**IAES International Journal of Artificial Intelligence (IJ-AI)**
Vol. 15, No. 6, December 2026, pp. 5605~5622
ISSN: 2252-8938, DOI: 10.11591/ijai.v15.i6.pp5605-5622 

# TransGRAFCNet: combining capsule networks and graph attention to detect transaction fraud

**Anish Mary George, Sudha Seethambaram**
Department of Computer Applications, Hindustan Institute of Technology and Science, Chennai, India

**Article Info**

***Article history:***

Received Nov 8, 2025
Revised Aug 8, 2026
Accepted Aug 31, 2026

***Keywords:***

Capsule network
Class imbalance
Firefly algorithm
GraphSMOTE
Fraud detection
Graph neural networks
Transaction networks

**ABSTRACT**

As digital transactions continue to become more common, the risk of financial fraud increases accordingly. Standard fraud detection methods often lack interpretability, struggle with complex data structures, and perform poorly on imbalanced datasets. This paper proposes transformer–graph attention–capsule network (TransGRAFCNet), a hybrid deep learning framework for the identification of financial fraud. The model uses graph attention networks (GATs) to capture relational dependencies, capsule networks (CapsNets) for hierarchical spatial feature extraction, graph synthetic minority over-sampling technique (GraphSMOTE) to address class imbalance, and the firefly optimization algorithm for adaptive weight optimization. Dual-path fusion of LeCun network (LeNet) and principal component analysis (PCA) for feature learning by combining automatic feature interactions and effective dimensionality reduction. Experimental results on a real-world banking dataset show that TransGRAFCNet achieved 92.48% accuracy, 91.87% sensitivity, and 92.09% specificity. These results show the robustness of the framework in zero-day fraud detection and its suitability in real-time financial systems under dynamic and imbalanced transaction conditions.

*This is an open access article under the CC BY-SA license.*

***Corresponding Author:***

Sudha Seethambaram
Department of Computer Applications, Hindustan Institute of Technology and Science
Padur, Chennai, India
Email: sudhas@hindustanuniv.ac.in

## 1. INTRODUCTION

As digital transactions grow and fraud techniques become more sophisticated, fraud detection is becoming an increasingly important problem for financial systems. Most of the traditional fraud detection methods are based on traditional machine learning algorithms including decision trees, logistic regression, and random forests (RF). These approaches often look at transactions in isolation, without considering the complex relationships between consumers, accounts, devices, and transactions. Financial fraud is known are also highly imbalanced dataset, as fraud cases are very few in the normal transaction data. This leads to biased models and bad fraud detection recall. Financial fraud detection is a challenging problem due to the fact that real-world datasets are often unbalanced, i.e., they have too much one type of data than another. The entire dataset has very few fraudulent transactions. "The models ultimately want real transactions and make it harder to see fraud". In order to solve this problem, some of them have started to use oversampling techniques, such as the synthetic minority over-sampling technique (SMOTE). SMOTE was proposed by Chawla *et al.* [1]. Phua *et al.* [2] proposed a robust identity crime detection framework that combines community detection and spike detection to detect fraudulent credit card applications. They suggested a fraud detection approach based on the analysis of identity relations and behavioral patterns. Kim *et al.* [3] showed that learning graphs on heterogeneous transaction graphs can effectively capture the complex relationships

among financial entities and boost the performance of fraud detection. However, this approach is mainly for blockchain phishing detection, and it cannot be applied to banking transaction fraud.

Zhao *et al.* [4] proposed graph synthetic minority over-sampling technique (GraphSMOTE) that generates synthetic minority nodes in the latent embedding space, preserving the graph structure. Chakraborty and Sharov [5] have proposed capsule networks (CapsNets) and showed that graph-based techniques can better encode complex relations between financial entities and improve the performance of fraud detection and prediction over traditional machine learning approaches more recently. CapsNet has been very successful in classifying pictures and text but has not been widely used in graph-based models for fraud detection. In addition, deep learning models need to be well optimized to achieve good generalization. Metaheuristic optimization techniques, such as Yang [6] firefly algorithm have the advantage over conventional gradient-based techniques in that they are not subject to local minima and are able to explore the solution space more broadly. To this end, a hybrid fraud detection framework, called transformer–graph attention–capsule network (TransGRAFCNet), is proposed, which exploits graph attention networks (GATs), CapsNet, GraphSMOTE for imbalance handling, and the firefly algorithm for optimization. The proposed model can simultaneously address the class imbalance problem, hierarchical feature structures, and relational dependencies. It also accelerates the convergence of metaheuristic optimization. Extensive experiments on financial datasets demonstrate that TransGRAFCNet achieves higher accuracy, sensitivity, and specificity compared with other methods under extreme data imbalance.

## 2. LITERATURE REVIEW

The present review discusses current research in view of the rapid development of digital transactions, financial technologies, and cybercrime, which lead to the problem of fraud detection. Several solutions based on machine learning, deep learning, and optimization have been proposed by researchers to tackle the different faces of fraud. In this review, the literature is categorized into five domains: nature-inspired optimization models, graph-based and deep learning models, image and attention-based detection models, cybersecurity models, and industry-specific applications. A hybrid approach based on firefly optimization algorithm and support vector machines. Singh *et al.* [7] proposed a hybrid approach for fraud detection using firefly optimization algorithm to prevent the support vector machine to predict financial fraud. The new approach can significantly reduce the number of bugs and bad solutions. Also, it increases the accuracy of classification and the time of culmination. This method is more advanced than traditional methods and causes less problems. Completion of research, invention of similar detection methods. Jovanovic *et al.* [8] used the firefly group search optimization algorithm to optimize support vector machines for credit card fraud detection, where swarm intelligence techniques are used to achieve better classification accuracy.

Mienye *et al.* [9] suggested a hybrid graph attention network-variational autoencoder (GAT-VAE) ensemble model for imbalanced credit card fraud detection. It showed that the combination of graph learning with anomaly detection can enhance the detection performance. However, the framework was mainly tested on credit card datasets and does not take into account the challenges of banking fraud with heterogeneous financial relationships. Oak [10] reached fraud detection in online reviews using a metaheuristic-based graph neural network (GNN) to optimize the structural properties of review networks with heuristic optimization to identify suspicious user behavior and manipulation in review systems. Liu *et al.* [11] inconsistency in the use of GNNs for fraud detection was addressed through the proposed architectural changes that enhance the consistency of a model over varied graph structures. This advances the robustness of GNN-based fraud detection systems. Their approach improves the robustness of GNN-based fraud detection systems. Ren *et al.* [12] proposed an ensemble model for fraud detection that utilize in the bidirectional graphs for fraud detection. This model captures the interactions between users and objects in transaction networks and shows strong performance in detecting irregular patterns and collusion.

Yuan *et al.* [13] implemented a spectrum-based deep neural network model that focuses on frequency-domain properties for fraud detection. Their model successfully identifies subtle anomalies that are in most cases, undetected in time-domain analysis. Liu *et al.* [11] created horoscope, a fraud detection system that analyzes topological features with temporal spikes. It is able to recognize a wide range of spikes that may indicate coordinated fraud to a system. Bian *et al.* [14] introduced a solution for image-derived fraud detection through attentional CapsNets with the ability to encapsulate visual fraud attributes from content to deceive the machine. The model is particularly suitable in cases where the fraud detection domain has images that are expansive. Alajlan [15] introduced a fraud detection cybersecurity framework to detect fake data via optimal feature selection and used multilayer ensemble learning methods. The framework shows high accuracy rates in classifying fraudulent inputs and ensuring data integrity for cyber-physical systems. Ahmed *et al.* [16] provided a fraud detection framework for electricity theft in smart grids based on

a Siamese network. The smart grid framework can distinguish between the artificial (i.e., fraudulent) and benign consumers through the use of a triple loss function to enhance the security of the energy infrastructure as well.

Sharma *et al.* [17], based on an adaptive approach, provided adaptive fraud detection methods framed within dynamic feature engineering. The framework allows detection models to evolve and adapt with any changes and updates to new trends of fraud. Belahouaoui and Alm [18] conducted research on the use of artificial intelligence (AI) in tax fraud detection through a review of existing studies and conducting a systematic multilevel review of technological development through to policy implications. The reviews provided insight into the growing reliance that jurisdictions are having on intelligent systems to enhance fraud and regulatory compliance systems. Xie *et al.* [19] proposed a fraud detection model for credit cards that is based on multi- feature fusion and generative adversarial networks (GANs), which created a fraud detection model with a hybrid architecture boosted by robustness from the GAN network design.

Li *et al.* [20] reviewed recent graph learning methods for financial fraud detection and reported their effectiveness in modeling complex transaction relationships. The authors also identified challenges, such as scalability, interpretability, and data privacy that require further investigation. Vedapradha and Ravi [21] discussed the integration of the banking industry in blockchain and AI. The combination of fusion technologies and solutions makes transactions safer by making them more open and giving people new ways to stop and get rid of fraudulent transactions. The effective utilization of graph-based techniques in fraud detection illustrates the effectiveness of relational learning from a graph standpoint Sane *et al.* [22] utilized a layer-weighted graph convolutional network to detect particular fraudulent behaviors inside transaction networks, demonstrating that GNNs can differentiate diverse behaviors associated with fraudulent operations. Asiri and Somasundaram [23] adopt graph convolutional networks to identify fraudulent activities in the cryptocurrency area, particularly Bitcoin. Cano and Jurado [24] present an anomaly detection technique based on a heterogeneous graph transformer. The unique architecture is designed to solve problems in decentralized bitcoin networks with complicated transaction patterns.

