JOURNAL OF INTERNATIONAL CRISIS AND RISK COMMUNICATION RESEARCH
ISSN: 2576-0017
2026, VOL 9, NO 1

Financial Fraud Identification Using Graph Neural Network And LSTM With Autoencoder-Based Data Refinement

Sriharsha Anand Pushkala

Director, Fraud Strategy and Analytics Atlanticus 512 Mountain Crossing, Woodstock, Georgia, USA – 30188 sanandpushkala@gmail.com

Abstract

Credit card fraud is a major issue globally, with a loss of millions of dollars each year. Due to the evolution of different fraudsters and high data volume, detecting fraudulent credit card transactions is a challenging task. Due to the availability of transaction data with high imbalance, developing an efficient deep learning approach-based fraud transaction detection system is still challenging. Also, extracting significant descriptors from the transaction data determines the performance of the classifier. Therefore, the paper proposes a generative adversarial network (GAN) and an auto-encoder (AE) based transaction data generation system to train the proposed fraud transaction detection approach. The fraud transaction detection system was constructed using a graph neural network (GNN) and long short-term memory (LSTM) that extracts significant descriptors from the transaction data to distinguish fraud and non-fraud transactions. The performance of the proposed fraud detection system was evaluated using the European cardholder 2013 (ECH-2013) dataset and the PaySim dataset with scales such as F1-score, recall, precision, and accuracy. The proposed hybrid GNN-LSTM approach yields an accuracy of 97.91% and 98.09% when evaluated using the ECH-2013 and PaySim datasets, respectively.

Keywords: Fraud detection, Generative adversarial network, Auto-encoder, Graph neural network, Long-short-term memory.

1. Introduction

Due to the development of mobile devices, in-app payments, and Internet of Things (IoT) devices, credit card-based transactions [1] are dominant in digital commerce. The increase in the usage of such credit cards also increases the fraudulent transactions. Fraudsters mimic the behaviour of legitimate users by producing counterfeit cards to perform fraudulent activities. Due to fraudulent activities, significant losses are caused to banks and cardholders, which is globally estimated to be more than 5.127 trillion dollars [2] each year. To prevent fraudulent transactions, the payment provider should use a fraudulent transaction identification system that blocks such activities without affecting legitimate transactions. Such security measures involve improving execution systems, enhancing the front-end system, and improving the middleware ecosystem. Recently, due to the development of deep learning and artificial intelligence (AI) techniques, several payment providers have deployed deep learning and machine learning-based fraud transaction identification systems [3] to enhance security.

The AI-based credit card fraud identification system also aims to reduce declined transactions and to improve the approval ratio. Due to the evolution of new fraudsters' tactics, it is difficult to follow their pattern of transactions. Also, the deep learning-based credit card fraud detection approaches rely on high-quality, balanced data to train the model [4]. However, the number of fraudulent transactions will be much lower than the legitimate transactions to effectively train the AI models. To address these challenges, several authors proposed different approaches to detect credit card fraud [5]. The authors Zioriris et al. [6] used a multistage network to detect the fraudulent transaction. This approach selects

the descriptor utilizing dual auto-encoders. Further, a deep convolutional neural network (CNN) is utilized to differentiate the non-fraud and fraud transactions with the use of significant features selected by the dual auto-encoders.

A hybrid scheme is proposed [7] for making decisions during the transaction. Two models, namely a recurrent neural network (RNN) and an LSTM, are used in this approach. To minimize the effect of the highly imbalanced dataset in model training, the authors utilized an oversampling process on the limited fraudulent transaction data. For the selection of significant descriptors, the authors Ileberi et al. [8] used a genetic algorithm. Different machine learning models such as Naïve Bayes, artificial neural networks, logistic regression, random forest and decision tree are then utilized to detect the fraudulent transaction that uses the selected significant descriptors. The authors utilized the European cardholder transactions to generate the dataset for training the model, such as ensemble bagging and ensemble independent models. These two models are used to construct an integrated multistage classifier [9]. The author addressed the data imbalance problem with the use of processes such as random undersampling, cluster centroids and instant hardness thresholding.

The authors Rehman et al. used a federated learning with a hybrid Fuzzy approach [10]. The authors used various attack models such as bad-mouthing attacks, whitewashing, etc. The authors report that the use of both the Fuzzy logic concept and federated learning yields a trust score value of 0.93. The fraud in a Bitcoin network is detected using blockchain and machine learning [11]. Two models, namely random forest and XGboost, are utilized to detect the abnormal transaction patterns. To evaluate the robustness of the approach against vulnerabilities and attacks, an attacker model is introduced in this approach. Four properties, namely trajectory behaviours, system behaviours, traffic behaviour, and transaction behaviours, are utilized in online payment fraud detection [12]. This approach integrates property and event levels by using coarse and fine-grained information between the behavioural events. The authors Baabdullah [13] used blockchain technology and a federated learning approach to detect credit card fraud. Three architectures, namely LSTM, CNN, and random forest, are utilised in the detection process. To address the data imbalance problem, the synthetic minority over-sampling technique (SMOTE) is used.

A GAN network [14] was used to detect the fraudulent transactions, which improves the discriminating strength with the use of modules like the generator and discriminator. The authors report that the GAN-based scheme results in better AUC when evaluated using European 2013 data. Both the feature information and network information are considered by Huang et al. [15] for identifying financial fraud during transactions. Both the real-world data and synthetic data are considered for evaluation by the authors. To minimize the data imbalance problem, GAN [16] was utilised to generate minority class cases. This generated class case was combined with the minority class cases present in the dataset and was used as training data for the models. The use of GAN-based up-sampling improves the performance of the classifier to detect the fraud. An attention process was included in the LSTM-RNN [17] where the vectors constructed from the sequence of information are used to detect fraud by the LSTM-RNN model.

CNN was used to detect the fraud in online transactions [18], where this approach constructs convolutional patterns using the row transaction information. The CNN network uses a feature sequencing layer, four pooling layers following four convolutional layers, and finally, a fully connected layer. An optimized gradient boosting approach [19] was used to detect credit card fraud transactions, in which the parameters are tuned using a Bayesian process. A dual-stage approach, including an auto-encoder and a supervised learning algorithm [20], is utilized to detect the fraudulent transactions. The author reports that the usage of deep auto-encoders improves the performance over other classifiers. A hybrid resampling approach [21] was used along with the ensemble classifier for improving the fraud detection performance. The hybrid resampling approach uses a modified nearest neighbour approach along with the SMOTE up-sampling process. Also, the ensemble classifier utilizes adaptive boosting and an LSTM network.

To improve the fraud detection performance due to non-stationary patterns, a fuzzy logistic regression [22] approach was proposed. This approach yields a Matthew's correlation coefficient and

sensitivity higher than 0.8 and 0.9, respectively. Different deep learning models are combined to derive a hybrid approach [23] for detecting fraud. The approach uses RNN and GAN, in which blocks like gated recurrent unit, LSTM, and RNN are utilized to construct the GAN discriminator. Though this approach provides reasonable performance, the complexity of this approach is higher. To detect relevant descriptors to differentiate non-fraud and fraud transactions, Cherif et al. [24] proposed a graph neural network-based approach to obtain complex patterns in transactions. The graph neural network also uses a graph converter to improve the classification performance. To reduce the internal covariance shift and to speed up the convergence, a normalization process was used, which is derived from the batch normalization process. The reinforcement learning and graph neural network were combined [25] to derive a context-aware adaptive approach. In this approach, the dynamic graph is constructed with the use of transactions and users (edges are transactions and nodes are users). This approach simultaneously collects semantic descriptors, temporal, and spatial patterns, which reduces the false positive cases and provides an F1-score of 97.3%.

