IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 22 August 2024, accepted 19 September 2024, date of publication 23 September 2024, date of current version 31 January 2025.

Digital Object Identifier 10.1109/ACCESS.2024.3466288

RESEARCH ARTICLE

Enhancing Fraud Detection in Banking With Deep Learning: Graph Neural Networks and Autoencoders for Real-Time Credit Card Fraud Prevention

FAWAZ KHALED ALARFAJ[1] **AND SHABNAM SHAHZADI**[2]

[1]Department of Management Information Systems (MIS), School of Business, King Faisal University (KFU), Al-Ahsa 31982, Saudi Arabia
[2]Department of Mathematics and Big Data, Anhui University of Science and Technology, Huainan 232001, China

Corresponding author: Shabnam Shahzadi (shabnamsarwar2@gmail.com)

This work was supported by the Deanship of Scientific Research, Vice Presidency for Graduate Studies and Scientific Research, King Faisal University (KFU), Saudi Arabia, under Grant KFU241890.

ABSTRACT Under the umbrella of artificial intelligence (AI), deep learning enables systems to cluster data and provide incredibly accurate results. This study explores deep learning for fraud detection, utilizing Graph Neural Networks (GNNs) and Autoencoders to enhance business practices and reduce fraudulent activities in large organizations. For real-time fraud detection, we propose Graph neural network with lambda architecture while for credit card fraud detection, we use an autoencoder, validated through case studies from two banks. The findings demonstrate that these methods effectively detect fraud with balance of precision and recall, improving the efficiency of banking systems. Python is employed for analysis, emphasizing the ability of deep learning to manage and prevent fraud in real-time on dynamic datasets. In the end, this study concludes that by using deep learning algorithms, we can control online credit card fraud detection in banks, improve the efficiency of the banking system. We can manage fraudulent activity in real-time and on dynamic datasets by utilizing deep learning algorithms, which allows for ongoing improvement of the fraud detection and prevention system.

INDEX TERMS Deep learning, credit card, fraud detection, graph neural network, autoencoders.

I. INTRODUCTION

Banking fraud encompasses a wide range of illegal activities, including identity theft, credit card fraud; check fraud, account takeover, phishing scams, and more [1]. These activities can include unauthorized access to accounts, fraudulent transactions, identity theft, and other fraudulent behaviors. To prevent these frauds, banks employ a multi-pronged approach. Banking fraud detection systems aim to detect and respond to these fraudulent activities in real-time or near real-time to mitigate the potential damage. Financial organizations employ fraud detection systems capable of analyzing transaction data in real-time, enabling the prompt identification of suspicious activities [2]. Strong customer authentication methods, such as two-factor authentication (2FA) and biometric verification, are employed to verify the identity of individuals accessing accounts.

Detecting banking fraud presents a set of complex challenges for financial institutions [3]. One of the primary challenges is the evolving nature of fraudulent methods. Fraudsters develop new methods to exploit vulnerabilities in the system, making it difficult for traditional rule-based systems to keep up. The sheer volume of transactions and data processed by banks further complicates the task. Manual analysis is time-consuming and often ineffective in identifying subtle patterns and anomalies indicative of fraud. Additionally, the need to balance fraud detection with customer convenience is a delicate task; overly strict security

The associate editor coordinating the review of this manuscript and approving it for publication was Chuan Li.

© 2024 The Authors. This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License.
For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

measures can lead to false positives and inconvenience legitimate customers, potentially driving them away.

Deep learning has emerged as a powerful solution to solve the challenges of banking fraud detection [4]. By leveraging advanced algorithms and artificial intelligence, deep learning models can process huge amounts of data and learn from historical transaction patterns [5]. This enables them to identify unusual activities and patterns that may signify fraud. Deep learning models continuously adapt and improve their accuracy as they encounter new data, making them highly effective in staying ahead of evolving fraud techniques. It can also reduce false positives by considering a broader range of factors and contextual information, enhancing the overall security of the banking system while maintaining a positive customer experience. As technology continues to advance, machine learning [6] remains at the forefront of the fight against banking fraud, providing a dynamic and robust defense for financial institutions and the clients.

Understanding the role that model-free, uncertain AI-based graph theory approaches play in enhancing the precision and resilience of graph-based machine learning models is essential for addressing recent developments in these areas. In particular, incorporating these cutting-edge methods into your study can offer a thorough comprehension of how uncertainties might be managed more skillfully. We address these techniques' applicability and possible advantages below, mentioning the particular strategy outlined for large-scale epidemiological modeling.

State-of-the-Art, Uncertain AI-Based Graph Theory Methods

Synopsis of Advanced Techniques

1. Model-Free Approaches: These methods don't rely on pre-established models or distributional assumptions about the data. Rather, they concentrate on actively learning from the data, which can be very useful when working with noisy, complex, and high-dimensional data.
2. *Uncertain AI-Based Methods:* These techniques take data and model uncertainties into explicit account. By strengthening the learning process' resistance to fluctuations and noise, they hope to improve the model's capacity to generalize to previously unobserved data.
3. *Multi-Dimensional Constrained Optimization:* This technique gives the optimization process more exact control over the learning dynamics by incorporating a number of constraints and targets. Applications such as epidemiological modeling, where multiple parameters (e.g., transmission rates, intervention effects) must be considered at the same time, benefit greatly from this.

Mainly, this study is based on fraud detection with graph neural networks and anomaly detection by using autoencoders in credit cards to improve business practices in large organizations and reduce fraudulent activities. To increase productivity, we merged well-known deep learning algorithms with data mining for business intelligence. A successful experiment relies on discovering significant patterns in vast amounts of data, which transforms experience into expertise. The primary focus of the study is on the two screwdrivers of Graph Neural Network and Autoencoders Models to process enormous volumes of data for enhancing visibility and transparency throughout all business aspects and across sectors, which help explore decision-making processes, databased prediction leads, sales processes, marketing optimization, and the detection of cyber risks in the corporate world. In addition, this research work also defines, interprets, and assesses laborious data used in the study to find anomalies in business intelligence operations.

Along with the rise of online shopping, e-commerce, and e-retail, an unprecedented number of new online users and new routes for online shoppers have created opportunities for fraud and abuse. Figure 1 indicates that there are more credit card scams every year Access to a wealth of current and historical data about the customer and the transactions, including the customer's shopping profile, their connections to other customers, and the locations and methods of prior purchases and returns can help solve most of the Fraud.

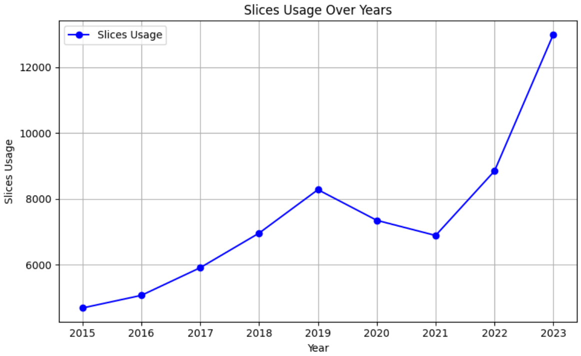

FIGURE 1. Credit card frauds in worldwide.

The primary goal of the study is to identify fraud in banks and strengthen the security of the financial system. Deep learning and business intelligence (BI) are used to stop financial fraud utilizing a combination of preventive measures and detection techniques. Remember that preventing fraud is a constant effort and that updating fraud detection is essential to stay ahead of emerging fraud tactics. The overall contribution of this study is summarized below.

i This study focuses on using different types of deep learning methods on graphs that are helpful for fraud detection in online transactions.

ii This study is mainly based on Fraud Detection with Graph Neural Networks and Anomaly detection using Autoencoders in Credit cards to improve business practices in large organizations and reduce fraudulent activities. Using these two algorithms in business, we discuss the decision-making process, database prediction leads, and sales processes, optimize marketing, and detect cyber threats.

iii For real-time fraud detection, we propose a Graph neural network with lambda architecture. For credit card fraud detection, we use autoencoders and check their validity by taking two Pakistani banks as a case study. This assists the banking system in determining the gap between current operational methods and emerging procedures that utilize deep learning and business intelligence to detect fraud.

iv Finally, we summarized the study to improve efficiency in the domain of business, online fraud, and malicious behaviors. We took the banking system and calculated the fraud detections to enhance the efficiency of the banking system using deep learning. Deep learning with business intelligence offers feasible explanations for the various business domains. By using artificial intelligence with deep learning in business intelligence, we can achieve our targets more efficiently.