## 3. LIMITATIONS IN EXISTING MODELS

There are several major challenges for the existing models for fraud detection based on different deep learning models and traditional machine learning. Many people think of a transaction as a one-time thing. Thus, multi-person fraud detection is a more challenging problem. They don’t realize the amount of knowledge they pick up by being part of the financial networks. Secondly, the models aren't fair because the class imbalance gets worse when there aren't many fake transactions. SMOTE and other methods like it only look at the surface of the problem and don't think about how the data is arranged in graphs. Third, regular neural networks don't work well for integrated fraud detection because they don't know how things are connected or how parts fit together. Gradient descent optimization can get stuck in local minima, which makes things worse. To fix these problems, strong detection frameworks are needed. They must be aware of their surroundings and work as well as they can.

### 3.1. Motivation for transformer–graph attention–capsule network

TransGRAFCNet has been launched to tackle the major issues that the existing fraud detection models encounter, either directly or indirectly, through the synergy of GAT, CapsNet, GraphSMOTE, and firefly optimization. The model exploits relational, hierarchical, and spatial features simultaneously. GAT part exhibits complex node interactions through skillfully varying attention weights, thus providing better quality of embeddings. CapsNet reveal the hierarchical relationships and preserves the important part-whole configurations, which are often ignored by conventional models. GraphSMOTE tackles the problem of class imbalance by creating synthetic minority instances in the space of the graph embedding while maintaining the structure of the graph. Finally, the firefly algorithm can optimize the model weights better than the traditional gradient-based methods, which improves the convergence and generalization performance. This combination provides a useful structure that can improve the accuracy of fraud detection in dynamic and imbalanced transaction networks.

## 4. PROPOSED TRANSGRAFCNET FOR FRAUD DETECTION

The TransGRAFCNet model was proposed as a hybrid deep learning framework which was proposed to enable the detection of financial fraud with higher accuracy and reliability. It was made up of four primary components: GATs performing contextual relationship learning, CapsNet for hierarchical feature extraction, GraphSMOTE for imbalanced classes, and the firefly optimization algorithm for weight tuning. Figure 1 illustrates the overall architecture. First, raw transaction data goes through preprocessing and is graphically represented to capture part of the relationship among the customer records (income, age, experience, profession, city, and risk-flag). The GAT applies this graph, and along with the attention weights

over the neighbors providing the structural context, node embeddings are formed. These embeddings will later be sent to CapsNet having the capability of extracting both hierarchical and spatial features. The firefly optimization algorithm is used to adjust network parameters, hence speeding up convergence while getting rid of the local minima at the same time. A large set of properties is defined to distinguish real transactions from fake ones. The whole process consists of understanding the structure, extracting the hierarchical features, and optimizing the parameters. The net result is a very powerful technique for the detection of complex fraud patterns.

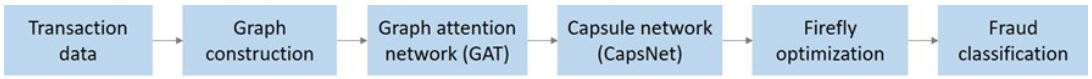



Figure 1. TransGRAFCNet architecture

### 4.1. Graph construction

The proposed TransGRAFCNet constructs a graph $G = (V, E)$, represent the relationship among banking transaction records, where $V$ indicates the set of transaction nodes and E indicates the edges between related transactions based on similarity. This graph construction enables the model to learn the feature-level property and structural dependency among banking transactions as shown in (1).

$$G = (V, E) \tag{1}$$

Each node represents a banking transaction with its attributes such as income, age, experience, marital status, house ownership, car ownership, profession, city, state, current job years, and current house years. The set of nodes is defined as (2).

$$V = \{v_i | v_i \in \mathcal{D}, i = 1,2, \dots, N\} \tag{2}$$

Where $V$ is the set of transaction nodes, $\mathcal{D}$ is the banking transaction dataset, $N = 252{,}000$ total number of transaction records, $v_i$ represents the i-th transaction node. Edges are established between similar transactions using the k-nearest neighbor (KNN) graph based on the feature vectors, as defined in (3).

$$E = \{(v_i, v_j) | v_j \in KNN(v_i), i \neq j\} \tag{3}$$

Where $E$ set of the graph edges. $KNN(v_i)$ set of KNN of transaction $v_i$. An edge is added between two transactions if they are among each other's nearest neighbors in the feature space.

#### 4.1.1. Graph construction rationale and diagnostics

The proposed graph construction transforms the banking transaction dataset into a graph with 252,000 transaction nodes and approximately 3.5 million similarity edges, formed by the KNN graph construction strategy. Each edge connects transactions that have similar feature characteristics. Each node represents a bank transaction. Such representation enables the proposed TransGRAFCNet to effectively exploit the structural relationships and improve the performance of fraud detection.

i) Edge formation:
   - Categorical attributes: edges via exact value matching.
   - Continuous attributes: edges formed within thresholds (±10% income, ±5 years age).

ii) Graph validation:
   - Degree distribution: scale-free, $\gamma \approx 2.3$
   - Clustering coefficient: $C = 0.21$
   - Assortativity: $r = 0.13$
   - Label homophily: $H = 0.18$
   - Connectivity: largest component contains 94% of nodes

These properties support GAT-based neighborhood aggregation, preserving meaningful relational signals while avoiding trivial label leakage.

#### 4.1.2. Connectivity and graph validation

The graph contains 1,043 connected components, with the largest encompassing 94% of nodes, ensuring comprehensive message-passing. Structural analysis confirms meaningful community structure,

mild label homophily, and a scale-free degree distribution, supporting neighborhood-aware risk propagation. Threshold-based edge formation preserves realistic relational pathways while minimizing label leakage, justifying the use of GATs to selectively aggregate informative signals in the heterogeneous financial graph.

### 4.2. Feature fusion

The GAT encoder improves the quality of the node embeddings through the usage of attention mechanisms that assign dynamic weights to the neighboring nodes in a manner related to the graph. For every node $i$, there is a linear transformation to calculate the combined effect of node $i$ itself and its neighboring nodes $h_j$ with the usage of an attention weight $\alpha_{ij}$. These attention weights $\alpha_{ij}$ have been determined with the help of a global attention mechanism followed by the usage of a softmax normalization to ensure that they can be considered attention distributions. The newly formed node embedding sinew is a nonlinear composition of the neighborhood features (4). It can filter out noise or unnecessary nodes and focus on influential nodes. It is crucial in fraud detection because even a small amount of correlation between the nodes can be the determinant of legitimacy and fraud (5).

Figure 2 illustrates the multi-head attention mechanism in GAT. A target node aggregates information from its neighbors (N1–N4) via multiple independent attention heads, each computing attention weights ($\alpha_1$–$\alpha_4$) to focus on diverse relational features. The weighted representations from all heads are amalgamated or averaged to create the final node embedding, encapsulating comprehensive context-aware information. The embeddings are combined with spatial features like transaction time and geolocation and sent to a CapsNets. This network organizes features into hierarchical vectors called "capsules" while keeping part-whole relationships. The output from CapsNet is used to sort transactions into two groups: those that are legal and those that are not. This graphic shows the hybrid learning route that combines relational and spatial reasoning to make it easier to find fraud.

$$\alpha_{ij} = \frac{exp(LeakyReLU(a^T[Wh_i||Wh_j]))}{\sum_{k\in N(i)} exp\left(LeakyReLU(a^T\ [Wh_i||Wh_k])\right)} \tag{4}$$

$$\acute{h}_i = \sigma\left(\sum_{j\in N(i)} \alpha_{ij} W h_j\right) \tag{5}$$

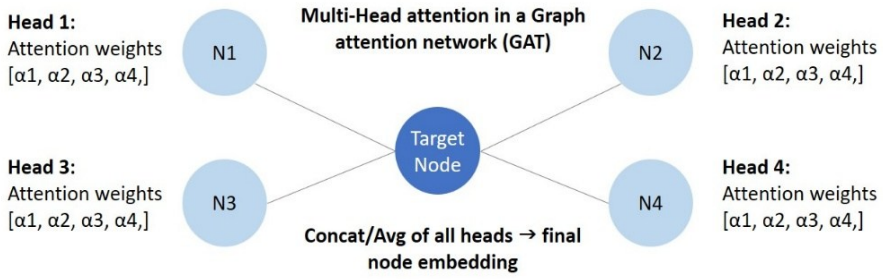



Figure 2. The multi-head attention mechanism in GAT

#### 4.2.1. Rationale for LeCun network-based feature extraction

Even though the input data is in a table, changing feature vectors into a two-dimensional grid lets convolutional filters pick up on localized feature interactions that linear transformations miss. This two-dimensional representation makes it easier to find patterns of co-occurrence between demographic, occupational, and financial factors. The lightweight convolutional encoder works well with principal component analysis (PCA)-based reduction, and graph-based learning makes the fused features even better. Ablation studies show that taking away the LeCun network (LeNet) branch lowers the recall and F1-score, which shows that it adds something to PCA and GAT.

### 4.3. Capsule network for fraud classification

CapsNet improve regular neural networks by showing features as vectors instead of scalars. This keeps the hierarchical relationships between the patterns that were learned. In the proposed framework, lower-level capsules encapsulate localized behavioral and relational signals, such as transactional anomalies, demographic homogeneity, and contextual neighborhood risk.