The GAN-based approaches that are discussed above use the transaction data generated by the generator network to train the model. However, there is no guarantee that there exists a fuzziness between the fraud and non-fraud classes on the generated transaction data. Therefore, the proposed approach also uses an autoencoder to ensure that the transaction data generated by the GAN model belongs to the fraud class type. The use of an autoencoder avoids the usage of fuzzy non-fraud transaction types as fraud transaction types, which improves the performance of the fraud detection system. The usage of a graph neural network in the traditional scheme does not collect temporal patterns from the transaction data. Therefore, the proposed approach also uses an LSTM network to collect both temporal patterns and significant descriptors in categorising the transaction type as fraud / non-fraud. Thus, the paper has the following contributions

(i) The paper proposes a GAN and autoencoder-based oversampling approach to generate a sufficient number of fraud transactions for training the fraud detection approach. The use of an autoencoder minimizes the usage of fuzzy non-fraud data as fraud data in training the fraud transaction detection system.
(ii) The paper proposes a graph neural network and LSTM (GNN-LSTM) based fraud detection approach that effectively utilizes the significant transaction pattern and temporal descriptors in classifying the transaction type. A convolutional network is further utilized before the LSTM network for extracting significant patterns.
(iii) An ensemble averaging layer is used to obtain the actual predicted probability from the predicted probability obtained from the GNN and LSTM networks.
(iv) Finally, the evaluation of the proposed GNN-LSTM approach was evaluated using classification scales such as precision, F1-score, accuracy, and recall with the use of the ECH-2013 and PaySim datasets.

The rest of the manuscript is constructed as below. Section 2 elaborates on the proposed hybrid GAN and autoencoder approach to minimize class imbalance in training the fraud detection system. This section also elaborates on the hybrid fraud detection system that uses both GNN and LSTM structures. Section 3 discusses the experimental results conducted on the suggested credit card fraud transaction detection approach, and Section 4 concludes the manuscript with its key findings.

2. Proposed Methodology

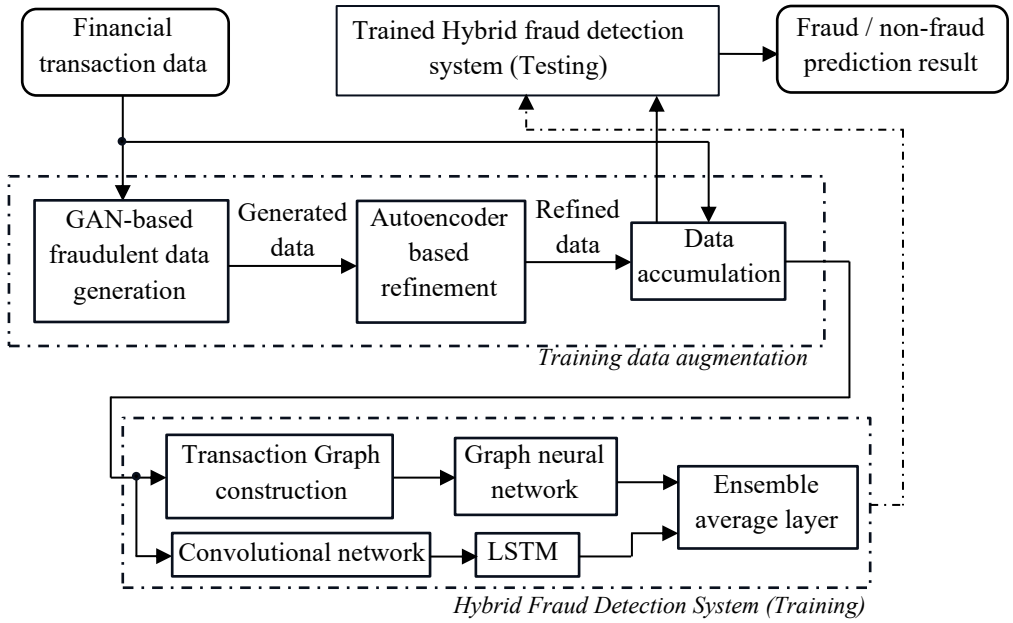

Fig. 1: Framework of proposed financial fraud detection system

The framework of the suggested financial fraud detection approach is illustrated in Fig. 1. The suggested approach has two major modules, namely (i) GAN-autoencoder-based data augmentation approach and (ii) GNN-LSTM-based fraud detection system. The data required to train the proposed GNN-LSTM approach is generated using the GAN-autoencoder network. The autoencoder network is utilized to refine the transaction data generated by the GAN network.

(i) GAN-autoencoder for fraud data augmentation

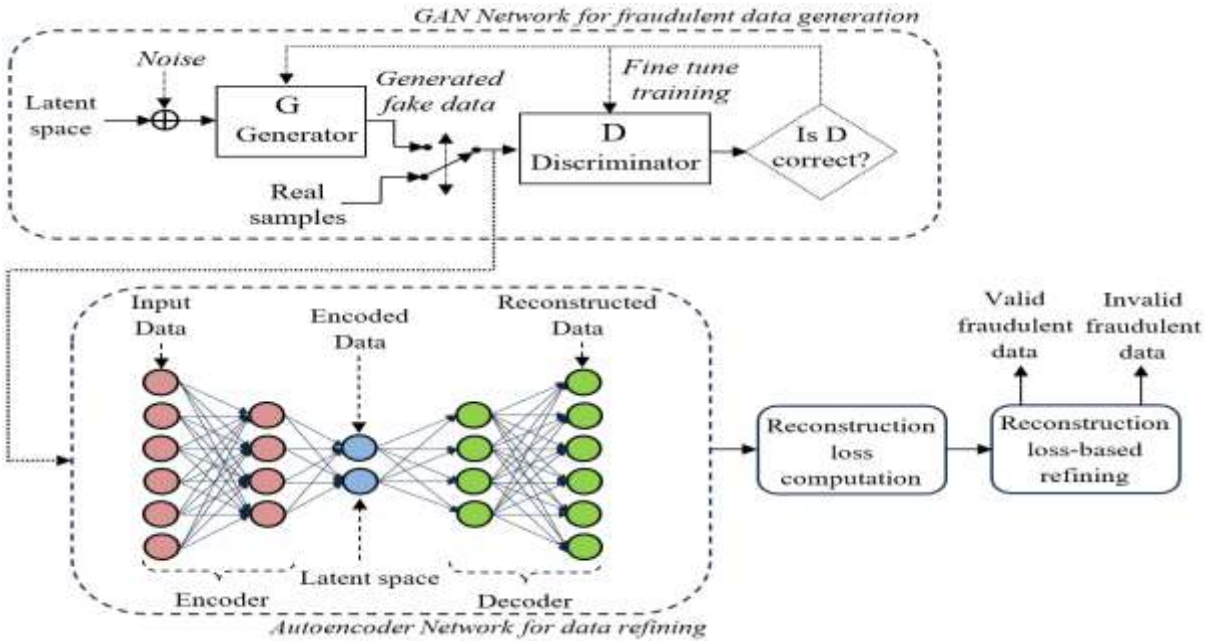

Fig. 2: Structure of proposed GAN-autoencoder-based fraudulent data refinement

The structure of the GAN-autoencoder-based synthetic data generation and refining process is illustrated in Fig. 2. The GAN is used to generate the synthetic transaction data, while the autoencoder uses the reconstruction loss to refine the fraudulent transaction data.

(a). GAN for fraudulent data augmentation

The generative adversarial network (GAN) [26] contains two sub-networks, namely the discriminator (D) that follows the generator (G). During the training process, the generator generates fake data, and the discriminator tries to correctly detect the fake data. Thus, a generator of the trained GAN is used to generate fraudulent synthetic data to balance the data of the imbalanced dataset.