This paper is structured as follows: Section II discusses the literature review. Section III presents the theoretical framework of the research work. Section IV evaluates the proposed methodology. Section V presents the evaluation criteria and section VI results and discusses them. Finally, section VII concludes our paper.

II. RELATED WORK

This section gives an overview and discusses the key theories and concepts regarding Business Intelligence, Information Intelligence, and Deep Learning and studies regarding deep learning algorithms like Graph Neural Networks, Autoencoder, etc., to increase the organization's productivity and be helpful for fraud detection.

Bao et al. [7] detected the satisfaction tone using machine learning and deep learning (DL) techniques as its most common issue nowadays. The dataset for this study is collected from social media with Big Data analytics. The results were evaluated using Recall, F1-measure, and Precision measures. Random Forest classifier won the race with 99.1% accuracy.

Mahalle et al. [8] identified the need for machine learning and deep learning algorithms to drive a considerable number of datasets. Google Assistant, Alexa, Uber, and Siri are impressive existing services based on big data. The authors tried to figure out all the possibilities of ML and DL in business intelligence. The big challenge is to design effective techniques to manage a huge amount of data.

Recent advancements in banking fraud detection [9] using deep learning involve the integration of more sophisticated and adaptive algorithms that enhance the accuracy and efficiency of fraud identification. These advanced models can analyze vast datasets in real time, enabling banks to detect evolving and complex fraud patterns. To fight fraud while ensuring a smooth experience, banks combine advanced techniques like behavioral analysis, biometric verification, and deep learning. This comprehensive approach creates a detailed picture of each customer, allowing them to identify fraudsters without interrupting legitimate transactions. Moreover, machine learning systems are increasingly being complemented by big data analytics, enabling banks to harness the power of extensive data sources for more effective fraud prevention [10]. The ability to continually learn and adapt to new fraud tactics positions machine learning as a vital component of modern banking fraud detection systems, ensuring financial institutions can better protect their assets and maintain customer trust in an ever-evolving threat landscape.

The principles of Relational Density Theory were used to introduce a novel Hierarchical Attention-based GNN (HA-GNN) for fraud detection [8]. This network incorporates weighted adjacency matrices across various relationships to counteract fraudulent camouflage. An attention module for assessing the strength of connections between nodes was implemented, along with a neighborhood attention module designed to capture the broader structural similarities present within the graph. The experiments conducted on three real-world datasets illustrate that the proposed method delivers a notable improvement of 3.21% to 9.97% in terms of RUC when compared to the current state-of-the-art techniques. Kanan et al. [11] proved that the graph neural network model achieved higher fraud detection results as compared to the tree-based network but it requires more inference time.

The main purpose of the research is to combine data from information technology, and business intelligence, and use deep learning algorithms collectively to enhance business procedures in sizable firms and decrease fraudulent activity. A successful experiment relies on the discovery of significant patterns in vast amounts of data, which transforms experience into expertise. Bouguettaya et al. [12] introduced an effective framework for large-scale fraud detection driven by graph technology. To elaborate, a heterogeneous label propagation algorithm was introduced to identify additional potentially fraudulent cases for subsequent model training. Next, an innovative multi-view heterogeneous graph neural network model was added to enhance the accuracy of fraud predictions. Lastly, a methodology for uncovering concealed fraud clusters through a fraud pattern analysis approach was introduced. The results achieved by this approach were better than the previous methods. Another deep convolutional neural network (DCNN) [7] model was proposed which aimed at identifying anomalies in patterns generated through competitive swarm optimization (CSO), focusing particularly on detecting fraud scenarios that lack prior detection records or supervision. An unsupervised learning technique, referred to as CSO-DCNN, was introduced, which employs the Rectified Linear Unit (ReLu) to ensure the input and output alignment. The CSO-DCNN is employed for real-time fraud detection using available datasets and is compared against existing algorithms. The experimental outcomes reveal that the proposed CSO-DCNN achieved an accuracy of 98.20%, 99.77%, and 95.23% for credit card, insurance, and mortgage datasets, respectively.

Different deep learning models such as ANN, Support Vector Machine (SVM), Logistic Regression, and Random Forests were compared to determine the best approach for fraud detection [13]. The significance of robust methodologies in uncovering fraudulent transactions and mitigating financial risks was also covered. The implementation of machine learning models for identifying anomalies in banking transactions entails crucial steps; including preprocessing, feature extraction, and model training. The proposed approach achieved 92% precision while the random forests achieved 87% precision. Achary et al. [13] introduced an intelligent system utilizing a deep learning model, which will possess predictive and adaptive capabilities for the identification of fraudulent customer activities in banking transactions.

Another work entitled "A Design Science Research Agenda"by Liu et al. [14] presents a research agenda for using deep learning techniques in business intelligence (BI). The authors introduce BI and deep learning and discuss the advantages and drawbacks of applying deep learning to BI. They then suggest a design science research paradigm for creating and evaluating BI products based on deep learning. Problem identification, solution design, implementation, assessment, and diffusion are the five stages of this framework.

"Credit Card Fraud Detection using Deep Learning Based on Autoencoder and Variational Autoencoder," Liu et al. [15] suggest a technique for detecting credit card fraud using VAEs. They demonstrate that their method produces higher accuracy in detecting fraud by contrasting its performance with existing anomaly detection techniques, such as support vector machines and isolation forests. In "Credit Card Fraud Detection using Autoencoder and Decision Trees," Bukhori et al. [16] propose a method for credit card fraud detection using autoencoders and decision trees. They compare the performance of their method with other machine learning methods, such as random forests and logistic regression, and show that their method achieves higher accuracy in detecting fraud.

III. THEROTICAL FRAMEWORK

A. GRAPH NEURAL NETWORKS

GNNs is a type of neural network specifically designed to work with graph-structured data. It operates by propagating and aggregating information across the nodes and edges of a graph. The key components of a GNN include:

- Nodes (Vertices): Represent entities or data points in the graph.
- Edges: Represent the relationships or connections between nodes.
- Message Passing: Information is passed between nodes along edges, typically involving aggregation of information from neighboring nodes.
- Node Embeddings: Each node is represented by a feature vector that is iteratively updated through message passing.
- Graph-level Readout: After processing, a global representation of the entire graph can be derived from the node embeddings, useful for tasks like graph classification.

B. LAMBDA ARCHITECTURE

Lambda Architecture is a data processing architecture designed to handle massive quantities of data by combining both batch and real-time processing. It consists of three layers:

- Batch Layer: Handles large-scale data processing and computes results over long time intervals. Typically implemented with systems like Hadoop or Spark.
- Speed Layer: Handles real-time data processing, providing low-latency updates to the system using stream processing frameworks like Apache Kafka or Apache Storm.
- Serving Layer: Combines outputs from the batch and speed layers to provide a comprehensive view of the data, enabling query processing.

Integrating GNNs with Lambda Architecture:

i Batch Layer:
  - Training GNNs: In the batch layer, GNNs can be trained on large historical datasets. The training process involves learning node embeddings, edge predictions, or graph-level embeddings from the data.
  - Graph Construction: Large-scale graph construction and preprocessing tasks are performed here. This includes generating and updating graphs based on the historical data.

ii Speed Layer:
  - Real-time Inference: The speed layer can be used to deploy trained GNN models for real-time inference. For instance, as new data arrives, the system can make predictions or update node embeddings in real-time.
  - Streaming Graph Updates: As data streams in, the graph structure can be incrementally updated in the speed layer. This allows the GNN to continuously adapt to new information.

iii Serving Layer:
  - Query Processing: The serving layer provides a unified interface for querying the results. This layer combines the insights from both the batch and speed layers to answer queries. For example, it might use embeddings generated in the batch layer for in-depth analysis, while real-time embeddings from the speed layer handle recent changes.