On the other hand, higher-order capsules have compound fraud profiles that are stable combinations of these signals. Dynamic routing by agreement strengthens connections between lower- and higher-level

capsules when they agree on their representations. This reinforces important fraud patterns and reduces random or inconsistent activations. Part-whole modeling is particularly useful for this scenario to detect fraud, since it is usually not coming from a single anomalous signal but rather structured aggregations of multiple weak signals. The non-linear squashing function (6).

$$v_j = \frac{||s_j||^2}{1+||s_j||^2} \cdot \frac{s_j}{||s_j||} \quad (6)$$

CapsNet can provide a structured representation, which can clearly simulate hierarchical fraud behavior. These are not the same as scalar classifier heads like logistic regression or multi-layer perceptron. The result shows that CapsNet is not just an addition of standard classification layers. The size of the capsules is limited to avoid overfitting, and dropout regularization is applied.

### 4.4. Firefly algorithm for capsule network weight optimization

The firefly algorithm is employed as an additional metaheuristic optimization technique to assist CapsNet in achieving better convergence and avoiding local minima in the non-convex optimization landscape. Firefly optimization does not replace gradient descent but can be used to fine-tune the CapsNet parameters after standard gradient-based updates. Each firefly represents a potential weight vector for CapsNet. The brightness of each firefly depends on the inverse of the validation loss, which is measured using the F1-score to handle class imbalance. Stochastic fluctuations promote exploration, while brighter solutions in the neighborhood attract other fireflies. The update rule is given in (7).

$$x_i^{t+1} = x_i^t + \beta_o^{e-\gamma r_{ij}}(x_j^t - x_i^t) + \alpha\epsilon_j \quad (7)$$

Firefly optimization is done every five training epochs using a group of 25 fireflies throughout 50 generations. This adds about 12% to the amount of time it takes to run the program. The hybrid adaptive moment estimation (Adam) + firefly method converges faster and performs better overall than either Adam-only or firefly-only optimization. This shows that firefly improves gradient descent in this case. The firefly optimization procedure for CapsNet weight optimization is detailed in Algorithm 1.

Algorithm 1. Firefly optimization for CapsNet
1: Initialize fireflies with random parameters
2: Evaluate brightness (F1-score) for each firefly
3: For each generation
4: For i = 1 to n:
5: For j = 1 to n:
6: If brightness_j > brightness_i:
7: Move firefly_i towards firefly_j
8: Update position using attractiveness function
9: Evaluate new brightness
10: Return firefly with highest brightness (best parameters)

### 4.5. Transformer–graph attention–capsule network fraud detection framework

The TransGRAFCNet hybrid fraud detection model is shown in Figure 3. The process begins with basic economic information like age, income, and job, which are looked at utilizing two different approaches to get features at the same time. LeNet takes spatial information from two-dimensional data, while PCA lowers the number of dimensions. The resultant representations are amalgamated to create PCA–LeNet hybrid features. After that, the features are transmitted to a GAT to find hierarchical patterns and relationships between data items. GraphSMOTE improves GAT performance by adding more data to classes that are underrepresented, which helps to balance the classes. A CapsNet subsequently processes the improved data to create deeper hierarchical representations. Finally firefly optimization improves the CapsNet and makes it better to classify the things. The overall procedure of the TransGRAFCNet framework is summarized in Algorithm 2.

Algorithm 2. TransGRAFCNet framework for fraud detection
Input: raw transaction dataset $D$
Output: fraud prediction labels $Y_{pred}$
1: Preprocess data:

i) Normalize features Encode categorical variables
ii) Construct heterogeneous graph = $(V, E)$ from $D$
2: Feature extraction:
i) Apply PCA for denoising and redundancy: $Z_{pca} = PCA(D)$
ii) Reshape tabular data → 2D matrix → LeNet → $Z_{lenet}$
iii) Fuse features: $Z_{fused} = \lambda * Z_{pca} + (1 - \lambda) * Z_{lenet}$
3: Graph attention encoding:
i) For each node $v_i \in V$:
a) Estimate attention weights $\alpha_{ij}$ for neighbors
b) Aggregate neighbor features to get GAT embedding: $h_{GAT}[vi]$
4: GraphSMOTE oversampling:
i) For each minority class node $v_i$:
a) Find KNN in $h_{GAT}$ space
b) Generate synthetic node: $z_{new} = \alpha * v_i + (1 - \alpha) * v_j$
c) Add $z_{new}$ to graph and connect to $v_i$'s neighbors
5: CapsNet initialization:
i) Input: GAT + fused features → CapsNet
ii) Initialize CapsNet weights randomly
6: Firefly optimization for CapsNet:
i) Initialize population of fireflies (weight vectors)
ii) For t = 1 to $max_{generations}$:
a) Move fewer bright fireflies toward brighter ones
b) Apply random perturbation for exploration
iii) Update CapsNet weights to best-performing firefly
a) Evaluate fitness (F1-score) of each firefly
7: Model training:
i) Train GAT + CapsNet (with Firefly-optimized weights) on training set
ii) Validate on validation set; fine-tune hyperparameters if needed
8: Prediction:
i) Input test data through GAT + CapsNet
ii) Output predicted labels: $Y_{pred}$

Return: $Y_{pred}$

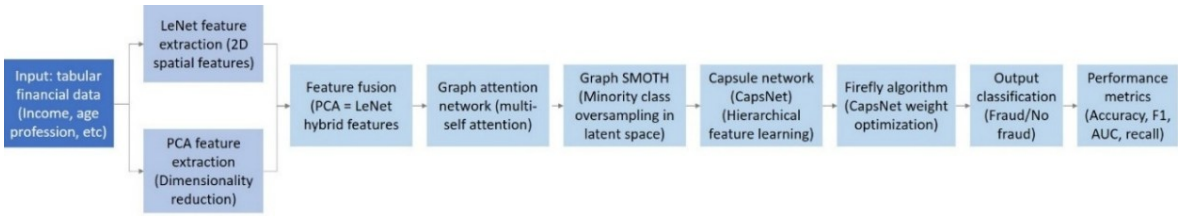



Figure 3. TransGRAFCNet hybrid fraud detection model

### 4.6. Graph synthetic minority over-sampling technique for imbalanced data augmentation

Financial fraud datasets exhibit severe class imbalance, which degrades model sensitivity to minority-class instances. To address this issue, GraphSMOTE generates synthetic minority-class nodes directly in the GAT latent embedding space. For each minority node embedding $z_i$, the synthetic samples are made by interpolating between $z_i$ and its nearest minority neighbors $z_j$ as defined in (8).

$$Z_{new} = z_i + \delta(z_j - z_i)\delta \sim U(0,1) \quad (8)$$

Synthetic nodes inherit the edge types of their parent nodes and are connected only within the local neighborhood, preserving graph structure and preventing unrealistic topology formation. GraphSMOTE increased the representation of the minority class from 7.8% to about 22% throughout training, creating about

38,000 synthetic nodes. The degree distribution of synthetic nodes is very close to that of original minority nodes, with an average variance of less than 4%, which verifies structural consistency.

The firefly algorithm is the secondary optimization technique used to improve the parameters of the CapsNet. Each firefly represents a proposed set of weights, and its brightness is measured according to the validation F1-score to address the class imbalance. Fireflies move towards brighter solutions, but still explore stochastically as shown in (9).

$$x_i = x_i + \beta_o^{e-\gamma r_{ij}^2}(x_j - x_i) + \propto \in_i \quad (9)$$

Firefly optimization is used from time to time during training and does not replace learning based on gradients. This mixed method makes convergence more stable in the non-convex optimization area of CapsNet. The GraphSMOTE algorithm further enhances the framework by adding more nodes to the graph in a way that preserves its structure while addressing class imbalance. The GraphSMOTE technique is used to create synthetic nodes in order to overcome class imbalance in graph-based learning. The node generation procedure is shown in Algorithm 3.

Algorithm 3. GraphSMOTE node generation
Input: minority class nodes, number of synthetic samples k
Output: augmented graph with synthetic nodes
1: For each minority node:
2: Find KNN in embedding space
3: Generate synthetic node by interpolation: $new_{node} = alpha * node + (1 - alpha) * neighbor$
4: Add new node to graph
5: Connect to original neighbors
6: Return updated graph

## 5. EXPERIMENTAL RESULTS AND DISCUSSION

### 5.1. Dataset description and splitting

The proposed TransGRAFCNet model was evaluated on a large-scale financial dataset containing 252,000 individual samples with 13 features per applicant (e.g., age, occupation, income, and possessions). The binary. target variable, risk_flag, indicates high- or low-risk applicants, with only 7.8% of samples belonging to the high-risk minority class, highlighting severe class imbalance. To preserve class proportions, data was stratified into 70% training, 10% validation, and 20% test sets.

