Journal of Information Systems Engineering and Management
2025, 10(59s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

An Efficient Advanced Data Integration From Multiple Sources For Fraud Detection Using Etl And Fl-Zsl-Glnrp3n

Rajesh Kumar Kanji[1], Vinodkumar Reddy Surasani[2], Sangeetha Govindarajan[3], Sai Tejaswi Bellapukonda[4]

[1]Independent Researcher, USA

[2]Sr Software Engineer, RBC Wealth Management, MN, USA

[3]Independent Researcher, USA

[4]Independent Researcher, USA

ARTICLE INFO

Received: 12 July 2025

Revised: 22 Aug 2025

Accepted: 02 Sept 2025

ABSTRACT

Advanced data integration combines data from multiple sources, which are then transformed and loaded into a unified system for decision-making. However, the existing studies didn't integrate structured and unstructured data for fraud detection in U.S financial transactions, affecting accurate decision-making. Therefore, this paper presents advanced data integration from multiple sources for fraud detection using ETL and FL-ZSL-GLNRP3N. Initially, the users are registered in the financial app, followed by key generation. While the user logs in to the financial app, the transaction is initiated. Afterward, identity theft mitigation is done. If the verifications are successful, the transaction details are encrypted by utilizing Koblitz Zorro Curve Cryptography (KHCC). Using FL, the fraud detection system is trained in a local model and updated in the global model. In the fraud detection system, ETL is employed for advanced data integration. Here, unstructured review data is structured using a Python library. Afterward, the data are integrated using COMA++. Next, temporal behavior analysis by Discrete Six-Hump Camel Jordan Wavelet Transform (DSHCJWT), word embedding, and feature extraction are performed. Lastly, fraud detection is done by using FL-ZSL-GLNRP3N. Here, Local Interpretable Model-agnostic Qing Explanation (LIMQE)-based DeepXplainer and ZSL are used. If the transaction is fraudulent, then it is stopped; otherwise, the transaction is completed. During testing, reviews obtained from original users and decrypted transaction details are employed for the fraud detection system. The results proved that the proposed model achieved a high accuracy of 98.76%.

Keywords: Advanced Data Integration, Fraud Detection, Extract Transform Load (ETL), Federated Learning (FL), COmbination MAtch (COMA++), One Time Password (OTP), Identity Theft Mitigation, and Federated Learning - Zero Shot Learning - Global Local Neuron Recurrent Path Norm Neural Network (FL-ZSL-GLNRP3N).

1. Introduction

In the United States, millions of financial transactions are processed every minute. Thus, credit card fraud remains a growing threat. Credit card fraud is the unauthorized and deceptive use of credit card information to obtain financial gain (Singh et al., 2022). Due to the expansion of digital payment channels and shifting of consumer behavior toward mobile and online banking, fraudsters develop more sophisticated schemes to exploit vulnerabilities across platforms (Han et al., 2021). Nowadays, advanced data integration is revolutionizing fraud detection in the U.S financial sector by enabling more intelligent and context-aware systems (Schneider & Brühl, 2023). For example, Bank of America employs multi-source data analytics to explore millions of daily credit card transactions and detect suspicious patterns like account takeovers or fraud attempts (Islam et al., 2023). Similarly, Fidelity Bank has adopted a layered fraud prevention framework that incorporates behavioral analytics, cross-channel transaction histories, and network-based risk scoring to detect fraudulent activities (Bello et al., 2023).

Moreover, by leveraging integrated data environments, banks can stay compliant with regulatory standards imposed by U.S federal financial institutions. Owing to the continuous evolution of fraud tactics, advanced data integration remains essential for building resilient and trustworthy fraud detection systems across the U.S financial ecosystem (Ghimire, 2024). Traditional fraud detection systems that rely on limited transactional data are no longer sufficient to protect financial institutions. To solve this problem, advanced data integration techniques are developed to enhance fraud detection capabilities (Wu & Wang, 2021). By combining various data streams, such as real-time transaction logs, customer profiles, spending behavior, device identifiers, geolocation data, and login patterns, the fraudulent activities are identified (Sánchez-Aguayo et al., 2021). In recent years, many Artificial Intelligence (AI) techniques have been developed to detect fraud transactions. In existing studies, Machine Learning (ML) techniques, such as Support Vector Machine (SVM), decision tree, and random forest, were used to detect credit card fraud transactions (Dileep et al., 2021) (Kumar et al., 2022).