To use Graph Neural Networks (GNNs) and Lambda Architecture for real-time fraud detection in a banking system, we designed a system that efficiently handles both historical and real-time data to detect fraudulent activities as they occur. Here's a detailed approach:

GNNs can model complex relationships between entities (e.g., transactions, accounts, and devices) and detect suspicious patterns indicative of fraud. Whereas, Lambda Architecture provides a robust framework to handle both batch processing (for historical data analysis and model training) and real-time processing (for immediate fraud detection).

C. ARCHITECTURE DESIGN

Batch Layer:

i Purpose: Analyze historical data to identify patterns and train GNN models for fraud detection.

ii Components:

  - Historical Data Aggregation: Collect and aggregate historical transaction data, account details, and other relevant information to construct a comprehensive graph.
  - Graph Construction: Create a graph where nodes represent entities such as accounts, transactions, devices, and IP addresses, and edges represent relationships or interactions between them (e.g., transactions between accounts).
  - GNN Training: Train GNN models on this historical graph data. The model learns to identify normal and abnormal patterns by analyzing the relationships between entities.
  - Fraud Pattern Detection: Identify known fraud patterns or suspicious subgraphs within the historical data to enhance the GNN model's training.
  - Embedding Storage: Store node embeddings generated by the GNN model, representing learned patterns from the historical data.

Speed Layer:

i Purpose: Process real-time transaction data to detect fraud immediately as it occurs.

ii Components:

  - Real-time Data Ingestion: Stream real-time transaction data (e.g., via Apache Kafka) as it arrives in the system.
  - Graph Updates: Incrementally update the graph with new transactions, creating new nodes and edges as necessary (e.g., adding a new transaction between two accounts).
  - Real-time Inference: Apply the pre-trained GNN model to the updated graph to detect suspicious activities in real-time. The GNN can flag transactions or accounts that deviate from learned patterns.
  - Alerting System: If the GNN identifies a potential fraud, trigger an alert or automatically block the transaction. This can be integrated with a rule-based system for further verification.

Serving Layer:

i Purpose: Combine batch and real-time data to provide a comprehensive and up-to-date view of potential fraud.

ii Components:

  - Query Interface: Provide an interface for querying both historical and real-time data. This interface can be used by analysts to investigate flagged transactions or review the effectiveness of the fraud detection system.
  - Model Updates: Periodically update the GNN model in the batch layer based on new patterns detected in the speed layer. This ensures the model evolves with new fraud techniques.
  - Decision Engine: Integrate the outputs from both batch and speed layers to make informed decisions about transactions (e.g., accepting, flagging for review, or rejecting).

D. IMPLEMENTATION WORKFLOW

i Historical Data Processing:

  - Collect past transaction data and build a large-scale graph in the batch layer.
  - Train the GNN model to identify potential fraud based on historical patterns.

ii Real-time Data Processing:

  - Stream new transactions into the system using a real-time data ingestion pipeline.
  - Update the graph with new transactions and apply the GNN model for real-time fraud detection.

iii Alerting and Decision Making:

  - When a suspicious transaction is detected, trigger alerts, and possibly block or flag the transaction for manual review.
  - Continuously monitor and refine the system using feedback from real-time detection results.

iv Model Retraining and Evolution:

  - Periodically retrain the GNN model with new data to capture emerging fraud techniques and improve detection accuracy.

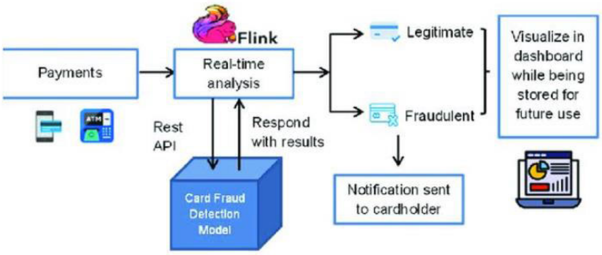
Flink
Payments
Real-time analysis
Legitimate
Fraudulent
Visualize in dashboard while being stored for future use
Rest API
Respond with results
Card Fraud Detection Model
Notification sent to cardholder

FIGURE 2. Architecture of GNN fraud detection.

E. CHALLENGES AND CONSIDERATIONS

i Scalability: Managing and processing large graphs in real-time can be computationally intensive. Efficient graph storage and incremental updating mechanisms are crucial.

ii Latency: Ensuring low-latency inference in the speed layer is critical for real-time fraud detection. Optimizing the GNN model for fast inference is important.

iii Accuracy: Balancing false positives and false negatives in fraud detection is challenging. The model should be continuously evaluated and updated to improve its accuracy.

A GNN could be trained on historical credit card transactions to detect fraud patterns, such as unusual spending behavior or transactions from unlikely locations. In real-time, the system could monitor new transactions and flag any that match suspicious patterns, allowing immediate intervention. This approach enables a banking system to effectively detect and respond to fraudulent activities in real-time while continuously learning from new data.

Deep neural networks that operate on network data include graph neural networks. By combining node attributes from nearby nodes, the neural network creates a vector representation for each node, known as node embedding. Subsequently, the classification task, such as fraud detection, is carried out using node embedding. This is how GNN algorithms learn the node features and the graph topology shown in figure 2.

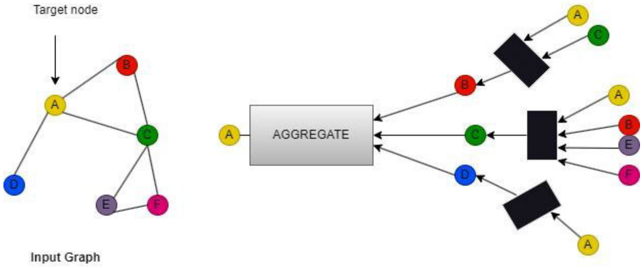

FIGURE 3. **A simple input graph containing 6 nodes and 7 edges.**

F. AUTOENCODERS

A type of neural network called an autoencoder is employed in unsupervised learning tasks. Encoder and Decoder are the two neural network components that they have. When high-dimensional data represent in a low-dimensional space, like PCA, we use autoencoders because traditionally it used for dimensionality reduction. Whereas PCAs cannot decode data from a high-dimensional non-linear manifold into a low-dimensional space due to their linearity limitations. The general formula for defining an autoencoder is $\boldsymbol{f(x) = h}$, where $\boldsymbol{h}$ denotes the latent variables in information bottleneck and $\boldsymbol{x}$ is the input data. The encoder portion of the network is indicated by this formula. The decoder converts the latent variables from the information bottleneck into an output that can be represented by the notation $\boldsymbol{g(h) = x'}$. The decoder is often the encoder's mirror image shown in figure 4. The whole autoencoder can be described as:

$$L(x, g(f(x))),$$

Here, both f and g are nonlinear functions.