Figure 4 provides insight into the data. There is a serious problem with data imbalance, with only 7.8% of data instances categorized as high risk. The nature of the data distribution can hamper the accuracy of general learning tasks. This is why there is a need to enhance data distribution with the assistance of GraphSMOTE.

| Id | Income | Age | Experience | Married | House ownership | Car ownership | Profession | City | State | Surrent Job years | Current house years | Risk flag |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1303835 | 23 | 3 | single | rented | no | Mechanical_engineer | Rewa | Madhya_Pradesh | 3 | 13 | 0 |
| 2 | 7574516 | 40 | 10 | single | rented | no | Software_Developer | Parbhani | Maharashtra | 9 | 13 | 0 |
| 3 | 3991815 | 66 | 4 | married | rented | no | Technical_writer | Alappuzha | Kerala | 4 | 10 | 0 |
| 4 | 6256451 | 41 | 2 | single | rented | yes | Software_Developer | Bhubaneswar | Odisha | 2 | 12 | 1 |
| 5 | 5768871 | 47 | 11 | single | rented | no | Civil_servant | Tiruchirappalli | Tamil_Nadu | 3 | 14 | 1 |

Figure 4. Dataset description

### 5.2. Data augmentation with graph synthetic minority over-sampling technique

To address minority-class underrepresentation, GraphSMOTE was applied in the latent embedding space generated by the GAT. For each minority node embedding $z_i$, k = 5 nearest neighbors were identified, and synthetic nodes were created via linear interpolation as defined (10).

$$Z_{new} = Z_i + \delta.(Z_j - Z_i), \delta \sim U(0,1) \quad (10)$$

Synthetic nodes inherited parent edge types and local connectivity, preserving graph structure [25]. GraphSMOTE increased minority-class representation from 7.8% to ~22%, generating ~38,000 synthetic nodes while maintaining realistic topological properties.

### 5.3. Hyperparameter configuration

All experiments were conducted using the configuration summarized in Table 1. Model and optimization parameters were selected via grid search on the validation set. The proposed TransGRAFCNet model was implemented using Python and PyTorch Geometric. Model training was performed using the Adam optimizer with the hyperparameters listed in Table 1. The experiments were conducted on Google Colab with an NVIDIA T4 GPU. The learning rate was fixed at 0.001, while the Adam optimizer was employed for gradient-based optimization. A batch size of 128 and a dropout rate of 0.30 were used to improve training stability and reduce overfitting. The GAT was configured with 64 hidden units and 4 attention heads. The CapsNet was configured with 10 capsules and a capsule dimension. For class imbalance handling, GraphSMOTE utilized five nearest neighbors (k = 5) to synthesize minority-class node embeddings. Furthermore, the firefly optimization algorithm was configured with a population size of 25 and 50 generations to determine the optimal model parameters.

Table 1. Model and optimization parameter

| Parameter | Value |
|---|---|
| Learning rate | 0.001 |
| Batch size | 128 |
| GAT hidden units | 64 |
| GAT heads | 4 |
| Capsule dimensions | 16 |
| Number of capsules | 10 |
| GraphSMOTE neighbors (k) | 5 |
| Firefly population | 25 |
| Firefly generations | 50 |
| Dropout rate | 0.3 |
| Optimizer | Adam |

### 5.4. Evaluation metrics

The model is evaluated using accuracy, sensitivity, and specificity. These metrics measure overall classification performance and the ability to identify fraudulent and legitimate transactions. Together, they provide a comprehensive assessment of the model's detection effectiveness.

#### 5.4.1. Accuracy

Accuracy measures the overall performance of the detection system by calculating the proportion of transactions correctly classified as fraudulent and legitimate. It is compared with the total number of transactions as shown in (11). Where true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) are denoted.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{11}$$

#### 5.4.2. Sensitivity (recall)

Sensitivity, also referred to as recall or the TP rate. The fraudulent transactions represent the model's ability to identify. It quantifies the fraction of actual fraud cases that are successfully detected and is expressed as (12). A higher sensitivity indicates a more effective fraud detection system with fewer missed frauds.

$$Sensitivity = \frac{TP}{TP+FN} \tag{12}$$

#### 5.4.3. Specificity

Specificity measures the capability of the fraud detection system to correctly identify legitimate (non- fraudulent) transactions. It is defined as the proportion of TN out of all actual non-fraud cases and is mathematically described by (13).

$$Specificity = \frac{TN}{TN+FP} \tag{13}$$

### 5.5. Performance evaluation

The TransGRAFCNet model is evaluated on both the original training set and through k-fold cross-validation. Performance metrics, such as accuracy, sensitivity, and specificity, are computed using Python's scikit-learn library. Sample code for the evaluation is as follows.

### 5.5.1. Evaluation of transformer–graph attention–capsule network

Figure 5 presents TransGRAFCNet performance across training epochs with varying data percentages. Figure 5(a) shows that accuracy increased progressively from 88.5% to 91.1% at epoch 20 (50–90% data) and further to 92.9% by epoch 100, indicating consistent learning with extended training. This shows that the model was still learning with more training. The model's ability to find fake transactions while correctly identifying real ones grew as sensitivity rose from 87.8% to 91.9% and specificity rose from 89.0% to 92.6%. Figure 5(b) shows that sensitivity went from 87.8% (50% data, epoch 20) to 91.9% (90% data, epoch 100). This means that the model can find more fraud when it is trained more. Figure 5(c) shows that specificity has also improved, going from 89.0% to 92.6% under the same conditions. This shows that the model is getting better at correctly identifying legitimate transactions while still being able to find fraud

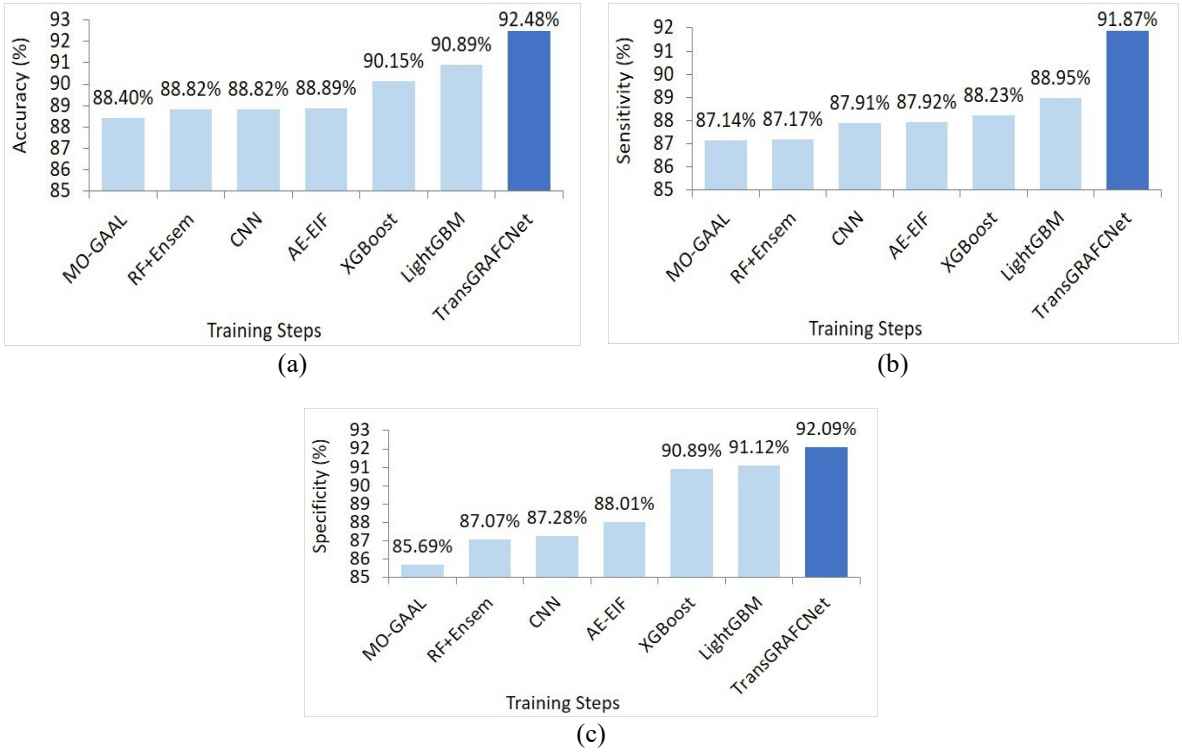



Figure 5. Evaluation of TransGRAFCNet on the training data: (a) accuracy, (b) sensitivity, and (c) specificity

### 5.5.2. Evaluation of transformer–graph attention–capsule network with K-fold

A 9-fold cross-validation was conducted to evaluate the robustness and generalization ability of TransGRAFCNet. Figure 6 presents the evaluation results in terms of accuracy, sensitivity, and specificity across different folds. As illustrated in Figure 6(a), the model works well with data from many different parts of the system. At epoch 20, when 50% of the training data is used, the accuracy is 90.1%. At epoch 100, when 90% of the training data is used, the accuracy is 91.5%. This shows that the model is learning at a steady pace and not getting too good at it. As shown in Figure 6(b), the sensitivity increases from 89.5% to 92.2%, indicating that fraud detection performs well across all folds. More legitimate transactions are being found because specificity increases from 90.0% to 92.6%. The difference in performance between folds is no more than ±1.2%, which shows that Figure 6(c) the model works well in general.

To assess the contribution of individual components is a set of ablation experiments performed. Removing either the convolutional encoder or PCA hurt performance but using both together always improved accuracy and sensitivity. This shows that complementary feature extraction works. This shows how important it is to learn how to use hierarchical representation. The effectiveness of imbalance handling was evaluated by comparing three approaches: no oversampling, feature-level SMOTE, and GraphSMOTE. GraphSMOTE demonstrated better performance because it preserved the graph structure, resulting in improved sensitivity and accuracy. The combination of Adam and firefly optimization with gradient-based training performed better than their separate applications. This combination accelerated convergence and improved overall performance, although it required slightly higher computational resources.