Similarly, certain prevailing works utilized Artificial Neural Network (ANN), naïve bayes, logistic regression, FL, and blockchain technology for detecting the fraud transactions (Bin Sulaiman et al., 2022) (Tanouz et al., 2021). Likewise, Interpretable canonical deep tabular data learning architecture (TabNet), eXtreme Gradient Boosting (XGBoost), and advanced transformer techniques were utilized for credit card fraud detection (Yu et al., 2024). Also, some existing studies employed Generative Adversarial Network (GAN) to augment the data for effective credit card fraud detection (Langevin et al., 2022) (Strelcenia & Prakoonwit, 2023). However, the existing studies didn't focus on integrating the structured (transaction data) and unstructured (review data) for fraud detection in U.S financial transactions, thus affecting the accuracy. To conquer this shortcoming, a novel ETL and FL-ZSL-GLNRP3N-enabled advanced data integration from multiple sources for fraud detection is proposed in this article.

1.1 Problem Statement

The existing advanced data integration with fraud detection in U.S financial transactions models has some limitations, which are listed below,

- None of the existing works concentrated on integrating structured (Transaction data) and unstructured (social media feeds) data for fraud detection in U.S financial transactions, thus affecting the accurate decision making.
- The existing (Strelcenia & Prakoonwit, 2023a) failed to authenticate the identity theft, leading to financial loss and unauthorized access.
- Fraud patterns were not static and evolved over time (e.g., new scams, tactics). The prevailing (Lebichot et al., 2021) lacked the capability to capture and adapt to these temporal dynamics, thus making the system less responsive to shifts in fraud trends.
- The existing (Razaque et al., 2022) failed to identify previously unknown fraud patterns, thus hindering performance.
- Fraud detection systems required real-time processing of transactional data, user activity logs, and reviews to identify and respond to suspicious behavior promptly. However, existing

(Karthik et al., 2022) had problems with data latency, which delayed the delivery and analysis of critical information, ultimately hindering timely fraud detection.

- Most of the existing works didn't ensure transparency and trust in automated decisions, thus lacking an understanding of the reasons behind fraud predictions.
- In existing studies, integrating sensitive personal and financial information into fraud detection systems might lead to data breaches during transmission.

1.2 Objective

The key objective of the proposed model is explained as follows,

- Structured (transaction data) and Unstructured (social media feeds) data are integrated for accurate fraud detection.
- Password verification, OTP verification, and Secretion verification by ZUMHKP are performed to mitigate identity theft.
- DSHCJWT is employed to capture and adapt to the temporal dynamics, thus making the system more responsive to fraud trends.
- ZSL is established to learn unknown fraud patterns.
- FL is used to avoid the data latency, thus enabling timely fraud detection.
- LIMQE is employed to provide explanations about the fraud detection outcomes.
- KHCC is introduced to secure the transaction details, thus avoiding data breaches.

This paper is structured as follows: Section 2 conveys the literature survey, Section 3 illustrates the proposed research methodology, Section 4 elucidates the results and discussion, and Section 5 concludes the proposed model with future enhancements.

2. Literature Survey

(Karthik et al., 2022) presented a behavior pattern-based credit card fraud detection system. In this work, a bagging-based ensemble classifier, which included a random forest and an extra tree classifier, was employed for credit card fraud detection. Also, the model handled the data imbalance problem and detected the unseen fraudulent transactions. Hence, the model achieved better performance with respect to detection rate, accuracy, and area under the curve. Yet, the model had problems with data latency, thus hindering timely fraud detection.

(Strelcenia & Prakoonwit, 2023a) developed a framework for improving the performance of credit card fraud detection. Here, a data augmentation technique named Kullback-Leibler Conditional Generative Adversarial Network (K-CGAN) was employed to solve the data imbalance problem. Afterward, hybrid ML techniques, such as logistic regression, random forest, etc., were used to detect the fraud transactions. The model obtained higher precision and accuracy. But, the model failed to authenticate identity theft, leading to unauthorized access.