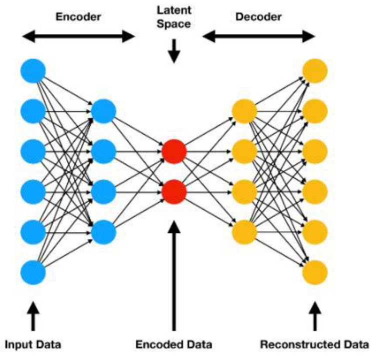

FIGURE 4. **Autoencoders work.**

Mathematically, the encoder can be represented as:

$$z = f(Wx + b)$$

Mathematically, the decoder can be represented as:

$$\hat{x} = g\left(W'z + b'\right),$$

Autoencoder has the power to encode the dataset into a subspace and decode the feature back while normalizing the data. Anomalies are easy to detect with autoencoders, but the input and the output will be quite different. Autoencoders encode the value x using a function f and then decode the encoded value using a function g to create an identical value as the input value. $f(W, b(x)) \approx x$

Reconstruction error = input vector – output vector

$$L(x, x') = ||x - x'||2$$

The main objective of the Autoencoders is to minimize the reconstruction error among input and output values shown in figure 5. The network topology in an autoencoder has connections between the layers but none within them; layer x_i corresponds to the input sample, and layer $\hat{x}_i$ to the feature. Using the provided samples, the autoencoder neural network is trained to minimize reconstruction error. The project's definition of the autoencoder neural network's cost function is.

$$J_{A,E} = \frac{1}{m}\sum_{i=1}^{m}(\frac{1}{2}\parallel \hat{x}_i - x_i \parallel^2)$$

By reducing the amount of information passing through the network and limiting the capacity of the model as much as feasible, under complete autoencoders seek to map input x to output x'. Using the same loss function as a learning tool, under complete autoencoders can discover features:

$$L(x, g(f(x))),$$

L is the loss function that punishes $g(f(x))$ for deviating from the initial input $x.L$ could represent a mean absolute error or even a mean squared error. To solve the issues with under complete autoencoders, whereas regularized autoencoders construction is based on the complexity of the data.

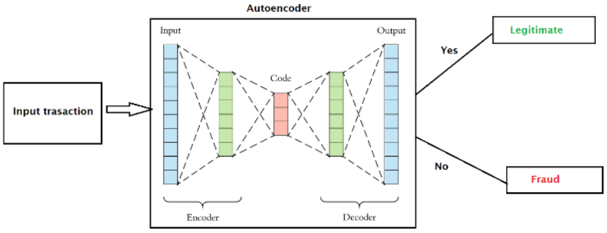

FIGURE 5. Architecture of autoencoders.

The capacity of the encoder, decoder, and information bottleneck can all be increased. They become more powerful and adaptable as a result. Regularized autoencoders having a penalty on the hidden layer and a reconstruction loss are called sparse autoencoders.

$$L(x, g(f(x)) + \Omega(h))$$

whereas h here stands for buried layers. The main purpose of the autoencoder training is to reduce the discrepancy between input and reconstruction data. This is often accomplished by employing optimization methods like stochastic gradient descent (SGD) or its variations to minimize the loss function.

G. MODEL ARCHITECTURE

The attribute embedding look-up and feature learning layer, which incorporates feature aggregation with a multi-layer perception (MLP), initially learns the raw properties of transaction data. The card's attributes in our solution include its kind, cardholder type, card limit, remaining limit, etc. The channel ID, currency ID, transaction amount, etc., are only a few examples of the transaction properties. The merchant attributes include information about the merchant's kind, terminal type, location, industry, charge ratio, etc. Then, to gather and understand the significance of past transaction embeddings, we developed a gated temporal attention network.To extract the likelihood of fraud from these representations, we then use a two-layer MLP. The entire model can be optimized using the current stochastic gradient descent technique in an end-to-end process shown in figure 6.

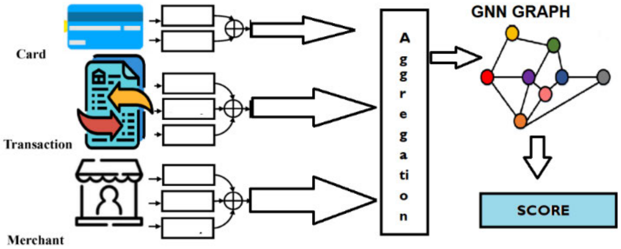

FIGURE 6. Model architecture.

Each record in the given transaction data includes card attributes. We do not exclude any cards or merchants from preprocessing because they have few authorized transaction records. We employ entire user transaction records to keep track of all potential frauds because there are far more cards and merchants than are manually examined. Then, for each record, we create the numerical attribute representation.

$$Y_{num} \epsilon R^{N_{xd}}$$

where N stands for the quantity of transactions and d for the feature dimensions.

$$e_{attr} = onehot(f_{attr}) \odot E_{attr}$$

We aggregate these embeddings to get the category embedding for each transaction after collecting the embedding vector for each attribute in the card, transaction, and merchant databases.

$$Y_{cat}^{(u)}$$

$$= \sum_{i}^{ca} y'_{c^{(u)}}, i \in \{card, trans, mchnt\}, \quad wherey_{cat}^{(u)} \epsilon RI \times d$$

Our suggested feature learning layer can represent all categorical attributes and project them to a single spatial dimension, which is advantageous for our attribute-driven graph learning model because it addresses the variability of categorical attributes. Hence, we create the temporal transaction graph and aggregate messages on this network to update the embedding of each transaction to learn the temporal fraud trends. The directed temporal edges are produced with the goal being the current transactions and the source being the prior transactions. The Temporal Graph Attention (TGAT) technique is then used to aggregate messages [17]. One of the hyper-parameters that will be researched in the experiment part is the quantity of created temporal edges per node.

We use various transaction embeddings to become familiar with each transaction record's temporal embedding. First, as the input to the GTAN network, we blend category attributes and numerical attributes.

$$Y = \{y_{t_0}, y_{t_1}, \ldots\ldots . y_{t_n}\}$$

with

$$y_{t_i} = y_{(t_i)_{num}} + y_{(t_i)_{cat}}$$
$$H_0 = Y$$

Then, we use multi-head attention to calculate each neighbor's relevance separately and update embedding, which can be stated as follows:

$$H = Concat(Head_1, \ldots\ldots . Head_{hatt})W_{\circ}$$

where h_{att} denotes the number of heads, $W_{\circ}$ shows learnable parameters, H denotes the aggregated embedding's with H,

$$Head = \sum\nolimits_{y_i \in \chi} \sigma(\sum\nolimits_{y_t \in \mathcal{N}(y_i)} \alpha_{y_t, y_i} y_t),$$

$$\alpha_{y_t, y_i} = \frac{\exp\left(LeakReLU\left(a^T\left[y_t \| y_i\right]\right)\right)}{\sum\nolimits_{y_j \in \mathcal{N}(y_t)} \exp\left(LeakyReLU\left(a^T\left[y_t \| y_i\right]\right)\right)}$$

where $N(y_i)$ denotes the temporal neighbors of the i^{th} transaction, $\alpha y_t, x_i$ denotes the importance of temporal edge (y_t, x_i) in each attention head, and $a \in R2d$ defines the weight vector of each head. Additionally, to mimic the temporal fraud pattern through message passing on the temporal transaction graph without stealing future information, the neighbor transactions sampled for each transaction must be the previous transactions from the same cardholder. After obtaining aggregated embedding, we use the embedding and raw attributes to infer the significance of the aggregated embedding and raw attributes after each layer of TGAT. This formulation can be used to further increase the effectiveness and interpretability of our method.

$$gate_{t_i} = \sigma\left(\left[y_{cat}, t_i \,||y_{num}, t_i||\, h_{t_i}\right] \beta_{t_i}\right),$$
$$z_{t_i} = gate_{t_i}.h_{t_i} + \left(1 - gate_{t_i}\right).y_{t_i}$$

where $gate_{t_i}[0, 1]$ stands for the $t_i - t_h$ transactions gate variable. According to our paradigm, the output of the k^{th} gating mechanism is used as the input of the $(k+1)^{th}$ TGAT if a new TGAT layer is stacked on top of the attribute-driven gated residual mechanism. We can accomplish both attribute propagation and label propagation using a single graph neural network. As a result, by including the transaction label as one of the transaction categorical features, our fraud detection model can jointly predict temporal fraud patterns and risk propagation.

H. FRAUD RISK PREDICTION

We use a two-layer MLP to estimate the fraud risk after getting the aggregated embedding of transactions. The formulation is as follows:

$$\hat{y} = \sigma(PReLU\,(HW_0 + b_0)\, W_1 + b_1)$$

$\hat{y}$ represents the results of all transactions' risk prediction. Where W and b are the parameters. We semi-supervise our model, in contrast to earlier credit card fraud detection methods, by propagating transaction attributes and risk embedding across labelled and unlabeled transactions. Our fraud detection model will experience label leaking during training if we use an unmasked objective. In this scenario, our model will just use the observed labels and ignore the intricate hidden fraud patterns that cannot be broadly applied to the prediction of upcoming fraud transactions. Therefore, rather than learning from the label of the transaction itself, we suggest learning from the risk information of the transaction's neighbors. Utilized specifically is a training approach for masked fraud detection.