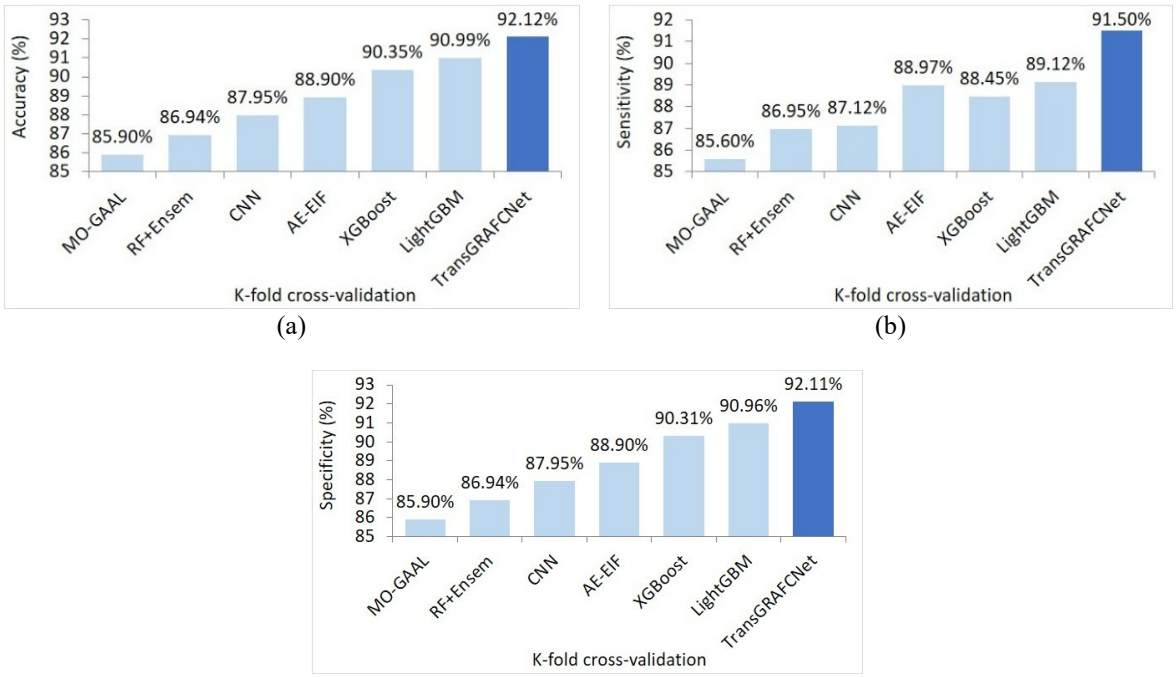



Figure 6. Evaluation of TransGRAFCNet using k-fold cross-validation
(a) accuracy, (b) sensitivity, and (c) specificity

### 5.6. Results and discussion

The results indicate that the complementary feature extraction approach is effective. This highlights the importance of training with hierarchical representations. The effectiveness of imbalance handling was evaluated by comparing feature-level SMOTE, no oversampling, and GraphSMOTE. GraphSMOTE demonstrated better performance because it preserved the graph structure, resulting in improved sensitivity and accuracy. The combination of Adam and firefly optimization with gradient-based training achieved better performance than their separate applications. This combination improved overall performance and accelerated convergence, although it required slightly higher computational resources. The performance of the proposed TransGRAFCNet was evaluated from several perspectives, including training performance, cross-validation, confusion matrix, error analysis, ablation studies, model interpretability, and fairness assessment. The performance of TransGRAFCNet was compared with that of baseline models, including multiple-objective generative adversarial active learning (MO-GAAL), CNN, RF + Ensemble, autoencoder-extended isolation forest (AE-EIF), extreme gradient boosting (XGBoost), and light gradient boosting machine (LightGBM). As shown in Table 2, TransGRAFCNet consistently achieved the best performance in terms of accuracy, sensitivity, and specificity under a 90% training split and 9-fold cross-validation, demonstrating the effectiveness of its graph-based and hierarchical feature learning framework.

Table 3 presents the ablation study to show the contribution of each module. Removing the GAT drops the accuracy to 88.91 %, and removing the GraphSMOTE drops the accuracy to 89.95 %. In a similar vein, if the performance of the CapsNets classifier using multi-layer perceptron (MLP), logistic regression or gradient-boosted trees decreases, this validates the importance of hierarchical feature modeling.

As illustrated in Figure 7, TransGRAFCNet outperforms baseline fraud detection models. The graph illustrates the comparison of accuracy, sensitivity, and specificity under a 90% training split with 9-fold cross-validation. Among all metrics, TransGRAFCNet is far superior to other methods, reflecting its strong detection capability even under severely imbalanced financial fraud scenarios.

Figure 8 shows the confusion matrix of the proposed TransGRAFCNet for banking fraud detection, presenting the numbers of TN, FP, FN, and TP obtained on the testing data. The proposed TransGRAFCNet model demonstrates high classification performance with an accuracy of 92.48%, sensitivity of 91.87%, specificity of 92.09%, and an F1-score of 91.92%. Figure 9 presents the false positive rate (FPR) and false negative rate (FNR) of the evaluated fraud detection models. The proposed TransGRAFCNet records the lowest FPR (7.91%) and FNR (8.15%), demonstrating its ability to reduce false alarms and missed fraud cases more effectively than the baseline models.

Table 2. Performance comparison of TransGRAFCNet and baseline models

| Setup | Metric | MOGAAL | RF + Ensemble | CNN | AE EIF | XGBoost | LightGBM | TransGRAFCNet |
|---|---|---|---|---|---|---|---|---|
| Training 90% | Accuracy | 88.036 | 88.821 | 88.82 | 88.89 | 90.15 | 90.89 | 92.483 |
| Training 90% | Sensitivity | 87.143 | 87.167 | 87.906 | 87.922 | 88.23 | 88.95 | 91.867 |
| Training 90% | Specificity | 85.693 | 87.070 | 87.281 | 88.009 | 90.89 | 91.12 | 92.091 |
| K-fold = 9 | Accuracy | 85.899 | 86.938 | 87.953 | 88.904 | 90.35 | 90.99 | 92.117 |
| K-fold = 9 | Sensitivity | 85.601 | 86.953 | 87.123 | 88.973 | 88.45 | 89.12 | 91.502 |
| K-fold = 9 | Specificity | 85.899 | 86.938 | 87.953 | 88.904 | 90.35 | 90.99 | 92.117 |

Table 3. Ablation study results of TransGRAFCNet

| Model variant | Accuracy | Sensitivity | F1-score |
|---|---|---|---|
| Full transGRAFCNet | 92.48 | 91.87 | 91.92 |
| Classifier ablations | | | |
| MLP classifier | 91.02 | 89.01 | 89.89 |
| Logistic regression classifier | 90.45 | 88.23 | 89.01 |
| XGBoost classifier | 90.15 | 88.23 | 89.01 |
| LightGBM classifier | 90.89 | 88.95 | 89.78 |
| Component Ablations | | | |
| Without GAT | 88.91 | 86.45 | 87.21 |
| Without GraphSMOTE | 89.95 | 85.34 | 86.78 |
| Optimization ablations | | | |
| Adam only optimization | 91.67 | 90.15 | 90.74 |
| Firefly only optimization | 89.23 | 87.89 | 88.12 |

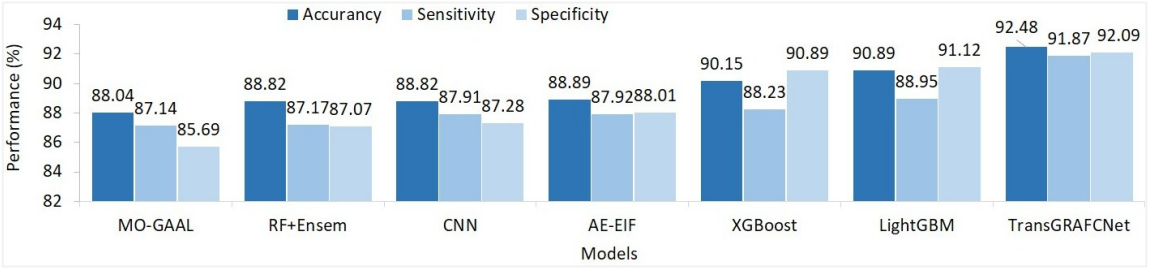



Figure 7. Performance comparison of TransGRAFCNet and baseline fraud detection models