Let ε resemble a random noise vector. Thus, the generator uses ε to produce a fraudulent synthetic transaction represented as, $G(\varepsilon; p_g)$. Here p_g resembles the distribution parameter of the generator. Similarly, the discriminator function is represented as, $D(o_d, p_d)$. Here o_d represent the input for the discriminator and p_d resemble the distribution parameter of the discriminator. The generator G learn to create realistic fake data to fool the discriminator network D whose loss function can be represented as

$$\eta_g = E_{\varepsilon \sim q_\varepsilon}\left[\log\log \frac{1}{D(G(\varepsilon))}\right] \quad (1)$$

Where, q_ε resembles the latent noise distribution and E resemble the mathematical expectation operator. The discriminator D, learn to identify fraud and real data generated by the generator, in which the loss function of the discriminator can be represented as,

$$\eta_d = E_{\varepsilon \sim q_\varepsilon}\left[\log\log \frac{1}{1-D(G(z))}\right] + E_{r \sim q_r}\left[\log\log \frac{1}{(D(d))}\right] \quad (2)$$

Here, q_r resembles the distribution of true data (real fraud) and r resembles the actual data. The GAN model was trained with Adam optimizer, number of epochs as 300, and learning rate of 0.0005. The batch size and noise vector size used are 128 and 32, respectively. The generator and discriminator of the GAN use the LeakyReLU activations. In the testing phase, the GAN generate fraudulent transaction data, which is refined using the autoencoder.

(b) Autoencoder (AE) for data refining

The auto-encoder [27] has two sub-networks, namely the encoder and decoder. The auto-encoder was trained only using the non-fraud transaction data. Let $b^{(n)}$ resembles the non-fraud data; therefore, the encoder compresses the data $b^{(n)}$ to a lower dimension. Let the function performed by the encoder be represented as $e(\cdot)$. Therefore, the encoder results in output

$$\hat{b}^{(n)} = e\left(b^{(n)}\right) = \sigma\left(w_e b^{(n)} + \mu_e\right) \quad (3)$$

Where, w_e and μ_e resembles the weight and bias of the encoder function, and σ resembles the sigmoid function. The decoder tries to reconstruct the data $b^{(n)}$ from the encoder output $\hat{b}^{(n)}$. Let $d(\cdot)$ resembles the decoder function. Thus, the output of the decoder is represented as,

$$\tilde{b}^{(n)} = d\left(\hat{b}^{(n)}\right) = \sigma\left(w_d \hat{b}^{(n)} + \mu_d\right) \quad (4)$$

Where w_d and μ_d resembles the weights and biases of the decoder. During the training process, the auto-encoder tries to minimize the reconstruction loss

$$\eta_r\left(b^{(n)}, \tilde{b}^{(n)}\right) = \|b^{(n)} - \tilde{b}^{(n)}\| \quad (5)$$

Here $\|\cdot\|$ resembles the Euclidean distance. The autoencoder uses two layers in the encoder and decoder sections with ReLU activations. The model was trained with Adam optimizer at a learning rate of 0.001 and a batch size of 64. In the testing phase, the reconstruction loss will be less if the model is

tested with the non-fraud data $b^{(n)}$, since the model is trained with the same (non-fraud transaction data type), i.e. $\eta_r(b^{(n)}, \tilde{b}^{(n)}) < \Delta$. If the auto-encoder model is tested with fraud data $b^{(f)}$, the reconstruction loss will be higher since the data class is not trained in the auto-encoder, i.e. $\eta_r(b^{(f)}, \tilde{b}^{(f)}) > \Delta$. Thus, Δ be the threshold to refine the transaction data augmented by the GAN network. The proposed GAN-AE uses a threshold of $\Delta = 0.2$. The fraudulent data must be considered only if the loss function corresponding to the data generated by the GAN generator (G) is greater than Δ. Otherwise, the generated data must not be considered. Thus, the auto-encoder refines the data generated by the GAN network.

(ii) Hybrid GNN-LSTM network for fraud detection

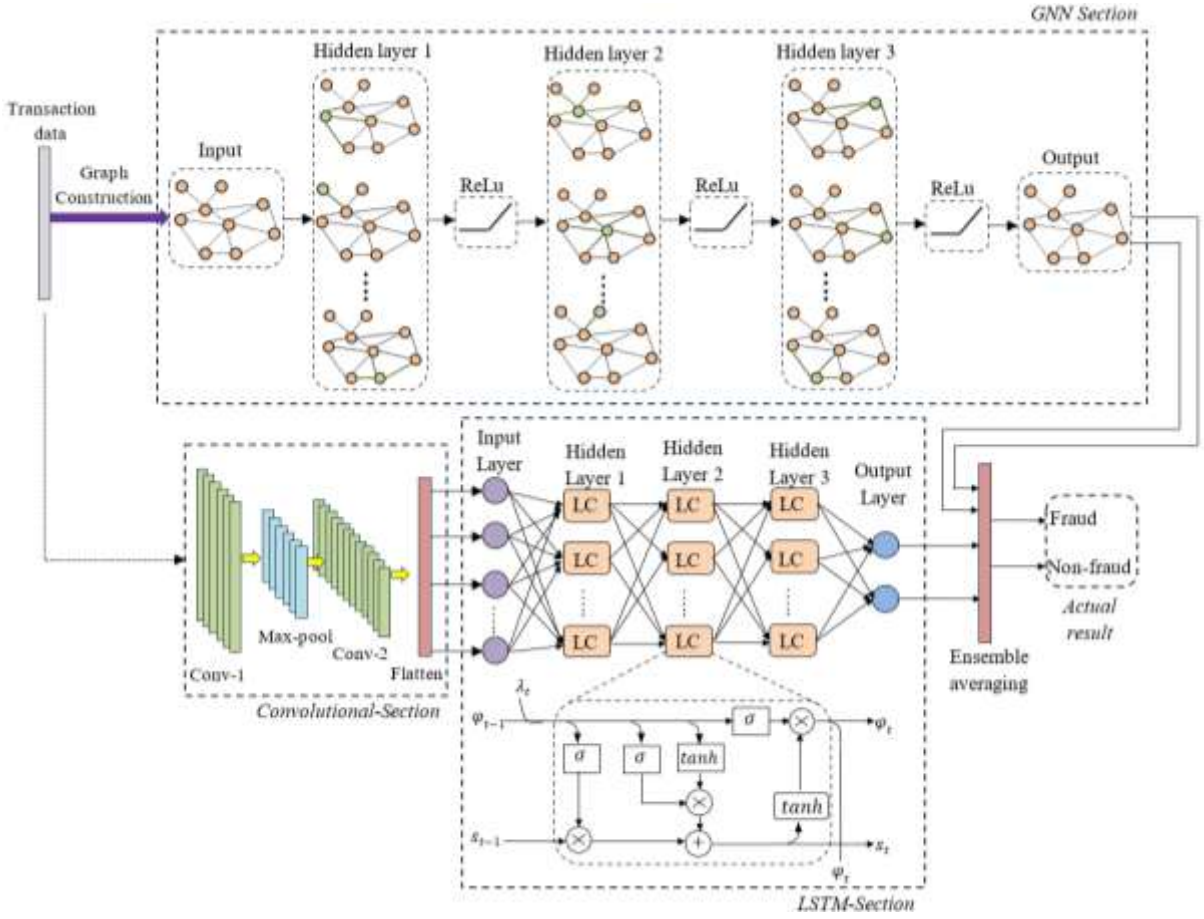

Fig. 3: Structure of the proposed Hybrid GNN-LSTM network

The structure of the suggested GNN-LSTM network to detect the fraud transaction data is illustrated in Fig. 3. The architecture has two subnetworks, namely GNN and LSTM, which extract the relational and sequential descriptions, respectively. To enhance the sequential descriptor pattern, different convolutional filters are utilized in the convolutional network and used before the LSTM network. Finally, an ensemble averaging layer is utilized to compute the overall prediction probability from the prediction probability of the GNN and LSTM networks.