IV. PROPOSED METHODOLOGY

Proposed methodology of this study is focused on detecting credit card fraudulent transactions in Pakistani banks by using unsupervised Deep Learning algorithms such as graph neural network and autoencoder. The proposed architecture of the study comprises of preprocessing the dataset, which includes partitioning and normalizing the skewed data, training deep learning models, and finally, obtaining predictions. The evaluation metrics are used to assess the performance of different classifiers. A reliable and efficient methodology for identifying credit card fraud is proposed, and as a result, their frameworks are shown in Figure 7 below.

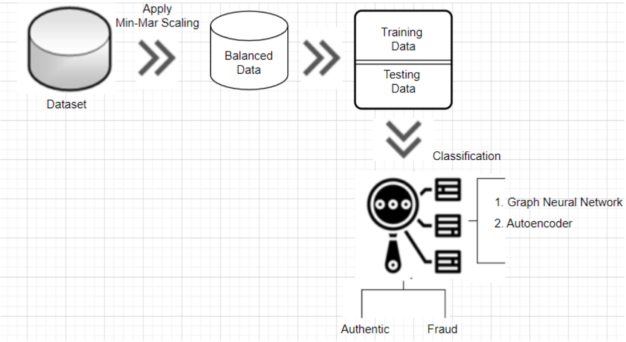

FIGURE 7. Architecture of proposed methodology.

- A real-world dataset of credit card holders from Pakistani private banks (Mazeen and UBL), is use in this study in which 492 out of 284,807 total transactions are fraudulent.
- By using the Min-Max scaling technique we handle the credit card imbalance.
- We obtain a balanced dataset with an equal ratio of fraudulent and nonfraudulent/legitimate transactions after extraction.
- Next, separate the dataset into training and testing data.
- Graph Neural Network and Autoencoder deep learning techniques are being tested to improve the performance of the proposed framework.
- In the end, employ the F1 score, accuracy, recall, and precision to evaluate comparative analysis.

Precision-recall, F1-Score, and accuracy are used to find the performance metrics to evaluate the significance of this study's proposed methodology. A balanced skewed dataset is employed for this. As we use Deep Learning techniques to evaluate and ensure the effectiveness and classification. This study uses a very unbalanced synthetic dataset to detect credit card fraud; this may add a unique dimension to the research. This research aims to fill and identify substantial gaps in the existing corpus of knowledge. The following stages are illustrated in Figure 8. for the deployment of Deep Learning algorithms to obtain accurate predictions for fraud detection.

A. DATASET PERPARTION

The dataset is taken from two Pakistani private banks (Meezan Bank and UBL) for credit card transactions used in this study. This dataset includes the credit card transactions that cardholders completed over the course of one year from 1st January 2022 to 1st December 2022. Out of a total of 284,807 transactions, this dataset has 492 false transactions, indicating a severe imbalance. The "class" feature in the dataset indicates whether a transaction is fraudulent

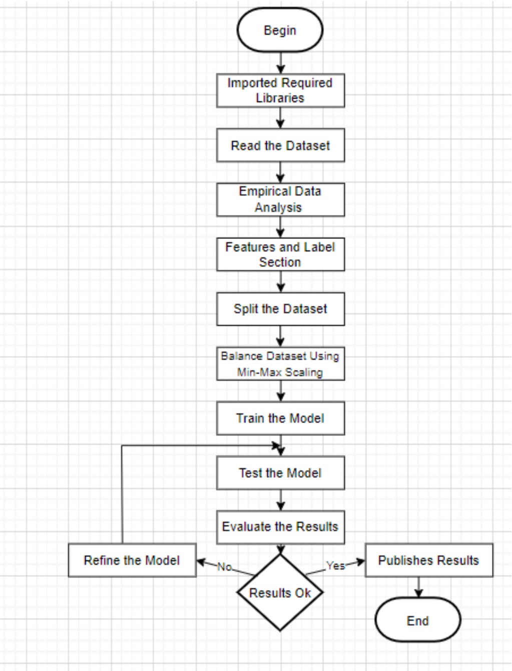

FIGURE 8. Execution process of the models.

or not; a value of 0 indicates that the transaction is not fraudulent, while a value of 1 indicates that it is. Due to the significantly lower ratio of fraudulent transactions than usual, the dataset was extremely unbalanced. As a result, preparing the data was necessary to get better performance outcomes.

In this study, the GNN model is used to learn the independent representation of nodes, so we mainly deal with the inductive unsupervised approach. For fraud detection, it's quite important to deal with the dynamic graphs to deal with the abnormal behavior of fake reviews or something. A one-year of data from Pakistani private banks (Meezan Bank and UBL) is used in this study. The first 11-month data, from 1st January 2022 to 1st November 2022, is used for testing, and the last month, 1st November 2022 onward to 1st December 2022 data is used for training the model. The first 10 days of sample data are given as a reference in Table 1. The Transaction ID, Date, Time, Amount, Merchant, Cardholder Name, and Card Number are just a few of the columns included in this study. The dataset contains transactions from various businesses, cardholders, and amounts, and each row represents a transaction that took place from 1st January 2022 onward. A few entries of our data are shown below:

1) PREPROCESSING DATASET

Because of the extreme imbalance in the dataset, the Min-Max Scaling method (A normalization technique for deep learning) is used here to overcome the imbalance issue of our dataset. Hence, Table 2 displays the dataset with a balanced distribution of fraudulent and non-fraudulent/legitimate transactions.

TABLE 1. Sample of dataset credit card transction.

Transaction ID	Date	Time	Amount	Merchant	Cardholder Name	Card Number
1	2022-01-01	10:23:45	250.00	XYZ Store	Khalid Ahmad	************2334
2	2022-01-02	15:46:12	75.50	ABC Market	Majid Ahmad	************5178
3	2022-01-02	18:30:05	120.00	XYZ Store	Sajid Ahmad	************6234
4	2022-01-03	09:15:30	500.00	Online Shop	Naila Siddique	************8876
5	2022-01-04	14:20:55	80.25	XYZ Store	Ramzan Khan	************2234
6	2022-01-05	19:35:10	300.00	ABC Market	Muhammad Sidiq	************5678
7	2022-01-06	08:10:20	150.00	Online Shop	Amjad Hussain	************9176
8	2022-01-06	12:40:15	95.75	XYZ Store	Khalid Hussian	************1231
9	2022-01-07	16:55:35	400.00	ABC Market	Najma Nadeem	************2678
10	2022-01-08	11:25:50	175.50	XYZ Store	Fahad Hussian	************1534

TABLE 2. Dataset normalization by using min-max scaling.

Results	Before Normalization	After Normalization
Fraudulent Cases	1.90%	50%
Legitimate Cases	98.10%	50%

B. IMPLEMENTATION

Python packages like Pandas, NumPy, Matplotlib, Seaborn, Sklearn, and Imblearn are used to implement detection methods. Detecting credit card fraud is a classification problem; that is, a transaction's fraudulent or authenticity is represented by its class feature in the dataset. In this study, unsupervised deep learning algorithms are implemented. The Anaconda platform in Jupyter is used for this experiment. After implementing graph neural networks and autoencoders, we compare them by using evaluation criteria

V. EVALUATION CRITERIA

A. CASE STUDY 1; FRAUD DETECTION WITHIN WORK FOR FRAUD DETECTION WITH GRAPH NEURAL NETWORK IN CREDIT CARDS

Credit card fraud detection is a crucial area of research in financial fraud detection. Credit card fraud is a catch-all phrase for the unauthorized use of funds in a transaction, generally using a credit or debit card [16]. Fraud detection with graphs is easy as we can read the patterns, such as node aggregation, when a lot of suspicious accounts commence to act in tandem. GNN has just become a prevalent mechanism in mining graph data. GNN is a neural network that aggregates the neighbours' information for downstream tasks like link prediction, node classification, and outlier detection (OD). To detect credit card fraud, we define the problem as an unsupervised node classification and describe credit card behaviours' as a temporal transaction graph. To detect credit card fraud, we introduce a brand-new attribute-driven temporal graph neural network. To extract temporal and attribute data, we specifically suggest a gated temporal attention network. To take advantage of both labelled and unlabelled data, in this study we pass attributes and risk information on the temporal transaction graph.