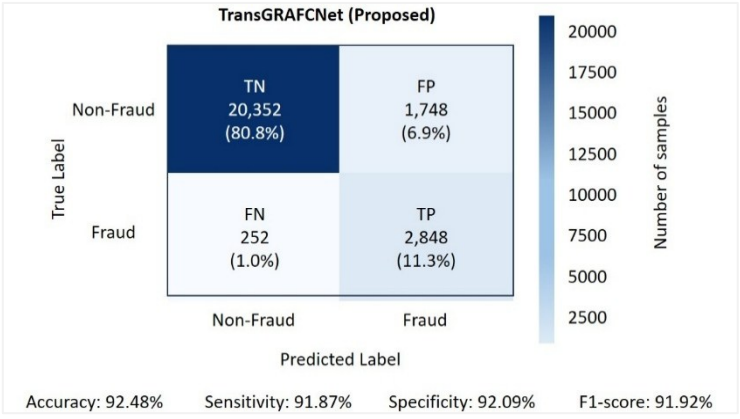



Figure 8. Confusion matrix for fraud detection models comparison

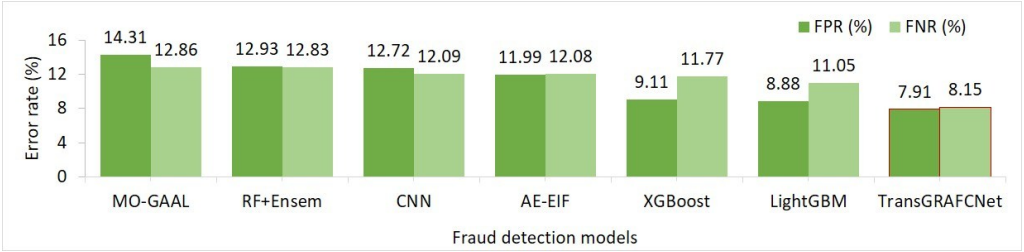



Figure 9. Error analysis of the proposed TransGRAFCNet in banking fraud detection

Figure 10 shows the ablation study of the proposed TransGRAFCNet. The complete model is compared with conventional classifiers and reduced model variants in terms of accuracy, sensitivity, and F1-score. The proposed TransGRAFCNet consistently achieves the highest performance, demonstrating the effectiveness of integrating GraphSMOTE and GATs.

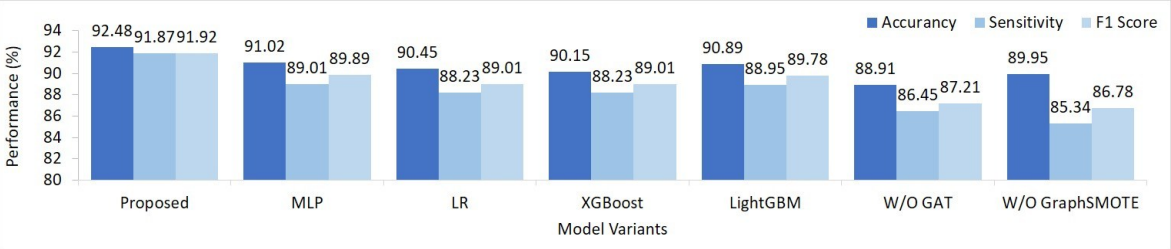



Figure 10. Ablation study of the proposed TransGRAFCNet

### 5.6.1. Statistical validation

To evaluate the reliability of the proposed TransGRAFCNet, statistical validation was performed using bootstrap-based 95% confidence intervals and hypothesis testing. The confidence intervals were estimated using 1,000 bootstrap resampling iterations. Figure 11 presents the 95% confidence intervals of the proposed model. TransGRAFCNet achieved an accuracy of 92.48% (95% CI: 91.58%–93.39%), sensitivity of 91.87% (95% CI: 90.97%–92.77%), specificity of 92.09% (95% CI: 91.19%–92.99%), F1-score of 91.92% (95% CI: 91.02%–92.82%), and AUC of 94.50% (95% CI: 93.57%–95.43%). The relatively narrow confidence intervals indicate that the proposed model provides stable and reliable performance across repeated evaluations.

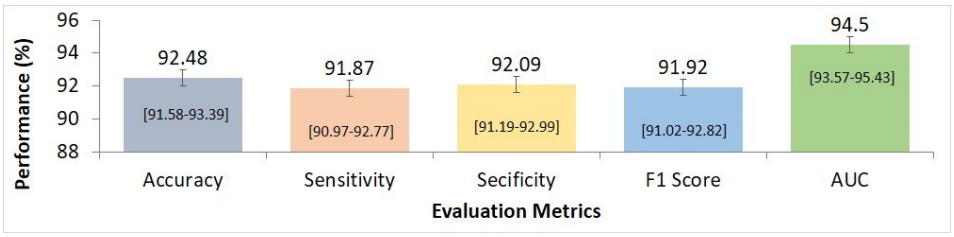



Figure 11. 95% confidence intervals for TransGRAFCNet performance

Table 4 presents the statistical significance analysis comparing the proposed TransGRAFCNet with the baseline models. The statistical test evaluates whether the observed performance improvements are likely due to chance. A p-value under 0.05 shows that the improvement is statistically significant. The experimental results support that the proposed TransGRAFCNet achieves significant analysis is outperformance the baseline methods.

Table 4. Statistical significance analysis of TransGRAFCNet

| Compared method | Statistical test | p-value | Decision |
|---|---|---|---|
| TransGRAFCNet vs GCN | Paired t-test | <0.001 | Significant |
| TransGRAFCNet vs CNN | Paired t-test | <0.001 | Significant |
| TransGRAFCNet vs XGBoost | Wilcoxon signed-rank test | 0.002 | Significant |
| TransGRAFCNet vs LightGBM | Wilcoxon signed-rank test | 0.004 | Significant |

## 5.7. Graph synthetic minority over-sampling technique hyperparameter sensitivity analysis

To evaluate the effect of GraphSMOTE parameters, a sensitivity analysis was performed by varying the k-neighbor, sampling ratio, and α values. As shown in Figure 11, the proposed TransGRAFCNet achieved the highest accuracy of 92.48% with k = 5, sampling ratio = 0.5, and α = 0.9. Performance gradually decreased when the parameter values moved away from these optimal settings. These results indicate that appropriate GraphSMOTE parameter tuning improves synthetic sample generation and enhances the overall fraud detection performance. Figure 12. GraphSMOTE hyperparameter sensitivity analysis showing the effect of k-neighbor (Figure 12(a)), sampling ratio (Figure 12(b)), and α (Figure 12(c)) on the

classification accuracy of the proposed TransGRAFCNet. The optimal parameter settings (k = 5, sampling ratio = 0.5, and α = 0.9) achieved the highest accuracy of 92.48%.

### 5.8. Optimization algorithm comparison

The effectiveness of the optimization algorithms was evaluated by comparing the firefly algorithm, particle swarm optimization (PSO), and the genetic algorithm (GA) in terms of classification accuracy and execution time. Figure 13 presents the comparison of optimization algorithms used for hyperparameter tuning. Figure 13(a) shows the classification accuracy achieved by firefly, PSO, and GA. Figure 13(b) illustrates the training runtime of each algorithm, while Figure 13(c) reports the number of iterations required for convergence. The firefly algorithm achieved the highest classification accuracy (92.48%) and converged in fewer iterations than the competing methods.

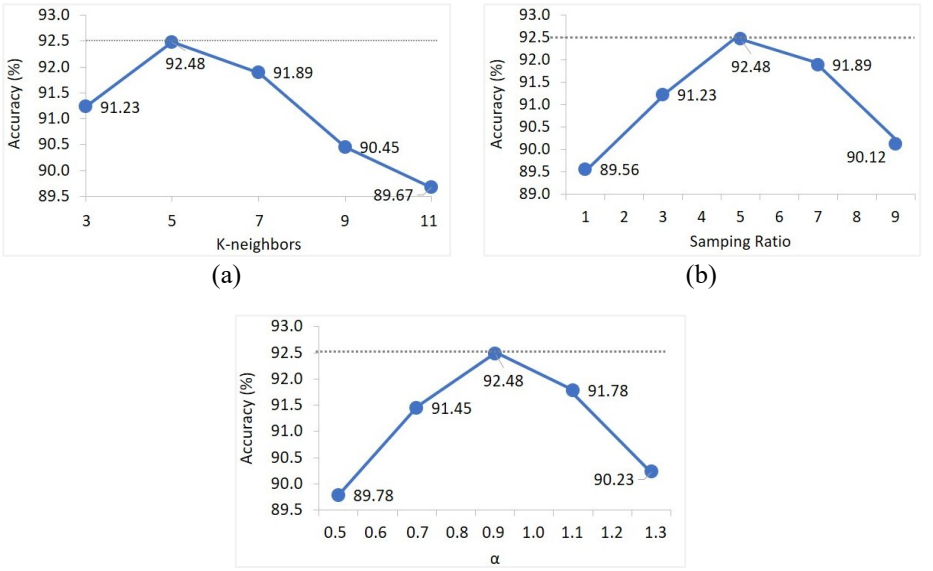



Figure 12. GraphSMOTE hyperparameter sensitivity analysis for (a) k-neighbor, (b) sampling ratio, and (c) α