(a) Graph neural network

The first stage of GNN [29] is the generation of the graph α that have a vertex ρ and edges γ i.e. $\alpha = (\beta, \gamma)$. The node β represent the information regarding merchant, user, IP address, credit card, etc, while edge γ resembles the relationship. In GNN, each edge and node is represented as a descriptor. This descriptor includes both the static descriptor and the dynamic descriptor. By aggregating the neighbour descriptor, the GNN updates the embeddings $z_\alpha^{(l)}$ of each node α at iteration l using the relation,

$$z_{\alpha}^{(l)} = UPDATE^{(l)}\left(z_{\alpha}^{(l-1)}, AGGREGATE^{(l)}\left(\left\{z_{s}^{(l-1)} : s \in \psi(\alpha)\right\}\right)\right) \quad (6)$$

Here, $\psi(\alpha)$ represents the neighbours of the node α. A soft-max layer is finally applied to the output of the final GNN layer to differentiate the transaction as fraud or non-fraud. The GNN uses the following binary cross-entropy-based loss function to update the node embedding.

$$\eta_a = \sum_{\alpha \in \alpha_L} \left[(1 - y_\alpha)\, log\, log\, \frac{1}{(1-\hat{y}_\alpha)} + y_\alpha\, log\, log\, \frac{1}{\hat{y}_\alpha}\right] \quad (7)$$

Here, $\alpha \in \alpha_L$ resembles the set of nodes that are labelled as non-fraud or fraud. In the above equation $\hat{y}_\alpha$ resembles the predicted probability of GNN, y_α represents the true label for the node α. The GNN uses 3 hidden layers, where the hidden layer has a size of 64. For training the GNN, the Adam optimizer with a learning rate of 0.005, batch size of 512 and epoch of 100 is used. The GCN type GNN network is utilized with a neighbourhood sampling of 20.

(b) LSTM

The LSTM [29] uses the descriptors that are enhanced by the convolutional section. The convolutional section has two sections of a convolutional layer (conv1 and conv2) and a max-pooling layer. The conv-1 and conv-2 filters are 1-dimensional, having 16 and 32 filters, respectively. The two filters have a kernel size of 1×3 and a stride of 1. The max-pooling layer uses a stride of 2. The convolutional filter section computes different sequential patterns, and these sequential patterns are flattened and fed as input to the LSTM network. Let λ_t resemble the flattened descriptor extracted by the convolutional section at time t, and τ resemble the number of past transactions made by the customer. Let s_t resembles the cell state of the LSTM, and φ_t resembles the current hidden state. Using the past hidden state φ_{t-1} and past cell state s_{t-1}, and the input λ_t the LSTM computes the current hidden state φ_t and current cell state as,

$$(\varphi_t, s_t) = LSTM(\lambda_t, \varphi_{t-1}, \varphi_{t-1}) \quad (8)$$

Fig. 3 resembles the structure of the LSTM network with a convolutional section. In this figure, LC resembles the LSTM cell. Let, φ_τ resembles the final hidden state which is utilized to estimate the predicted probability as,

$$\hat{y}_\omega = \sigma(w_c \varphi_t + \mu_c) \quad (9)$$

Here w_c and μ_c resembles the weight and bias in LSTM, and σ resembles the sigmoid activation. The LSTM network also uses the binary cross-entropy loss function to update the weights and biases. To train the LSTM along with a convolutional section, Adam optimizer at a learning rate of $1e-5$ is used. The training was done for 25 epochs with a batch size of 32. Let $\hat{y}_\alpha^{(x)}$ resembles the predicted probability computed for the test transaction data x on the GNN network. Similarly, let, $\hat{y}_\omega^{(x)}$ resembles the predicted probability computed for the test transaction data x on the LSTM network. The actual predicted probability to differentiate the fraud and non-fraud classes can be computed by the ensemble averaging layer using the probabilities $\hat{y}_\alpha^{(x)}$ and $\hat{y}_\omega^{(x)}$ as,

$$\hat{Y} = \rho \hat{y}_\alpha^{(x)} + (1-\rho)\hat{y}_\omega^{(x)} \quad (10)$$

Here, ρ resembles the balancing parameter that provides weightage to the predicted probabilities of the two networks, GNN and LSTM. The analysis of the GNN-LSTM network is presented in the next section.

3. Experimental Results

Datasets, namely the European credit card-2013 (ECH-2013) dataset [30] and the PaySim dataset [31], are utilized for evaluating the suggested GNN-LSTM approach. The credit card holder transaction of Europeans for the month of September 2013 is provided in the ECH-2013 dataset [30]. The dataset

contains 284,315 non-fraud transactions and only 492 fraud transactions. The number of fraudulent transactions is very much lower than the non-fraudulent transactions (fraudulent transaction is only 0.172%). The transaction is represented by 31 descriptors. To preserve the privacy of the cardholder, including transaction time and transaction amount, the principal component analysis (PCA) is utilized to obtain 28 principal components, V1 to V28. The PaySim dataset [31] contains synthetic data that was generated using the real transactions of African mobile money services. This dataset contains 8213 fraudulent transaction data out of 6,362,620 transactions. This dataset contains 11 features to represent the transaction, such as sender, amount, transaction type (transfer, payment, debit, cash-out, cash-in), final/initial account balances, recipient information, sender, and amount. The two datasets are initially pre-processed with processes such as mean value-based missing value imputation and min-max normalization-based feature scaling.

Table 1: Representation of the number of transaction data used for analysis

Data	ECH-2013 dataset		PaySim dataset	
	Fraud	Non-fraud	Fraud	Non-fraud
Actual	492	284,315	8213	6,354,407
Undersampling	492	28,431	8213	63,544
GAN-AE (augmentation)	28,431	28,431	63,544	63,544

To make the dataset balanced, the non-fraud data is under-sampled, and the fraudulent data is augmented by the GAN-AE. The number of transaction data after under-sampling and augmenting is listed in Table 1. In this, 80% of the data is utilized to train the GNN-LSTM network, and 20% is used to test the model. Only the fraudulent data is utilized to train the GAN-AE to generate augmented fraudulent data. Evaluation metrics such as F1-score, recall, precision and accuracy are utilized to evaluate the fraudulent transaction detection approach.

$$F1 - score = \frac{\delta_{tp}}{\frac{1}{2}(\delta_{fp}+\delta_{fn})+\delta_{tp}} \times 100\% \quad (11)$$

$$Recall = \frac{\delta_{tp}}{\delta_{tp}+\delta_{fn}} \times 100\% \quad (12)$$

$$Precision = \frac{\delta_{tp}}{\delta_{fp}+\delta_{tp}} \times 100\% \quad (13)$$

$$Accuracy = \frac{\delta_{tn}+\delta_{tp}}{\delta_{fn}+\delta_{fp}+\delta_{\mathrm{tp}}+\delta_{\mathrm{tn}}} \times 100\% \quad (14)$$

Here δ_{fp}, δ_{tp}, δ_{tn} and δ_{fn} resembles the false positives, true positives, true negatives and false negatives results obtained by GNN-LSTM in fraud transaction detection. The performance of the suggested GNN-LSTM approach was compared with the traditional credit card fraud detection schemes, such as Multistage [6], FedLearn [32], Fuzzy-detect [22], Encoder-GNN [24], Seq-Model [7], and GNN-RL [25] approaches.