(Lebichot et al., 2021) suggested a model named credit card fraud detection based on transfer learning strategies. In this work, the ensemble model, including a random forest and Deep Neural Network (DNN), was employed for identifying the fraudulent transactions. Here, the transfer learning strategies improved the performance and accuracy of credit card fraud detection. Nevertheless, the research didn't capture and adapt to the temporal dynamics, thus making the system less responsive to shifts in fraud trends.

(Abdul Salam et al., 2024) introduced a model for credit card fraud detection with data balancing techniques. In this work, for addressing the imbalanced class issues, a hybrid resampling method was employed. Also, the SMOTE resampling technique with the CNN and FL was utilized for credit card fraud detection. The model enhanced the classification efficacy and had high accuracy. However, the model had high computational time and security issues.

(Razaque et al., 2022) employed big data analytics for credit card fraud detection and prevention. Here, the Random Undersampling technique was employed to balance the data. Also, the model used t-distributed Stochastic Neighbor Embedding (t-SNE), Principal Component Analysis (PCA), Singular Value Decomposition (SVD), and Logistic Regression Learning (LRL) for credit card fraud detection. The model achieved high accuracy and excellently prevented fraudulent transactions. But, the research didn't learn previously unknown fraud patterns, thus affecting the performance.

(Khalid et al., 2024) propounded a model for credit card fraud detection. In this research, the ensemble technique that integrated SVM, random forest, KNN, bagging, and boosting classifiers was employed for credit card fraud detection. The model achieved superior performance and excellently mitigated the challenges associated with credit card fraud detection. Yet, the model struggled to handle the sequential behavior.

(Seera et al., 2024) established an efficient payment card fraud detection system. Here, thirteen statistical and machine learning methods, namely ANN, Multi-Layer Perceptron (MLP), SVM, etc., were employed for a payment card fraud detection system. Hence, the experimental outcomes proved that the model had high effectiveness. However, the model might memorize past patterns and fail to generalize to evolving fraud strategies.

(Van Belle et al., 2023) presented a network-enabled credit card fraud detection framework. In this work, node representation learning was utilized for credit card fraud detection that avoided manual feature engineering. The model outperformed the state-of-the-art methods and excellently detected the fraud and normal transactions. Nevertheless, the research didn't address the class imbalance issue, which could bias the model toward normal transactions.

(Ileberi et al., 2022) developed ML-enabled credit card fraud detection system. In this work, the optimal features were selected by employing a Genetic Algorithm (GA). Afterward, ML approaches, including decision tree, logistic regression, random forest, ANN, and Naïve Bayes (NB), were employed for detecting the credit card fraud. The model outperformed the prevailing approaches. But, the integration of financial information might lead to data breaches during transmission.

(Alharbi et al., 2022) established a credit card fraud detection system using the Deep Learning (DL) technique. In this work, the Convolutional Neural Network (CNN) algorithm was employed to detect fraud transactions. The model had high robustness and detection accuracy. However, the model had high complexity and data representation overhead.

3. Proposed Advanced Data Integration From Multiple Sources For Fraud Detection Methodology

Here, the proposed FL-ZSL-GLN-RP3N is introduced to detect fraudulent transactions. For excellently mitigating identity theft, the proposed ZUMHKP is employed. Also, the proposed KHCC is established to secure the transaction details. For capturing the temporal dynamics of transactions and fraud patterns over time, the proposed DSHCJWT is utilized. Likewise, the proposed LIMQE is used to provide explanations about the fraud detection outcomes. Similarly, the ETL technique is employed for advanced data integration. The pictorial representation of the proposed model is displayed in Figure 1.