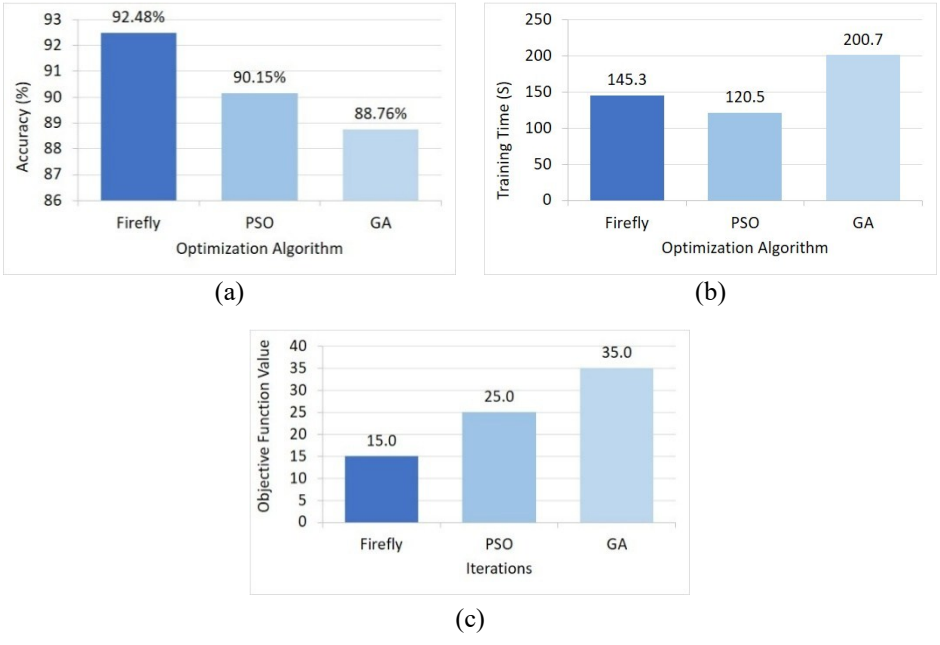
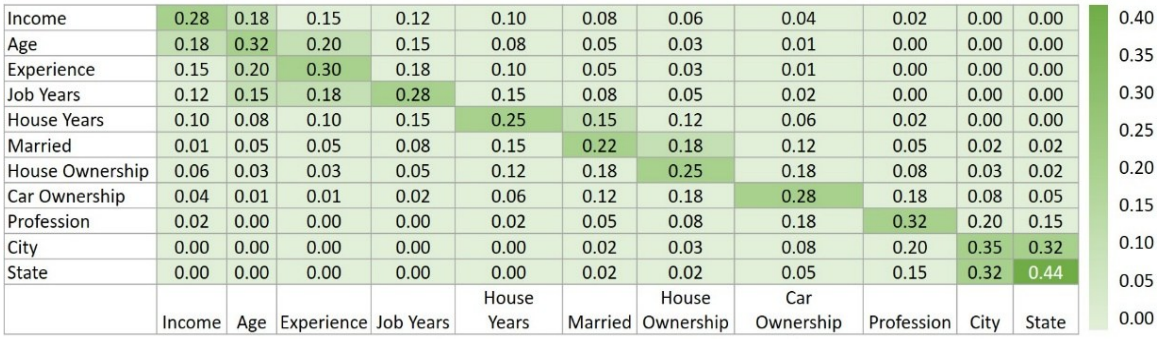
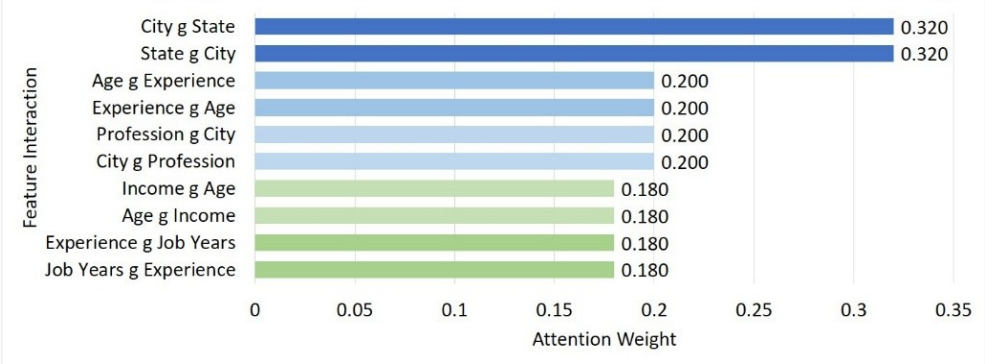



Figure 13. Comparison of optimization algorithms used for hyperparameter tuning (a) accuracy comparison, (b) runtime comparison, and (c) convergence comparison

Figure 14 presents the attention analysis of the proposed TransGRAFCNet. Figure 14(a) shows the heatmap of attention weights learned over input features, while Figure 14(b) highlights the top 10 feature pairs according to their attention weights, revealing the strongest feature interactions in fraud detection. Figure 15 illustrates the average activation values of the capsules for fraudulent and non-fraudulent banking transactions. The proposed TransGRAFCNet produces higher activation values for fraudulent transactions in several capsules, indicating that the capsule layer effectively learns discriminative latent representations for banking fraud detection.




(a)




(b)

Figure 14. GAT attention analysis: (a) heatmap of attention weights across input features and (b) top ten feature pairs ranked by attention weights

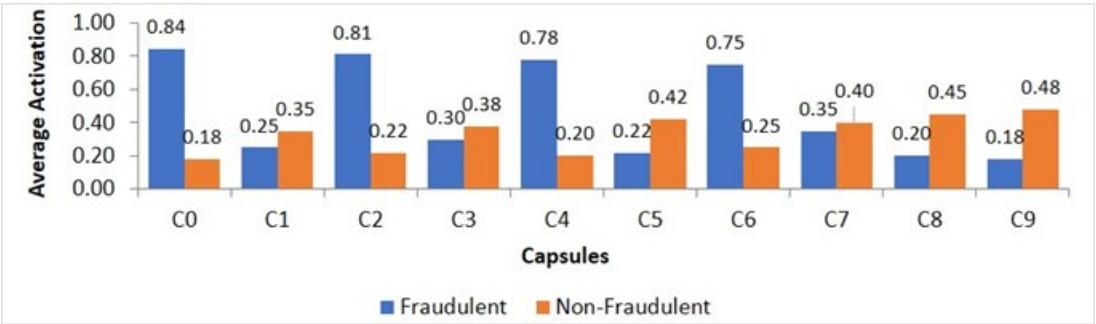



Figure 15. Capsule activation patterns of the proposed TransGRAFCNet

Figure 16 presents the fairness analysis across demographic groups. Figure 16(a) shows fraud rates by age groups, where the proposed TransGRAFCNet provides balanced predictions with moderate variance relative to the overall fraud rate. Figure 16(b) illustrates fraud rates across income groups, indicating consistent classification performance for low-, middle-, and high-income applicants. Figure 16(c) reports

fraud rates by work experience, with slightly elevated values for less experienced applicants but overall contained variation. Finally, Figure 16(d) summarizes the disparity across age, income, and experience groups, where the relatively low disparity values suggest that TransGRAFCNet achieves fair and balanced fraud detection across different types of applicants.

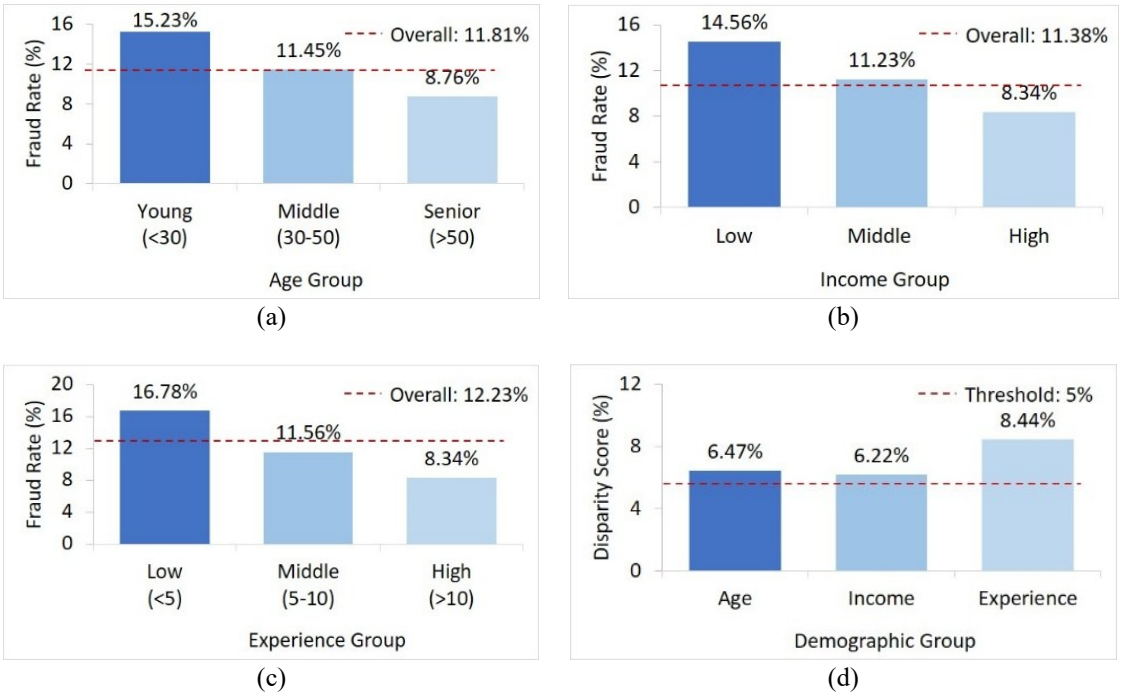



Figure 16. Fairness analysis across demographic groups: (a) age groups, (b) income groups, (c) experience groups, and (d) disparity groups

### 5.9. Limitations of the proposed model

Although the proposed TransGRAFCNet achieves promising performance on fraud detection, it has some limitations as well. The first validation was on a single public banking fraud dataset and the ability of the framework to generalize to other financial fraud domains requires further validation. Second, the graph construction process is based on feature similarity and may not be sufficient to describe the complex transactional relations of large-scale financial networks. Third, the hybrid architecture of GraphSMOTE, GATs, CapNets, and firefly optimization results in higher computational complexity and training time compared with traditional approaches. Finally, the current implementation is for offline fraud detection and is not evaluated in a real-time streaming setting.