Table 2: Performance comparison between GNN-LSTM and other similar fraud detection approaches when evaluated using the ECH-2013 dataset

Scheme	Recall (%)	Precision (%)	F1-score (%)	Accuracy (%)	AUC
Multistage [6]	85.29 ± 1.21	89.54 ± 0.98	87.46 ± 1.05	91.34 ± 0.72	0.9341 ± 0.010
FedLearn [32]	87.94 ± 1.09	90.76 ± 0.91	88.41 ± 0.97	89.79 ± 0.84	0.9463 ± 0.009

Fuzzy-detect [22]	89.81 ± 0.96	92.74 ± 0.83	89.76 ± 0.92	92.63 ± 0.68	0.9531 ± 0.008
Encoder-GNN [24]	92.57 ± 0.78	93.24 ± 0.71	91.48 ± 0.81	94.18 ± 0.55	0.9453 ± 0.009
Seq-Model [7]	95.36 ± 0.64	95.76 ± 0.59	93.45 ± 0.67	96.79 ± 0.42	0.9654 ± 0.006
GNN-RL [25]	97.83 ± 0.41	96.91 ± 0.48	92.17 ± 0.61	96.93 ± 0.39	0.9751 ± 0.005
Proposed	98.93 ± 0.28	96.12 ± 0.44	93.98 ± 0.52	97.91 ± 0.31	0.9879 ± 0.003

Table 2 illustrates the comparison of performance between the proposed GNN-LSTM approach and other recent approaches when evaluated using the ECH-2013 dataset. The precision computed on the suggested approach is 0.79% lower than the GNN-RL approach. However, the recall, F1-score, and accuracy computed on the suggested GNN-LSTM approach are 1.1%, 1.81% and 0.98% respectively higher than the GNN-RL approach. Also, the recall, precision, F1-score and accuracy provided by the suggested GNN-LSTM approach are 3.5%, 0.36%, 0.53% and 1.12% respectively higher than the Seq-Model approach. To ensure that performance gains are not artifacts of severe class imbalance or sampling variability, we report 95% confidence intervals computed via stratified bootstrap resampling. Statistical significance was evaluated against the strongest prior baseline (GNN-RL) using McNemar's test for paired classification errors and DeLong's test for AUC-ROC. The proposed model achieves statistically significant improvements in recall, F1-score, accuracy, and AUC ($p < 0.01$), with narrow confidence intervals indicating robust and stable performance.

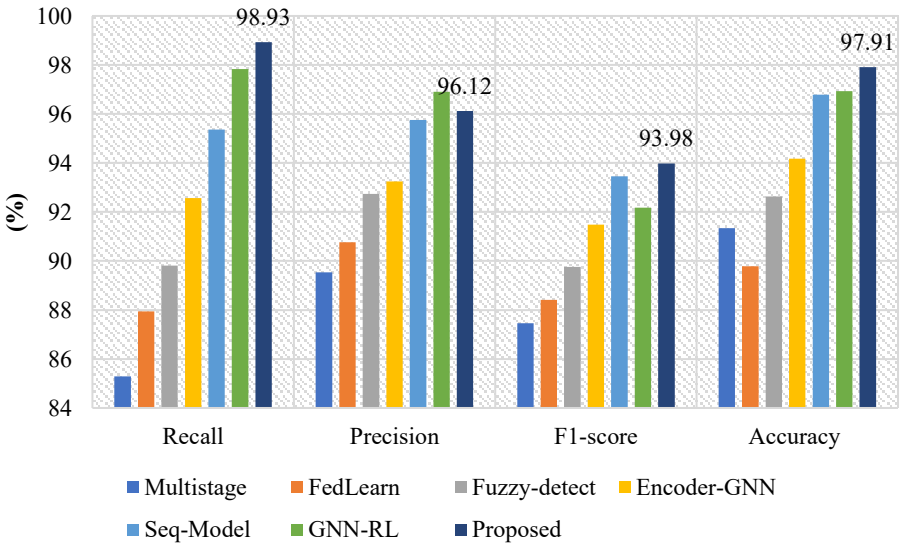

Fig. 4: Graphical comparison between GNN-LSTM and other similar fraud detection approaches performance when evaluated using the ECH-2013 dataset

The graphical comparison of performance between the suggested GNN-LSTM approach and other similar fraud detection schemes when evaluated utilizing the ECH-2013 dataset is depicted in Fig. 4. The GNN-LSTM approach results in a recall, precision, F1-score and accuracy of 98.93%, 96.12%, 93.98% and 97.91%, respectively, when evaluated using the ECH-2013 dataset

Table 3: Performance comparison between GNN-LSTM and other similar fraud detection approaches when evaluated using the PaySim dataset

Scheme	Recall (%)	Precision (%)	F1-score (%)	Accuracy (%)	AUC
Multistage [6]	85.92 ± 1.18	89.63 ± 0.96	88.06 ± 1.02	90.41 ± 0.76	0.9372 ± 0.010
FedLearn [32]	88.75 ± 1.04	91.68 ± 0.89	87.69 ± 0.98	90.49 ± 0.81	0.9397 ± 0.009
Fuzzy-detect [22]	89.06 ± 0.93	93.67 ± 0.79	89.60 ± 0.90	93.50 ± 0.62	0.9572 ± 0.007
Encoder-GNN [24]	93.40 ± 0.74	92.56 ± 0.77	92.31 ± 0.79	94.54 ± 0.53	0.9359 ± 0.010
Seq-Model [7]	95.62 ± 0.61	96.70 ± 0.56	94.03 ± 0.69	97.31 ± 0.41	0.9609 ± 0.006
GNN-RL [25]	97.03 ± 0.47	97.82 ± 0.43	93.09 ± 0.63	97.42 ± 0.38	0.9660 ± 0.005
Proposed	98.49 ± 0.31	97.75 ± 0.39	95.01 ± 0.51	98.09 ± 0.29	0.9798 ± 0.004

The comparison of performance between the GNN-LSTM approach and other similar approaches in detecting fraudulent transactions when evaluated using the PaySim dataset is presented in Table 3. The precision computed by the suggested GNN-LSTM approach is 0.07% lower than the GNN-RL approach. However, the recall, F1-score, and accuracy computed on the GNN-LSTM approach are 1.46%, 1.92% and 0.67% respectively higher than the GNN-RL approach. Also, the recall, precision, F1-score and accuracy computed on the GNN-LSTM approach are 2.86%, 1.05%, 0.98% and 0.78% higher than the Seq-Model approach. To verify that performance gains are not attributable to class imbalance or sampling variability, 95% confidence intervals were computed via stratified bootstrap resampling. Statistical significance was assessed against the strongest baseline (GNN-RL) using McNemar's test and DeLong's test. The proposed method achieves statistically significant improvements in recall, F1-score, accuracy, and AUC-ROC ($p < 0.01$), with narrow confidence intervals indicating stable and robust performance.