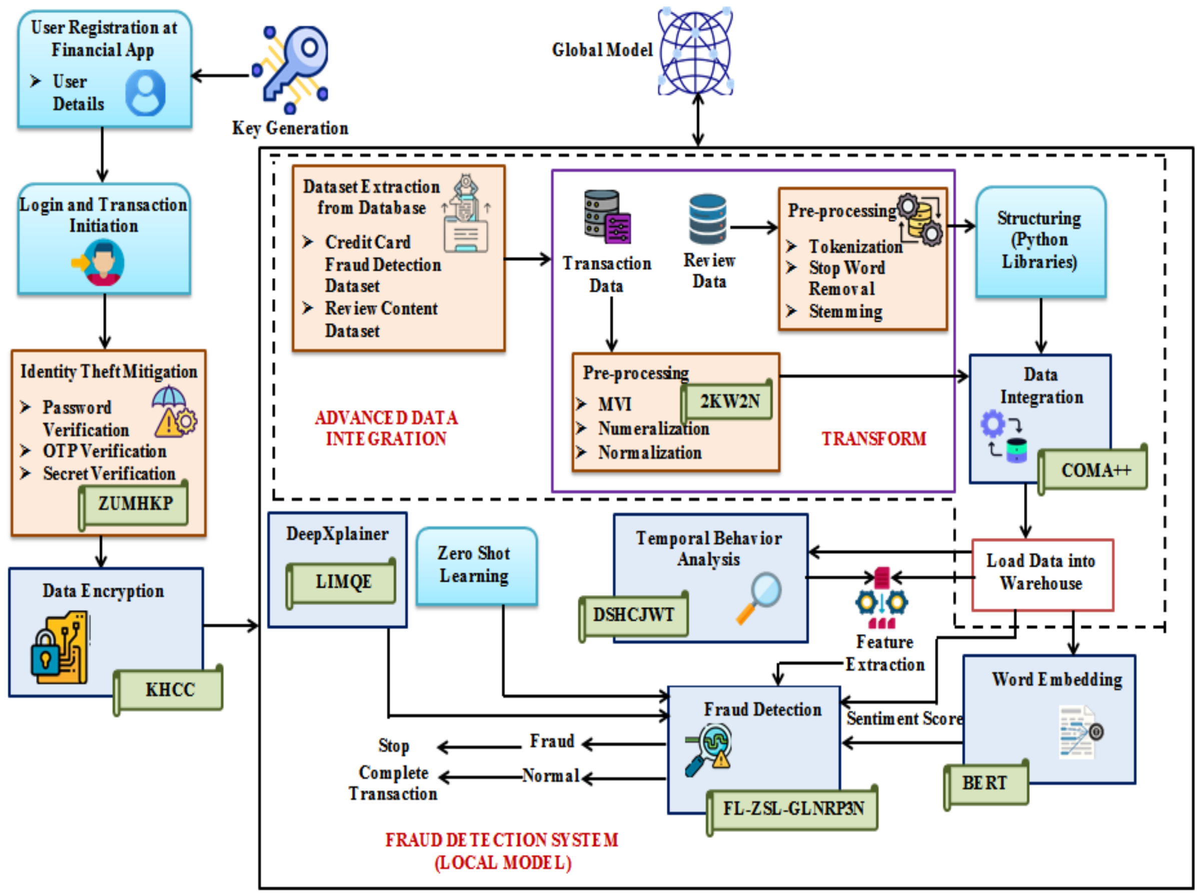


Figure 1: Pictorial layout of the proposed model

Fraud detection systems help the United States (U.S) financial transactions to prevent financial losses and protect customer assets by identifying suspicious activities in real time. Also, advanced data integration is vital for U.S. financial transactions to unify data from multiple sources, enabling a 360-degree view of customers and operations. Likewise, advanced data integration supports real-time decision-making and improves data quality across systems. The step-by-step working process of the proposed model is explained below,

3.1 User Registration at Financial App

Primarily, the users are registered with the financial application by using their details, such as username, residential address, Social Security Number (SSN), mobile number, email, bank account details, credit card information, password, etc. At the time of registration, the public and private keys are generated for the users by using KHCC. The process of KHCC is explained in Section 3.4. The total x number of registered users $\left(\mathfrak{R}_j^{user}\right)$ at the financial app is defined as,

$$\mathfrak{R}_j^{user} \rightarrow \left[\mathfrak{R}_1^{user}, \mathfrak{R}_2^{user}, \mathfrak{R}_3^{user}, \ldots\ldots, \mathfrak{R}_x^{user}\right] \tag{1}$$

Where, $\mathfrak{R}_x^{user}$ indicates the x^{th} registered user.

3.2 Login and Transaction Initiation

After successful registration, $\left(\Re_j^{user}\right)$ login to the financial application using the username and password to initiate the transaction process. If the details provided at the time of registration and login are matched, then the user can get access to the financial application.