1) GNN-BASED FRAUD DETECTION CYCLE

- Building the Graph
- Training the GNN on the Graph
- Classifying Unlabeled Nodes

Here in this study, we mark transaction 1 as fraudulent if the cardholder reports it or if financial professionals determine it to be such; otherwise, we mark it as 0. We assess the experimental outcomes on datasets for opinion fraud and credit card fraud detection using the following metrics: averaged precision (AP), F_{1macro}, and the area under the ROC curve (AUC). We keep track of how many True Positives, False Positives, and False Negatives are there.

$$F_{1macro} = \frac{1}{l}\sum\nolimits_{i=1}^{l} \frac{2 \times P_i \times R_i}{P_i + R_i}$$

$$AP = \sum_{i=1}^{l} (R_i - R_{i-1})P_i$$

For four sets of experiments, we adjust the percentages of nodes utilized for training from 10% to 80% with a 10% step-up, using the remaining nodes as the test set in each set of trials. As the dataset from Meezan and UBL banks are completely annotated, hence therefore the findings of unsupervised experiments with different ratios of label training dataset are shown in figure 9 below.

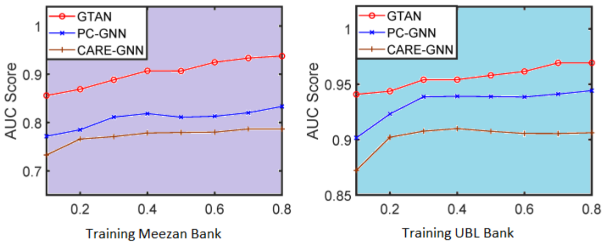

FIGURE 9. Findings of unsupervised experiments with different ratios of labeled training dataset.

In this study, the dataset of Meezan Bank shows that GTAN consistently outperforms other training ratios. Whereas GTAN performs effectively in situations where there is only a 10% training ratio of labelled data, and their performance improves where the dataset is labelled. Similarly, GTAN for the UBL bank dataset also shows consistently outperforms other training ratios. By comparing the GTAN model of UBL bank and Mazeen bank, it is shown in Figure 10 above that the dataset of UBL bank is less sensitive when adjusting the training ratio with no more than 2% fluctuations in the AUC of Meezan bank dataset. Therefore, this effectively proves that GTAN can perform well even with a small amount of labelled dataset ratio.

B. ABLATION STUDY

In banking fraud detection, combining GTAN-A, GTAN-R and GTAN models for feature extraction shows a powerful approach. GTAN works by collecting info from nearby nodes, and repeatedly building detailed node descriptions by blending characteristics from neighboring nodes. On the other hand, Graph Neural Networks (GNNs) use mathematical filters to grasp how nodes relate in a graph. When the results are merged from both GTAN and GNN models, a thorough mix of features that grasp both the local and global features in a financial transaction graph improves the performance of the proposed model. These combined features help fraud detection systems to detect patterns in banking transactions more reliably. The comparison of all the models is also shown in Figure 10.

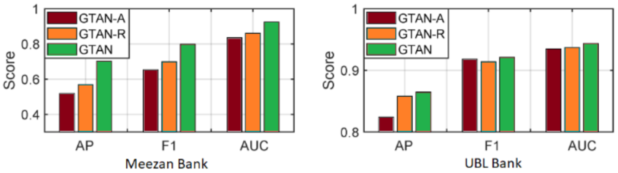

FIGURE 10. The ablation studies.

From the above, we conclude that when the AUC and AP peak, our model performs best with two GNN layers. Thus, we set the default depth to 2. We applied a propagation mechanism to accurately mimic the fraud patterns. Extensive studies showed that our suggested techniques outperformed other baselines in three fraud detection datasets.

C. CASE STUDY (2) ANOMALY DETECTION IN CREDIT CARD FRAUD USING AUTOENCODER

Anomaly detection detects a transaction tumble outside the scope of normal activity. For this study, we use the Autoencoder to check the features of normal transactions and detect unexpected data. For that, we gathered a highly unbalanced dataset that occurred in two months and applied the deep learning algorithm to check the fraud. While applying the graphs for the dataset, normal transactions usually cluster in a disk, while fraud transactions are more distributed. Financial institutions and their clients are very concerned about credit card fraud. The use of autoencoders, a class of neural networks that can learn to encode and decode data, is one method that we use to detect credit card fraud. When autoencoders are trained, they learn to reconstruct the input data from a compressed representation. Information like transaction amount, location, time, and other pertinent facts may be included in the input data for credit card transactions. A sizable dataset of credit card transactions, including both fraudulent and valid ones, is supplied to the autoencoder during training. To capture the underlying patterns in the input data while avoiding noise and outliers, the autoencoder must know a compressed representation of the data. Once the trained autoencoder can be used to spot credit card fraud by contrasting the output from reconstruction with the original input. The transaction may have been fraudulent if the reconstructed result significantly differs from the original input.

1) DATASET DESCRIPTION

The dataset was again obtained by a private bank (MEEZAN BANK) in Pakistan. It has 26 attributes. But we have used only three attributes from the dataset.

- Time
- Amount involved in transaction.
- Output status (1 *or* 0)

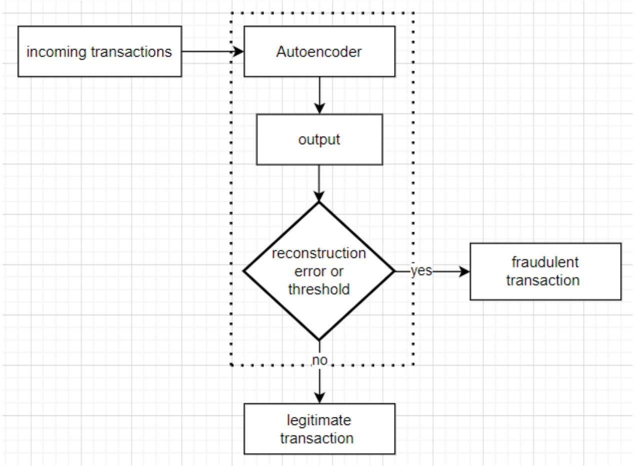

FIGURE 11. Architecture of auto-encoder dataset.

We used 80% of the data for training and 20% of the data for testing. Training the Autoencoder is going to be a bit different from the traditional one. Autoencoders are neural networks composed of multiple layers to decode the data back to the original smaller-size input. Training an Autoencoder on data without anomalies is the hypothesis that comprehends the underlying structure of a given dataset by knowing the dimensions that are important for reconstruction [18]. The Autoencoders reconstruct each dimension to use a neural network to duplicate the input, but during the replication procedure, the size of the input is decreased into its smaller representation. The middle layers have fewer units as compared to other layers. Therefore, by adjusting the hyperparameters and the model architecture, we enhance the accuracy of the model.

We are interested in anomalies in every new transaction. Therefore, by adjusting our training model on typical transactions, we can simulate these cases. We can assess the effectiveness of our model by reserving the appropriate class on the test set. As shown in Figure 14 20% of our dataset is set aside for testing and coding of building a model and reconstruction error with Fraud.

In Figure 15, high sensitivity and specificity are both represented by an area under the curve (AUC), whereas high sensitivity is correlated with a low false negative rate, and high specificity is correlated with a low false positive rate. When both metrics have high scores, classifiers successfully detect the most positive occurrences (high sensitivity) and correctly eliminate the most negative examples (high specificity). This shows that the classifier produces accurate and thorough findings.