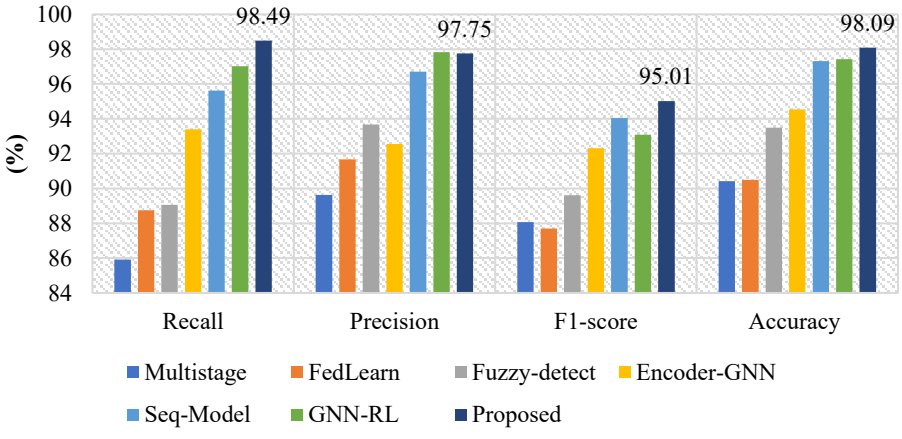

Fig. 5: Graphical comparison between GNN-LSTM and other fraud detection approaches performance when evaluated using the PaySim dataset

The graphical illustration of performance comparison between the suggested GNN-LSTM and other fraud detection approaches taken for comparison when evaluated using the PaySim dataset is provided in Fig. 5. In the case of PaySim dataset, the recall, precision, F1-score and accuracy was computed as 98.49%, 97.75%, 95.01% and 98.09% respectively. In both datasets, the proposed GNN-LSTM results in higher performance in terms of the scales, namely recall, F1-score, and accuracy, than traditional approaches.

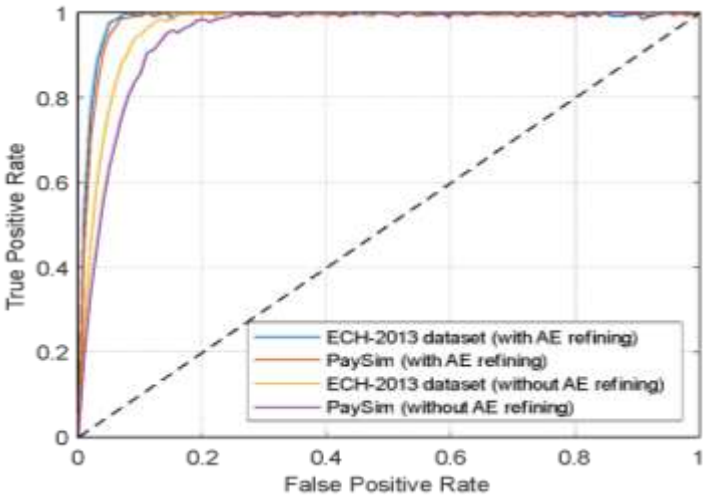

Fig. 6: ROC comparison with and without autoencoder-based refining

The suggested hybrid GNN-LSTM network was evaluated with the test images generated with AE refining and without AE refining. With the use of AE refining, the GNN-LSTM network results in a higher AUC than without using AE refining in both datasets, as illustrated in Fig. 6. Without using AE refining, the suggested GNN-LSTM network results in an AUC of 0.9475 and 0.9594 when evaluated using the ECH-2013 and PaySim datasets, respectively. With the use of AE refining, the suggested GNN-LSTM network results in an AUC of 0.9879 and 0.9798 when evaluated using the ECH-2013 and PaySim datasets, respectively. This suggests that the proposed model was well trained with the use of AE refined data.

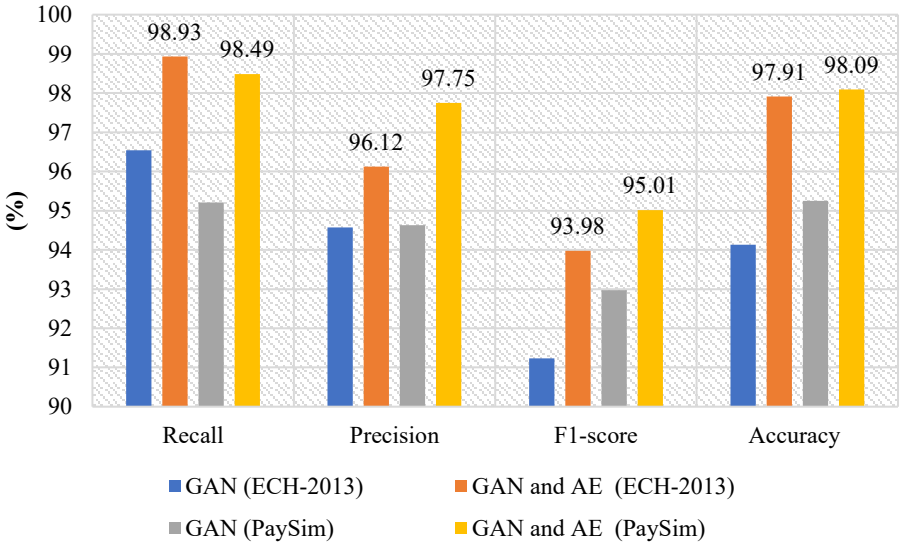

Fig. 7: Impact of autoencoder-based refining on generating fraudulent data in fraud transaction detection performance

The performance attained by the suggested GNN-LSTM network when the model is trained with GAN-generated data with and without AE-based refining is illustrated in Fig. 7. The fraud detection performance is higher if the model is trained with the training images generated by the GAN that are refined using the autoencoder. In the case of the ECH-2013 dataset, the recall, precision, F1-score, and accuracy improve by 2.39%, 1.55%, 2.75%, and 3.78%, respectively, when the model is trained with the autoencoder refined data. Similarly, in the case of the PaySim dataset, the recall, precision, F1-score, and accuracy improve by 3.28%, 3.12%, 2.04%, and 2.84%, respectively, when the model is trained with the autoencoder refined data.

Table 4: Ablation experiment conducted for the suggested GNN-LSTM-based fraud detection

	ECH-2013 dataset				**PaySim dataset**			
Process	Rec (%)	Pre (%)	F1 (%)	Acc (%)	Rec (%)	Pre (%)	F1 (%)	Acc (%)
GN	94.31 ± 0.82	92.57 ± 0.77	89.41 ± 0.91	93.17 ± 0.68	93.87 ± 0.85	92.79 ± 0.79	90.21 ± 0.88	94.03 ± 0.67
LN	93.63 ± 0.88	91.87 ± 0.83	87.48 ± 0.96	92.74 ±0.71	92.19 ± 0.91	92.41 ± 0.84	89.03 ± 0.93	93.94 ± 0.69
CN+LN	95.74 ± 0.69	93.67 ± 0.65	90.82 ± 0.78	94.26 ± 0.56	94.15 ± 0.74	93.46 ± 0.68	91.63 ± 0.80	95.71 ± 0.53
GN+LN	97.03 ± 0.52	95.09 ± 0.57	91.66 ± 0.71	96.08 ± 0.42	96.78 ± 0.57	95.63 ± 0.59	93.78 ± 0.66	96.95 ± 0.41
GN+LN+CN	98.93 ± 0.31	96.12 ± 0.54	93.98 ± 0.53	97.91 ± 0.29	98.49 ± 0.33	97.75 ± 0.39	95.01 ± 0.51	98.09 ± 0.29

Let GN, LN, CN resemble the process performed on GNN network, LSTM network and Convolution network, respectively, used in the proposed architecture. To evaluate the importance of each network in the proposed architecture, an ablation study was conducted, and the performance was evaluated as illustrated in Table 4. In this table, GN resemble the architecture that uses only the GNN network, which results in a recall (Rec), precision (Pre), F1-score (F1) and accuracy (Acc) of 94.31%, 92.57%, 89.41% and 93.17% when evaluated using the ECH-2013 dataset. With the same dataset, using only the LSTM network results in recall, precision, F1-score and accuracy of 93.63%, 91.87%, 87.48% and 92.74%, respectively. Using a Convolutional network (CN) before the LSTM network results in recall, precision, F1-score and accuracy of 95.74%, 93.67%, 90.92% and 94.26%, respectively. Using both GNN and LSTM without the convolutional network results in recall, precision, F1-score and accuracy of 97.03%, 95.06%, 91.66% and 96.08%, respectively. Using the Convolutional network along with GNN and LSTM improves the recall, precision, F1-score and accuracy by 1.9%, 1.03%, 2.32%, and 1.83%, respectively, when evaluated using the ECH-2013 dataset. With the PaySim dataset, the usage of a convolutional network before the LSTM and ensembling the GNN and LSTM network results in an accuracy of 98.09%, which is 1.14% higher than without using the convolutional network (only using GNN and LSTM). Thus, the three modules play a major role in the suggested fraud detection approach in identifying the fraudulent transactions.

