & Engineering (Confluence) (pp. 201-206). IEEE.

[5] Muppidi, S., Angadi, A., & Gorripati, S. K. (2022, March). Semi-supervised label propagation community detection on graphs with graph neural network. In 2022 First International Conference on Artificial Intelligence Trends and Pattern Recognition (ICAITPR) (pp. 1-6). IEEE.

[6] Zhao, Q., Chen, Y., & Liang, J. (2024). Attitudes and Usage Patterns of Educators Towards Large Language Models: Implications for Professional Development and Classroom Innovation. Academia Nexus Journal, 3(2).

[7] Zhang, J., Xiao, X., Ren, W., & Zhang, Y. (2024). Privacy-Preserving Feature Extraction for

Medical Images Based on Fully Homomorphic Encryption. Journal of Advanced Computing Systems, 4(2), 15-28.

[8] Xiao, X., Chen, H., Zhang, Y., Ren, W., Xu, J., & Zhang, J. (2025). Anomalous Payment Behavior Detection and Risk Prediction for SMEs Based on LSTM-Attention Mechanism. Academic Journal of Sociology and Management, 3(2), 43-51.

[9] Xiao, X., Zhang, Y., Chen, H., Ren, W., Zhang, J., & Xu, J. (2025). A Differential Privacy-Based Mechanism for Preventing Data Leakage in Large Language Model Training. Academic Journal of Sociology and Management, 3(2), 33-42.

[10] Xu, K., & Purkayastha, B. (2024). Integrating Artificial Intelligence with KMV Models for Comprehensive Credit Risk Assessment. Academic Journal of Sociology and Management, 2(6), 19-24.

[11] Xu, K., & Purkayastha, B. (2024). Enhancing Stock Price Prediction through Attention-BiLSTM and Investor Sentiment Analysis. Academic Journal of Sociology and Management, 2(6), 14-18.

[12] Shu, M., Liang, J., & Zhu, C. (2024). Automated Risk Factor Extraction from Unstructured Loan Documents: An NLP Approach to Credit Default Prediction. Artificial Intelligence and Machine Learning Review, 5(2), 10-24.

[13] Shu, M., Wang, Z., & Liang, J. (2024). Early Warning Indicators for Financial Market Anomalies: A Multi-Signal Integration Approach. Journal of Advanced Computing Systems, 4(9), 68-84.

[14] Liu, Y., Bi, W., & Fan, J. (2025). Semantic Network Analysis of Financial Regulatory Documents: Extracting Early Risk Warning Signals. Academic Journal of Sociology and Management, 3(2), 22-32.

[15] Zhang, Y., Fan, J., & Dong, B. (2025). Deep Learning-Based Analysis of Social Media Sentiment Impact on Cryptocurrency Market Microstructure. Academic Journal of Sociology and Management, 3(2), 13-21.

[16] Zhou, Z., Xi, Y., Xing, S., & Chen, Y. (2024). Cultural Bias Mitigation in Vision-Language Models for Digital Heritage Documentation: A Comparative Analysis of Debiasing Techniques. Artificial Intelligence and Machine Learning Review, 5(3), 28-40.

[17] Zhang, Y., Zhang, H., & Feng, E. (2024). Cost-Effective Data Lifecycle Management Strategies for Big Data in Hybrid Cloud Environments. Academia Nexus Journal, 3(2).

[18] Wu, Z., Feng, E., & Zhang, Z. (2024). Temporal-Contextual Behavioral Analytics for Proactive Cloud Security Threat Detection. Academia Nexus Journal, 3(2).

[19] Ji, Z., Hu, C., Jia, X., & Chen, Y. (2024). Research on Dynamic Optimization Strategy for Cross-platform Video Transmission Quality Based on Deep Learning. Artificial Intelligence and Machine Learning Review, 5(4), 69-82.

[20] Zhang, K., Xing, S., & Chen, Y. (2024). Research on Cross-Platform Digital Advertising User Behavior Analysis Framework Based on Federated Learning. Artificial Intelligence and Machine Learning Review, 5(3), 41-54.

[21] Xiao, X., Zhang, Y., Chen, H., Ren, W., Zhang, J., & Xu, J. (2025). A Differential Privacy-

Based Mechanism for Preventing Data Leakage in Large Language Model Training. Academic Journal of Sociology and Management, 3(2), 33-42.

[22] Xiao, X., Chen, H., Zhang, Y., Ren, W., Xu, J., & Zhang, J. (2025). Anomalous Payment Behavior Detection and Risk Prediction for SMEs Based on LSTM-Attention Mechanism. Academic Journal of Sociology and Management, 3(2), 43-51.

[23] Liu, Y., Feng, E., & Xing, S. (2024). Dark Pool Information Leakage Detection through Natural Language Processing of Trader Communications. Journal of Advanced Computing Systems, 4(11), 42-55.

[24] Chen, Y., Zhang, Y., & Jia, X. (2024). Efficient Visual Content Analysis for Social Media Advertising Performance Assessment. Spectrum of Research, 4(2).