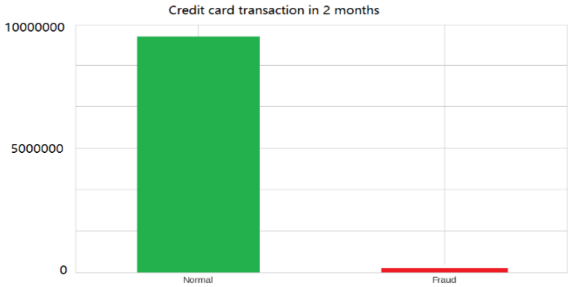

FIGURE 12. Transaction class distribution.

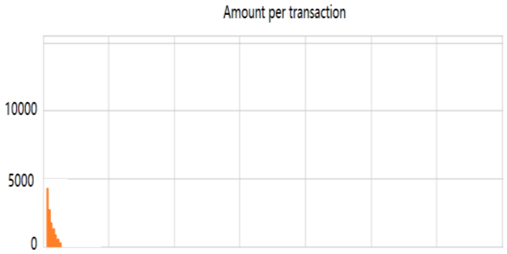

FIGURE 13. Amount transaction by classes.

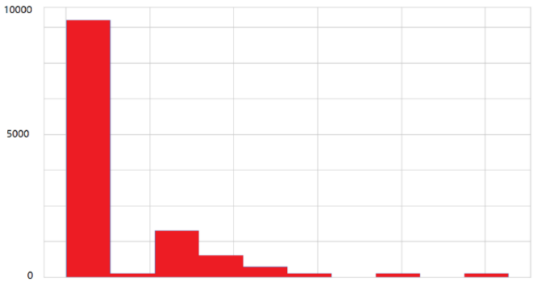

FIGURE 14. Reconstruction error with fraud.

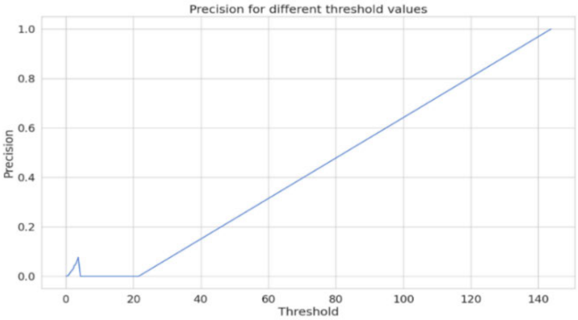

FIGURE 15. Precision and threshold.

The model becomes more conservative in predicting positive events as the classification threshold is raised. As a result, there are fewer occurrences that are anticipated to be positive, which lowers the number of both true and false positives. Hence the ratio of true positives to the total of true positives and false positives determines precision, a drop in anticipated positives (true positives plus false positives) while the genuine positives remain the same or decline would result in a drop-in precision.

VI. RESULTS AND DISCUSSION

A. CONFUSION MATRIX

Measuring the performance of classifiers is crucial for their design. Classification algorithms' performance is measured by their ability to accurately categorize samples of data. To evaluate classifier performance, predictions are classified as True Positive (TP), True Negative (TN), False Positive (FP), or False Negative (FN).

TABLE 3. Confusion matrix evaluation.

		Predicted Classes	
		Positive	Negative
Actual Classes	Positive	TP	FN
	Negative	FP	TN

As in figure 16 by confusion matrix for the training and testing models, some excellent results were obtained. The model successfully recognized all 306 fraudulent transactions and 2694 valid ones out of 3000 transactions, of which 306 were fraudulent and 2694 are authentic. This means it classified seven nonfraudulent transactions as fraudulent. This suggests that the classification was right in 99.76% of the cases (3000-7) / 3000. In the model tested, shown in figure 17 out of 2000 transactions, 217 were fraudulent and 1783 were genuine, the model correctly detected all 217 fraudulent transactions and 1778 genuine transactions. This means that it regarded five non-fraudulent transactions as fraudulent. This means that in the percentage of (2000-5) / 3000 = 99.75% the classification was correct.

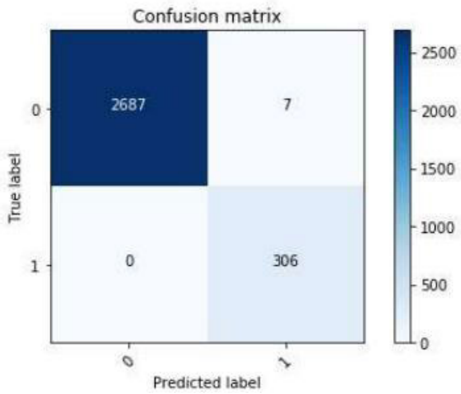

FIGURE 16. Confusion matrix for training dataset.

The resulting model is a good one, because the proportion of the correct classification is very high. It is also important to note that all fraudulent transactions have been correctly identified. By using this, even if certain payments are falsely categorized as fraudulent, a person can check again manually to confirm or deny the model result. From our perspective, it is much safer for the model to make mistakes by classifying some transactions as fraud even if they are not, rather than classifying fraudulent transactions as real, because in the latter case, there will either be unidentified frauds, or all transactions that have been detected as real should be checked again by another method.

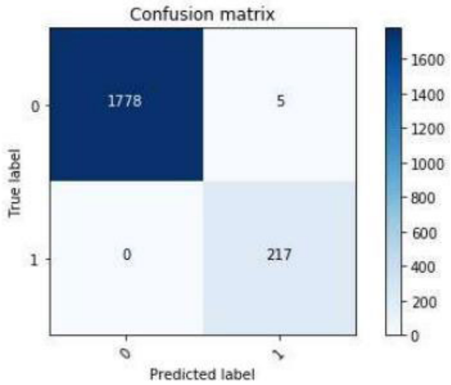

FIGURE 17. Confusion matrix for testing dataset.

The GNN model discussed in the previous section is trained on the Mazaan and UBL banking fraud detection datasets. The results of the GNN model for Mazaan and UBL bank is presented in table 3.

TABLE 4. Results of GNN model on Mazaan and UBL bank dataset.

GNN model	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
Mazaan bank	78.06	81.63	80.62	81.36
UBL bank	78.01	80.98	79.68	80.78

From Table 4, the present the model achieves accuracy, precision, recall, and F1 score of Mazaan and UBL bank in Pakistan. However, the performance of GNN is less in terms of accuracy which cannot be used in the real world.

Autoencoders are a type of neural network architecture used in unsupervised learning, falling under the category of generative models. The primary purpose of autoencoders is to learn an efficient representation of input data, typically for dimensionality reduction or feature learning.

From Table 5, the present the model achieves accuracy, precision, recall, and F1 score of Mazaan and UBL bank in Pakistan. However, the performance of Autoencoder model is less in terms of accuracy which cannot be used in the real world.

TABLE 5. Results of autoencoder model on Mazaan and UBL bank dataset.

Autoencoder model	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
Mazaan bank	77.06	80.63	81.62	80.36
UBL bank	79.01	79.98	80.68	79.78

VII. CONCLUSION

In the end, we conclude that deep learning can substitute humans, but many researchers are still not confident that robots can be instructed like humans. Globally, financial institutions are using deep learning with business intelligence to control money laundering. Our research work is based on Fraud Detection with Graph Neural Networks and Anomaly detection by using Autoencoders in Credit cards. The results of our research work help many financial organizations, especially banks, in achieving an adequate understanding of deep learning with business intelligence. Our study in future also assists the banks in Pakistan to determine the gap between current operational methods and emerging procedures that utilize deep learning and business intelligence to detect fraud. To detect credit card fraud, we define the problem as a semi-supervised node classification job and describe credit card behaviors as a temporal transaction graph. By using a brand-new attribute–driven temporal graph neural network, we can detect credit card fraud. For extracting temporal and attribute data, we specifically suggest a gated temporal attention network. So, by taking advantage of both labelled and unlabeled data, we pass attributes and risk information on the temporal transaction graph.