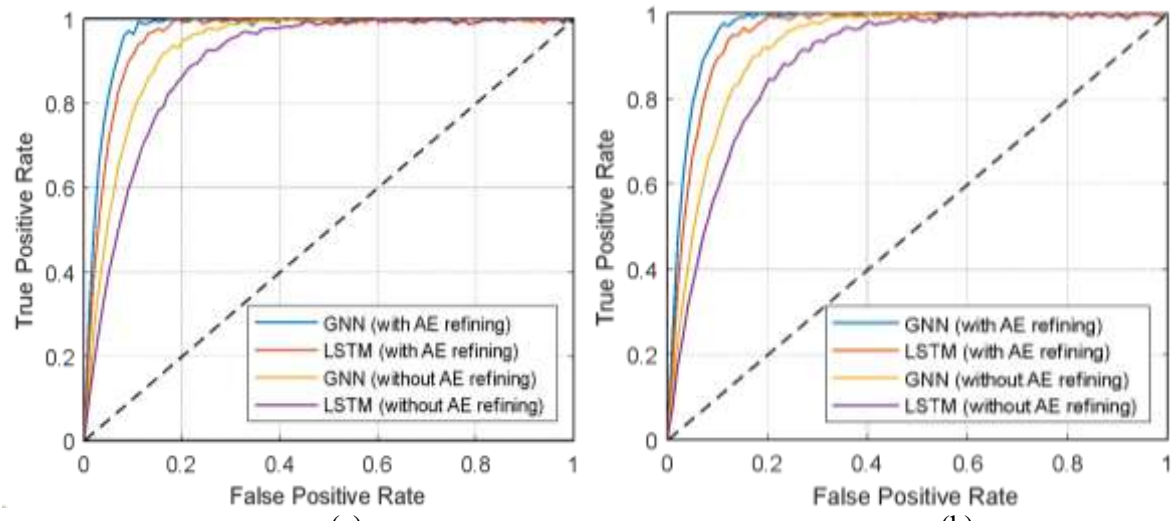

Fig. 8: ROC comparison between GNN and LSTM sub-network with and without autoencoder-based refining (a) ECH-2013 dataset (b) PaySim dataset

The ROC curve corresponding to training the model using the GAN-generated images with and without using the AE is presented in Fig. 8. The AUC obtained in GNN is higher than the AUC obtained in the LSTM network in both datasets, with both the AE-refined data and unrefined data. This shows that the model gets trained well with the use of AE refining on two datasets.

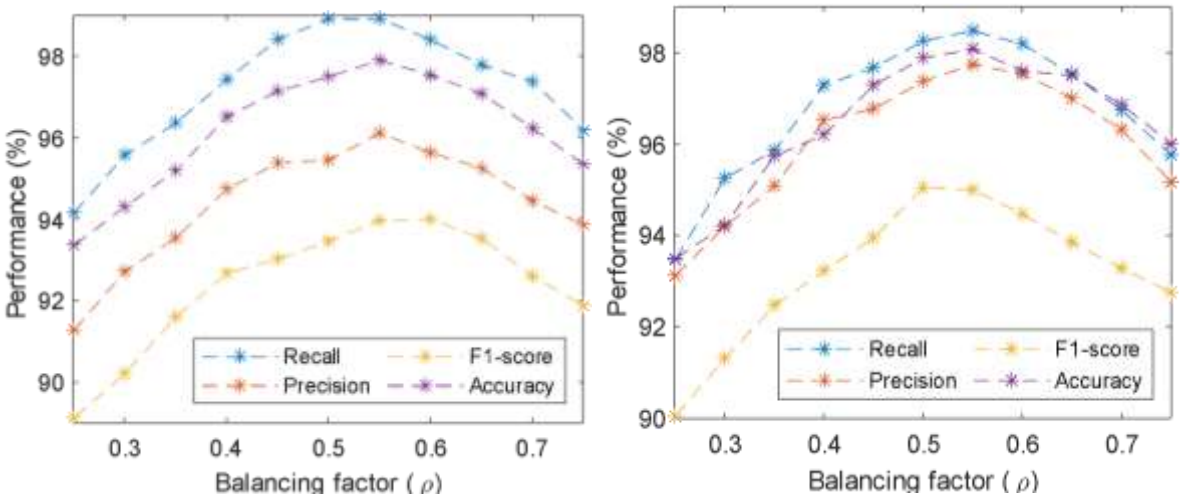

Fig. 9: Variation of GNN-LSTM network performance with respect to different values of balancing parameter when evaluated using two datasets (a) ECH-2013 dataset, (b) PaySim dataset

The variation of performance scales such as recall, F1-score, precision, and accuracy with respect to different values of the balancing parameter (ρ) is presented in Fig. 9. A balancing parameter of $\rho = 0.5$ resembles that the ensemble averaging layer assigns equal weightage to the predicted probability of the GNN and LSTM network in computing the actual predicted probability. This evaluation was performed by varying the balancing parameter between 0.25 and 0.75. As the balancing parameter is increased from 0.25, the maximum performance is attained with $\rho = 0.55$, for a further rise in the parameter ρ, reduces the detection performance of the GNN-LSTM network in both datasets.

Table 5: Computation complexity (in seconds) of the suggested GNN-LSTM-based fraudulent transaction identification approach

Process	ECH-2013 dataset			PaySim dataset		
	GNN	LSTM	Total	GNN	LSTM	Total
Training	1947	944	2,891	6437	3048	9,485
Testing	0.239	0.147	0.386	0.284	0.179	0.463

The suggested GNN-LSTM network for identifying transaction fraud was implemented using MATLAB 2023a on a PC with an Intel i7 processor on Windows 10 OS, having 16GB of RAM. The computational complexity of the suggested GNN-LSTM for the two datasets is presented in Table 5. When comparing the GNN and LSTM networks, the training time is higher for the GNN network than the LSTM network. For ECH-2013 and PaySim datasets, the proposed GNN-LSTM structure results in training time of 2,891s and 9,485s, respectively. The testing time (time to detect the fraud or non-fraud transaction) of the proposed GNN-LSTM network is estimated as 0.386s and 0.463s, respectively, when evaluated using the ECH-2013 dataset and PaySim dataset. The testing time of the GNN network is higher than the testing time of the LSTM network in both datasets.