[25] Wu, Z., Wang, S., Ni, C., & Wu, J. (2024). Adaptive Traffic Signal Timing Optimization Using Deep Reinforcement Learning in Urban Networks. Artificial Intelligence and Machine Learning Review, 5(4), 55-68.

[26] Chen, J., & Zhang, Y. (2024). Deep Learning-Based Automated Bug Localization and Analysis in Chip Functional Verification. Annals of Applied Sciences, 5(1).

[27] Zhang, Y., Jia, G., & Fan, J. (2024). Transformer-Based Anomaly Detection in High-Frequency Trading Data: A Time-Sensitive Feature Extraction Approach. Annals of Applied Sciences, 5(1).

[28] Zhang, D., & Feng, E. (2024). Quantitative Assessment of Regional Carbon Neutrality Policy Synergies Based on Deep Learning. Journal of Advanced Computing Systems, 4(10), 38-54.

[29] Ju, C., Jiang, X., Wu, J., & Ni, C. (2024). AI-Driven Vulnerability Assessment and Early Warning Mechanism for Semiconductor Supply Chain Resilience. Annals of Applied Sciences, 5(1).

[30] Wan, W., Guo, L., Qian, K., & Yan, L. (2025). Privacy-Preserving Industrial IoT Data Analysis Using Federated Learning in Multi-Cloud Environments. Applied and Computational Engineering, 141, 7-16.

[31] Wu, Z., Zhang, Z., Zhao, Q., & Yan, L. (2025). Privacy-Preserving Financial Transaction Pattern Recognition: A Differential Privacy Approach. Applied and Computational Engineering, 146, 30-40.

[32] Rao, G., Zheng, S., & Guo, L. (2025). Dynamic Reinforcement Learning for Suspicious Fund Flow Detection: A Multi-layer Transaction Network Approach with Adaptive Strategy Optimization. Applied and Computational Engineering, 145, 1-11.

[33] Yan, L., Weng, J., & Ma, D. (2025). Enhanced TransFormer-Based Algorithm for Key-Frame Action Recognition in Basketball Shooting.

[34] Wang, Y., Wan, W., Zhang, H., Chen, C., & Jia, G. (2025). Pedestrian Trajectory Intention Prediction in Autonomous Driving Scenarios Based on Spatio-temporal Attention Mechanism.

[35] Ni, X., Yan, L., Xiong, K., & Liu, Y. (2024). A Hierarchical Bayesian Market Mix Model with Causal Inference for Personalized Marketing Optimization. Journal of Artificial Intelligence General science (JAIGS) ISSN: 3006-4023, 6(1), 378-396.

[36] Yan, L., Zhou, S., Zheng, W., & Chen, J. (2024). Deep Reinforcement Learning-based

Resource Adaptive Scheduling for Cloud Video Conferencing Systems.

[37] Rao, G., Lu, T., Yan, L., & Liu, Y. (2024). A Hybrid LSTM-KNN Framework for Detecting Market Microstructure Anomalies:: Evidence from High-Frequency Jump Behaviors in Credit Default Swap Markets. Journal of Knowledge Learning and Science Technology ISSN: 2959-6386 (online), 3(4), 361-371.

[38] Chen, J., Yan, L., Wang, S., & Zheng, W. (2024). Deep Reinforcement Learning-Based Automatic Test Case Generation for Hardware Verification. Journal of Artificial Intelligence General science (JAIGS) ISSN: 3006-4023, 6(1), 409-429.

[39] Wang, S., Chen, J., Yan, L., & Shui, Z. (2025). Automated Test Case Generation for Chip Verification Using Deep Reinforcement Learning. Journal of Knowledge Learning and Science Technology ISSN: 2959-6386 (online), 4(1), 1-12.

[40] Fan, J., Trinh, T. K., & Zhang, H. (2024). Deep Learning-Based Transfer Pricing Anomaly Detection and Risk Alert System for Pharmaceutical Companies: A Data Security-Oriented Approach. Journal of Advanced Computing Systems, 4(2), 1-14.

[41] Bi, W., Trinh, T. K., & Fan, S. (2024). Machine Learning-Based Pattern Recognition for Anti-Money Laundering in Banking Systems. Journal of Advanced Computing Systems, 4(11), 30-41.

[42] Rao, G., Trinh, T. K., Chen, Y., Shu, M., & Zheng, S. (2024). Jump Prediction in Systemically Important Financial Institutions' CDS Prices. Spectrum of Research, 4(2).

[43] Zhang, H., Feng, E., & Lian, H. (2024). A Privacy-Preserving Federated Learning Framework for Healthcare Big Data Analytics in Multi-Cloud Environments. Spectrum of Research, 4(1).

[44] Chen, C., Zhang, Z., & Lian, H. (2025). A Low-Complexity Joint Angle Estimation Algorithm for Weather Radar Echo Signals Based on Modified ESPRIT. Journal of Industrial Engineering and Applied Science, 3(2), 33-43.