To improve overall performance, autoencoders can be used in conjunction with other fraud detection strategies. For instance, to develop a complete fraud detection solution, rule-based systems, machine learning classifiers, or network analysis algorithms can be combined with anomaly detection utilizing autoencoders. It is feasible to create a fraud detection system using a deep learning strategy, and it is possible to minimize the work required to manually label a dataset using both supervised and unsupervised learning approaches. During the model update, there is a possibility to integrate the identification of novel fraud types. The method for updating the model relies on the amount of labelled data. In our research work we use Python for the analysis purpose to boost the efficiency of the business organization. Hence, in the end, we conclude that deep learning with business intelligence offers feasible explanations for the various business domains, and by using artificial intelligence with deep learning in business intelligence, we can achieve our targets more efficiently.

DISCLOSUER STATEMENT

No potential conflict of interest was reported by the authors.

REFERENCES

[1] P. Tiwari, S. Mehta, N. Sakhuja, J. Kumar, and A. K. Singh, "Credit card fraud detection using machine learning: A study," 2021, *arXiv:2108.10005*.

[2] T. Micro, "Deep security? Software," Tech. Rep., 2020.

[3] G. K. Kulatilleke, "Challenges and complexities in machine learning based credit card fraud detection," 2022, *arXiv:2208.10943*.

[4] S. K. Hashemi, S. L. Mirtaheri, and S. Greco, "Fraud detection in banking data by machine learning techniques," *IEEE Access*, vol. 11, pp. 3034–3043, 2023.

[5] R. B. Sulaiman, V. Schetinin, and P. Sant, "Review of machine learning approach on credit card fraud detection," *Hum.-Centric Intell. Syst.*, vol. 2, nos. 1–2, pp. 55–68, 2022.

[6] E. Btoush, X. Zhou, R. Gururaian, K. Chan, and X. Tao, "A survey on credit card fraud detection techniques in banking industry for cyber security," in *Proc. 8th Int. Conf. Behav. Social Comput. (BESC)*, Oct. 2021, pp. 1–7.

[7] Y. Bao, G. Hilary, and B. Ke, "Artificial intelligence and fraud detection," in *Innovative Technology at the Interface of Finance and Operations*, vol. 1, 2022, pp. 223–247.

[8] A. Mahalle, J. Yong, X. Tao, and J. Shen, "Data privacy and system security for banking and financial services industry based on cloud computing infrastructure," in *Proc. IEEE 22nd Int. Conf. Comput. Supported Cooperat. Work Design ((CSCWD))*, Nanjing, China, May 2018, pp. 407–413, doi: 10.1109/CSCWD.2018.8465318.

[9] N. Karkashadze, G. Shanidze, M. Shalamberidze, and S. Mikabadze, "Modern challenges in agribusiness," *Int. J. Innov. Technol. Economy*, vol. 2, no. 38, Jun. 2022, doi: 10.31435/rsglobal_ijite/30062022/7828.

[10] F. Manessi, A. Rozza, and M. Manzo, "Dynamic graph convolutional networks," 2017, *arXiv:1704.06199*.

[11] T. Kanan, A. Mughaid, R. Al-Shalabi, M. Al-Ayyoub, M. Elbes, and O. Sadaqa, "Business intelligence using deep learning techniques for social media contents," *Cluster Comput.*, vol. 26, no. 2, pp. 1285–1296, Apr. 2023, doi: 10.1007/s10586-022-03626-y.

[12] A. Bouguettaya, H. Zarzour, A. Kechida, and A. M. Taberkit, "Machine learning and deep learning as new tools for business analytics," in *Handbook of Research on Foundations and Applications of Intelligent Bus. Analytics*, 2022, doi: 10.4018/978-1-7998-9016-4.ch008.

[13] Y. Liu, Z. Sun, and W. Zhang, "Improving fraud detection via hierarchical attention-based graph neural network," *J. Inf. Secur. Appl.*, vol. 72, Feb. 2023, Art. no. 103399, doi: 10.1016/j.jisa.2022.103399.

[14] H. A. Bukhori and R. Munir, "Inductive link prediction banking fraud detection system using homogeneous graph-based machine learning model," in *Proc. IEEE 13th Annu. Comput. Commun. Workshop Conf. (CCWC)*, Mar. 2023, pp. 246–251, doi: 10.1109/CCWC57344.2023.10099180.

[15] Z. Li, B. Wang, J. Huang, Y. Jin, Z. Xu, J. Zhang, and J. Gao, "A graph-powered large-scale fraud detection system," *Int. J. Mach. Learn. Cybern.*, vol. 15, no. 1, pp. 115–128, Jan. 2024, doi: 10.1007/s13042-023-01786-w.

[16] T. Karthikeyan, M. Govindarajan, and V. Vijayakumar, "An effective fraud detection using competitive swarm optimization based deep neural network," *Measurement, Sensors*, vol. 27, Jun. 2023, Art. no. 100793, doi: 10.1016/j.measen.2023.100793.

[17] R. Jain and S. Deshwal, "Anomaly detection in bank transactions using machine learning," Tech. Rep.

[18] R. Achary and C. J. Shelke, "Fraud detection in banking transactions using machine learning," in *Proc. Int. Conf. Intell. Innov. Technol. Comput., Electr. Electron. (IITCEE)*, Jan. 2023, pp. 221–226, doi: 10.1109/IITCEE57236.2023.10091067.

[19] R. Koldehofe, J. Treder, and A. Wagenknecht, "A design science research agenda," in *Proc. 15th Int. Conf. Wirtschaftsinformatik*, vol. 1, 2019, pp. 1378–1392. [Online]. Available: https://aisel.aisnet.org/wi2019/1378/

[20] J. H. Kim, H. Y. Kim, and Y. H. Kim, "Credit card fraud detection," Tech. Rep., 2020.

[21] J. Lewandowski and M. Ossowski, "Non-singular extension of the Kerr-NUT-(anti) de sitter spacetimes," 2021, *arXiv:2101.05802*.

[22] S. Bhattacharyya, S. Jha, K. Tharakunnel, and J. C. Westland, "Data mining for credit card fraud: A comparative study," *Decis. Support Syst.*, vol. 50, no. 3, pp. 602–613, Feb. 2011.

[23] S. Xiang, D. Cheng, C. Shang, Y. Zhang, and Y. Liang, "Temporal and heterogeneous graph neural network for financial time series prediction," in *Proc. 31st ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2022.

[24] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P. A. Manzagol, and L. Bottou, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *J. Mach. Learn. Res.*, vol. 11, pp. 3371–3408, Dec. 2010.

[25] X. Fan, C. Xiang, L. Gong, X. He, Y. Qu, S. Amirgholipour, Y. Xi, P. Nanda, and X. He, "Deep learning for intelligent traffic sensing and prediction: Recent advances and future challenges," *CCF Trans. Pervas. Comput. Interact.*, vol. 2, no. 4, pp. 240–260, Dec. 2020, doi: 10.1007/s42486-020-00039-x.

FAWAZ KHALED ALARFAJ received the M.Sc. and Ph.D. degrees in computer science from Essex University, U.K. He is currently an Associate Professor with the Department of Management Information Systems, King Faisal University. With a robust academic foundation and expertise, he continues to make significant contributions to the advancement of computer science. His research interests include information retrieval, natural language processing, machine learning, and artificial intelligence, focusing on its applications in bioinformatics and cybersecurity.

SHABNAM SHAHZADI received the master's degree in statistics from Arid Agriculture University, Rawalpindi, Pakistan. She is currently pursuing the Ph.D. degree with the School of Mathematics and Big Data, Anhui University of Science and Technology, China. She was a Lecturer with Pir Mehr Ali Shah Arid Agriculture University, Rawalpindi, and NUML University Islamabad, Pakistan. Other than that, she was the Manager Editor of *Journal of Rawalpindi Medical College* (JRMC). She has authored and co-authored research articles in various academic journals. Her research interests include stochastic petri nets, deep learning, machine learning, data management, advanced statistics, complex designs, probability theory, and statistical inference.