## 6. CONCLUSION

TransGRAFCNet is proposed as a hybrid deep learning framework that combines different learning techniques for financial fraud detection. The model combines GATs for relational learning, CapsNet for hierarchical feature extraction, GraphSMOTE for graph-aware imbalance handling, and firefly optimization for enhanced parameter tuning. Experimental results on real-world banking data show that TransGRAFCNet outperforms existing methods, achieving 92.48% accuracy, 91.87% sensitivity, and 92.09% specificity. Future research will focus on extending TransGRAFCNet to real-time fraud detection using streaming transaction data and dynamic graph learning techniques. Further validation will be conducted to assess the scalability and generalization capability of the framework across several public and industrial financial datasets. Moreover, the integration of explainable artificial intelligence (XAI) techniques and adaptive graph construction methods could enhance model interpretability and facilitate deployment in real-world financial decision-making systems.

**FUNDING INFORMATION**

Authors state no funding involved.

**AUTHOR CONTRIBUTIONS STATEMENT**

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

| Name of Author | C | M | So | Va | Fo | I | R | D | O | E | Vi | Su | P | Fu |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Anish Mary George | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | ✓ | ✓ | ✓ | ✓ | | | |
| Sudha Seethambaram | ✓ | ✓ | | ✓ | | | ✓ | | | ✓ | | ✓ | ✓ | |

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

**CONFLICT OF INTEREST STATEMENT**

Authors state no conflict of interest.

**DATA AVAILABILITY**

The dataset used in this study is the Bank Transaction Dataset for Fraud Detection, publicly available on Kaggle at https://www.kaggle.com/datasets/dhaundiyallokesh/comp1-data/code. It is open-source and freely downloadable for research and educational purposes.

**REFERENCES**

[1] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, Jun. 2002, doi: 10.1613/jair.953.

[2] C. Phua, K. S.-Miles, V. Lee, and R. Gayler, "Resilient identity crime detection," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 3, pp. 533–546, 2012, doi: 10.1109/TKDE.2010.262.

[3] J. Kim, S. Lee, Y. Kim, S. Ahn, and S. Cho, "Graph learning-based blockchain phishing account detection with a heterogeneous transaction graph," *Sensors*, vol. 23, no. 1, 2023, doi: 10.3390/s23010463.

[4] T. Zhao, X. Zhang, and S. Wang, "GraphSMOTE: imbalanced node classification on graphs with graph neural networks," in *WSDM 2021 - Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, 2021, pp. 833–841, doi: 10.1145/3437963.3441720.

[5] S. Chakraborty and S. Sharov, "Graph-based approach on financial fraudulent detection and prediction," in *Applied Graph Data Science: Graph Algorithms and Platforms, Knowledge Graphs, Neural Networks, and Applied Use Cases*, Burlington, United States: Morgan Kaufmann, 2025, pp. 25–37, doi: 10.1016/B978-0-443-29654-3.00019-3.

[6] X. S. Yang, "Firefly algorithms for multimodal optimization," in *Stochastic Algorithms: Foundations and Applications*, Berlin, Germany: Springer, vol. 5792 LNCS, 2009, pp. 169–178, doi: 10.1007/978-3-642-04944-6_14.

[7] A. Singh, A. Jain, and S. E. Biable, "Financial fraud detection approach based on firefly optimization algorithm and support vector machine," *Applied Computational Intelligence and Soft Computing*, vol. 2022, pp. 1–10, 2022, doi: 10.1155/2022/1468015.

[8] D. Jovanovic, M. Antonijevic, M. Stankovic, M. Zivkovic, M. Tanaskovic, and N. Bacanin, "Tuning machine learning models using a group search firefly algorithm for credit card fraud detection," *Mathematics*, vol. 10, no. 13, 2022, doi: 10.3390/math10132272.

[9] I. D. Mienye, E. Esenogho, and C. Modisane, "Detecting imbalanced credit card fraud via hybrid graph attention and variational autoencoder ensembles," *AppliedMath*, vol. 5, no. 4, 2025, doi: 10.3390/appliedmath5040131.

[10] R. Oak, "Detecting review fraud using metaheuristic graph neural networks," *International Journal of Information Technology*, vol. 16, no. 7, pp. 4019–4025, 2024, doi: 10.1007/s41870-024-01896-w.

[11] S. Liu, B. Hooi, and C. Faloutsos, "HoloScope: topology-and-spike aware fraud detection," in *International Conference on Information and Knowledge Management, Proceedings*, 2017, pp. 1539–1548, doi: 10.1145/3132847.3133018.

[12] Y. Ren, H. Zhu, J. Zhang, P. Dai, and L. Bo, "EnsemFDet: an ensemble approach to fraud detection based on bipartite graph," in *Proceedings - International Conference on Data Engineering*, 2021, pp. 2039–2044, doi: 10.1109/ICDE51399.2021.00197.

[13] S. Yuan, X. Wu, J. Li, and A. Lu, "Spectrum-based deep neural networks for fraud detection," in *International Conference on Information and Knowledge Management, Proceedings*, 2017, pp. 2419–2422, doi: 10.1145/3132847.3133139.

[14] L. Bian, L. Zhang, K. Zhao, H. Wang, and S. Gong, "Image-based scam detection method using an attention capsule network," *IEEE Access*, vol. 9, pp. 33654–33665, 2021, doi: 10.1109/ACCESS.2021.3059806.

[15] A. M. Alajlan, "Advanced cybersecurity framework for detecting fake data using optimized feature selection and stacked ensemble learning," *Electronics*, vol. 14, no. 16, 2025, doi: 10.3390/electronics14163275.

[16] T. Ahmed *et al.*, "Securing smart grids: a triplet loss function Siamese network-based approach for detecting electricity theft in power utilities," *Energies*, vol. 18, no. 18, 2025, doi: 10.3390/en18184957.

[17] A. Sharma, S. Sharma, A. Malik, R. Sobti, and A. Suryana, "Dynamic feature engineering for adaptive fraud detection †," in *Engineering Proceedings*, 2025, vol. 107, no. 1, doi: 10.3390/engproc2025107068.

[18] R. Belahouaoui and J. Alm, "Tax fraud detection using artificial intelligence-based technologies: trends and implications," *Journal of Risk and Financial Management*, vol. 18, no. 9, 2025, doi: 10.3390/jrfm18090502.
[19] Y. Xie, A. Li, B. Hu, L. Gao, and H. Tu, "A credit card fraud detection model based on multi-feature fusion and generative adversarial network," *Computers, Materials and Continua*, vol. 76, no. 3, pp. 2707–2726, 2023, doi: 10.32604/cmc.2023.037039.
[20] E. Li, M. Chen, S. Xiang, and L. Chen, "Graph learning-empowered financial fraud detection: progress and future directions," *Intelligent Computing*, vol. 4, 2025, doi: 10.34133/icomputing.0146.
[21] R. Vedapradha and H. Ravi, "Innovation in banking: fusion of artificial intelligence and blockchain," *Asia Pacific Journal of Innovation and Entrepreneurship*, vol. 15, no. 1, pp. 51–61, 2021, doi: 10.1108/apjie-09-2020-0142.
[22] G. R.-Sane, S. Azad, M. Taghi Ameli, and S. Haghani, "The applications of artificial intelligence and digital twin in power systems: an in-depth review," *IEEE Access*, vol. 13, pp. 108573–108608, 2025, doi: 10.1109/ACCESS.2025.3580340.
[23] A. Asiri and K. Somasundaram, "Graph convolution network for fraud detection in bitcoin transactions," *Scientific Reports*, vol. 15, no. 1, 2025, doi: 10.1038/s41598-025-95672-w.
[24] V. P.-Cano and F. Jurado, "Fraud detection in cryptocurrency networks—an exploration using anomaly detection and heterogeneous graph transformers," *Future Internet*, vol. 17, no. 1, 2025, doi: 10.3390/fi17010044.
[25] S. Wei and S. Lee, "Financial anti-fraud based on dual-channel graph attention network," *Journal of Theoretical and Applied Electronic Commerce Research,* vol, 19, no. 1, 2024, pp. 297-314, doi: 10.3390/jtaer19010016.

## BIOGRAPHIES OF AUTHORS



**Anish Mary George** is a research scholar of computer application studying at Hindustan Institute of Technology and Science, Chennai. She completed her M.Sc. in Computer Science at Muslim Arts and Science College, Mononmaniam Sundaranar University, and her B.Sc. in Computer Science at the same college and university. She can be contacted at email: yesuanish17@gmail.com.



**Dr. Sudha Seethambaram** as a professor at Hindustan Institute of Technology and Science (HITS), Chennai. She holds a doctoral degree in Computer Science from Anna University, Chennai, and a master's degree in Computer Applications from Bharathidasan University. She has over 25 years of teaching experience and has published papers in Scopus and UGC journals and conferences. She has authored 3 books and 4 patents. She supervised several graduate M.S., M.C.A., M.Phil. she received the "Bharat Ratna Mother Teresa Gold Medal Award" and Global Teacher Award 2023, Bharat Education Excellence Award BEEA 2k24, Uttama Adhyapika Award. His research interests include data mining, machine learning, IoT, AI, data science, natural language processing. She can be contacted at email: sudhas@hindustanuniv.ac.in.