4. Conclusion

This paper proposed a credit card fraud transaction identification approach that uses a Graph neural network along with an LSTM network to obtain significant feature patterns and temporal information from the transaction data. The paper also proposes a data augmentation approach for generating fraudulent transactions to train the model with the use of GAN and autoencoder architectures. The usage of the autoencoder helps to avoid the usage of non-fraudulent training data generated by the GAN as fraudulent data (fuzzy data) in training the proposed GNN-LSTM architecture. Thus, the data generated by the generator of the GAN network is refined using the autoencoder approach to confirm whether the data generated by the GAN network belongs to the correct fraud or non-fraud type. The refined data is then utilized to train the suggested GNN-LSTM approach. The use of LSTM on the GNN network helps to collect temporal information. The predicted probability of the LSTM and GNN networks is used to obtain the actual predicted probability of the detection system with the use of an ensemble averaging layer. The analysis of the suggested fraud detection approach is evaluated using the scales such as precision, F1-score, recall, and accuracy on the ECH-2013 dataset and the PaySim dataset. The proposed GNN-LSTM results in a recall and accuracy of 98.93% and 97.91% when evaluated using the ECH-2013 dataset. In the case of the PaySim dataset, the GNN-LSTM approach results in a recall and accuracy of 98.49% and 98.09%, respectively.

References

[1]. Khando, K., Islam, M. S., & Gao, S. (2022). The emerging technologies of digital payments and associated challenges: a systematic literature review. Future Internet, 15(1), 21.
[2]. Association of Certified Fraud Examiners, Report to the Nations: 2020 Global Study on Occupational Fraud and Abuse, West Ave, Austin: ACFE Global Headquarters, 2020.
[3]. Chaudhary, K., Yadav, J., & Mallick, B. (2012). A review of fraud detection techniques: Credit card. International Journal of Computer Applications, 45(1), 39-44.
[4]. Mienye, I. D., & Jere, N. (2024). Deep learning for credit card fraud detection: A review of algorithms, challenges, and solutions. IEEE Access.
[5]. Asha, R. B., & KR, S. K. (2021). Credit card fraud detection using artificial neural network. Global Transitions Proceedings, 2(1), 35-41.
[6]. Zioviris, G., Kolomvatsos, K., & Stamoulis, G. (2022). Credit card fraud detection using a deep learning multistage model. The Journal of Supercomputing, 78(12), 14571-14596.
[7]. Zioviris, G., Kolomvatsos, K., & Stamoulis, G. (2024). An intelligent sequential fraud detection model based on deep learning. The Journal of Supercomputing, 80(10), 14824-14847.
[8]. Ileberi, E., Sun, Y., & Wang, Z. (2022). A machine learning based credit card fraud detection using the GA algorithm for feature selection. Journal of Big Data, 9(1), 24.
[9]. Talukder, M. A., Khalid, M., & Uddin, M. A. (2024). An integrated multistage ensemble machine learning model for fraudulent transaction detection. Journal of Big Data, 11(1), 168.

[10]. Rehman, A., Awan, K. A., Al-Rasheed, A., Ara, A., Alruwaili, F. F., Al-Otaibi, S., & Saba, T. (2025). A novel hybrid fuzzy logic and federated learning framework for enhancing cybersecurity and fraud detection in IoT-enabled metaverse transactions. Egyptian Informatics Journal, 30, 100668.
[11]. Ashfaq, T., Khalid, R., Yahaya, A. S., Aslam, S., Azar, A. T., Alsafari, S., & Hameed, I. A. (2022). A machine learning and blockchain based efficient fraud detection mechanism. Sensors, 22(19), 7162.
[12]. Wang, C., & Zhu, H. (2022). Wrongdoing monitor: A graph-based behavioral anomaly detection in cyber security. IEEE Transactions on Information Forensics and Security, 17, 2703-2718.
[13]. Baabdullah, T., Alzahrani, A., Rawat, D. B., & Liu, C. (2024). Efficiency of federated learning and blockchain in preserving privacy and enhancing the performance of credit card fraud detection (CCFD) systems. Future Internet, 16(6), 196.
[14]. Almarshad, F. A., Gashgari, G. A., & Alzahrani, A. I. (2023). Generative adversarial networks-based novel approach for fraud detection for the european cardholders 2013 dataset. IEEE Access, 11, 107348-107368.
[15]. Huang, D., Mu, D., Yang, L., & Cai, X. (2018). CoDetect: Financial fraud detection with anomaly feature detection. Ieee Access, 6, 19161-19174.
[16]. Fiore, U., De Santis, A., Perla, F., Zanetti, P., & Palmieri, F. (2019). Using generative adversarial networks for improving classification effectiveness in credit card fraud detection. Information Sciences, 479, 448-455.
[17]. Roseline, J. F., Naidu, G. B. S. R., Pandi, V. S., alias Rajasree, S. A., & Mageswari, N. (2022). Autonomous credit card fraud detection using machine learning approach☆. Computers and Electrical Engineering, 102, 108132.
[18]. Zhang, Z., Zhou, X., Zhang, X., Wang, L., & Wang, P. (2018). A model based on convolutional neural network for online transaction fraud detection. Security and Communication Networks, 2018(1), 5680264.
[19]. Taha, A. A., & Malebary, S. J. (2020). An intelligent approach to credit card fraud detection using an optimized light gradient boosting machine. IEEE access, 8, 25579-25587.
[20]. Fanai, H., & Abbasimehr, H. (2023). A novel combined approach based on deep Autoencoder and deep classifiers for credit card fraud detection. Expert Systems with Applications, 217, 119562.
[21]. Esenogho, E., Mienye, I. D., Swart, T. G., Aruleba, K., & Obaido, G. (2022). A neural network ensemble with feature engineering for improved credit card fraud detection. IEEE access, 10, 16400-16407.
[22]. Charizanos, G., Demirhan, H., & İçen, D. (2024). An online fuzzy fraud detection framework for credit card transactions. Expert Systems with Applications, 252, 124127.
[23]. Mienye, I. D., & Swart, T. G. (2024). A hybrid deep learning approach with generative adversarial network for credit card fraud detection. Technologies, 12(10), 186.
[24]. Cherif, A., Ammar, H., Kalkatawi, M., Alshehri, S., & Imine, A. (2024). Encoder–decoder graph neural network for credit card fraud detection. Journal of King Saud University-Computer and Information Sciences, 36(3), 102003.
[25]. Cui, Y., Han, X., Chen, J., Zhang, X., Yang, J., & Zhang, X. (2025). FraudGNN-RL: A Graph Neural Network With Reinforcement Learning for Adaptive Financial Fraud Detection. IEEE Open Journal of the Computer Society.
[26]. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2020). Generative adversarial networks. Communications of the ACM, 63(11), 139-144.
[27]. Zhang, Y. (2018, March). A better autoencoder for image: Convolutional autoencoder. In ICONIP17-DCEC. Available online: http://users. cecs. anu. edu. au/Tom. Gedeon/conf/ABCs2018/paper/ABCs2018_paper_58. pdf (accessed on 23 March 2017).
[28]. Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., & Yu, P. S. (2020). A comprehensive survey on graph neural networks. IEEE transactions on neural networks and learning systems, 32(1), 4-24.
[29]. Yu, Y., Si, X., Hu, C., & Zhang, J. (2019). A review of recurrent neural networks: LSTM cells and network architectures. Neural computation, 31(7), 1235-1270.
[30]. Andrea Dal Pozzolo, Olivier Caelen, Reid A. Johnson and Gianluca Bontempi. Calibrating Probability with Undersampling for Unbalanced Classification. In Symposium on Computational Intelligence and Data Mining (CIDM), IEEE, 2015

[31]. E. A. Lopez-Rojas , A. Elmir, and S. Axelsson. "PaySim: A financial mobile money simulator for fraud detection". In: The 28th European Modeling and Simulation Symposium-EMSS, Larnaca, Cyprus. 2016
[32]. Aurna, N. F., Hossain, M. D., Taenaka, Y., & Kadobayashi, Y. (2023, July). Federated learning-based credit card fraud detection: Performance analysis with sampling methods and deep learning algorithms. In 2023 IEEE International Conference on Cyber Security and Resilience (CSR) (pp. 180-186). IEEE.