3.3 Identity Theft Mitigation

Initially, for mitigating identity theft, the password verification is done. Here, the passwords entered during registration and login are checked. If both passwords are matched, then layered OTP verification is performed to ensure a high level of security. A layered OTP needs multiple OTPs (login, transaction initiation, and transaction confirmation). If OTP verification is successful, then secret verification is performed by employing Zero Universal Modulo Hash Knowledge Proof (ZUMHKP) to avoid financial loss and unauthorized access. Here, a secret $(\varsigma\varepsilon c)$ is created by the user at the time of registration and stored on the server. In general, Zero Knowledge Proof (ZKP) allows individuals to prove knowledge or ownership without exposing sensitive information. Likewise, ZKP had improved efficiency and versatility. Nevertheless, ZKP mainly relies on a trusted setup phase in which numerous parameters are created and distributed, thus leading to security problems. To overcome this issue, the Universal Modulo Hash function (UMH) is coupled with the setup parameters for enhancing the security level. The mathematical expression of the proposed ZUMHKP is described as follows,

Step 1: Initially, the parameter setup is done. In the next step, the commitment (M) is generated, and the created commitment is sent to the verifier by the prover. It is written as,

$$M = gen \,\hat{}\, \partial A \tag{2}$$

Where, gen defines the generator and ∂A denotes a random number.

Step 2: Subsequently, the random challenge (γC) is generated by the verifier, and the created random challenge is sent to the prover. After that, the prover estimates the response, and the response is sent to the verifier by the prover. In this step, the UMH (U) is utilized to elevate the security level. It can be mathematically expressed as,

$$U(\varsigma) = \left(\left(\left(m \cdot \varsigma + n\right) \bmod \mathrm{Pr}\right)\varsigma \quad \bmod \mathrm{Pr}\right) \times \varsigma\varepsilon c \tag{3}$$

Where, m and n signifies the random integers, Pr implies the larger prime number, ς specifies the personal identity data of the original user, and $\bmod$ represents the modulo function. Also, the response computed by the prover is equated as,

$$Rs = U\left(\partial A + \gamma C * \varsigma\varepsilon c\right) \tag{4}$$

Where, Rs implies the response computed by the prover.

Step 3: Lastly, the secret verification is performed. Here, the verifier checks whether the prover knows the secret $(\varsigma\varepsilon c)$ or not. It is given as,

$$SV = \begin{cases} gen^{\wedge}Rs = M \cdot gen^{\wedge}(\gamma C \cdot \varsigma \varepsilon c) & s\mu \\ otherwise & Ns\mu \end{cases} \tag{5}$$

Where, SV indicates the secret verification outcomes, $s\mu$ represents that secret verification is successful, and $Ns\mu$ implies that secret verification has failed. If the secret verification is successful $(s\mu)$, then further transaction processes are done; otherwise, access is denied.

3.4 Data Encryption

After successful authentication, the transaction data $(\tau\partial)$ is encrypted by using KHCC to avoid data breaches. Generally, Elliptic Curve Cryptography (ECC) performs encryption and decryption faster and also consumes less memory and processing power. However, ECC generates the private keys using random values. If an attacker can predict the random number generation process, then they can easily derive private keys, exposing sensitive information. To address this problem, the Koblitz Zorro curve is used to generate the keys. The working process of the proposed KHCC is explained as follows,

- ***Koblitz Zorro Curve***

Primarily, for generating the keys, the Koblitz Zorro curve is computed. It is mathematically expressed as,

$$(q^2)^2 + pq = (p^3)^2 + \upsilon p^2 + \kappa + 5 \tag{6}$$

Where, p and q demonstrates the horizontal and vertical axes, correspondingly, and υ and κ depicts the random numbers.

- ***Key Generation***

Here, the keys are estimated from the Koblitz Zorro curve. The generated public key (Ω) is given as,

$$\Omega = P * \Gamma \tag{7}$$

Here, Γ implies the point on the Koblitz Zorro curve and P indicates the private key.

- ***Encryption***

Next, the transaction data $(\tau\partial)$ is encrypted to avoid data breaches. The encrypted data is expressed as,

$$K1 = rn \times \Gamma \tag{8}$$

$$K2 = \tau\partial + rn \bullet \Omega \tag{9}$$

Where, $K1$ and $K2$ represents the cipher texts, rn implies the random value, and the encrypted transaction data is indicated as Enc_c.

- ***Decryption***

Lastly, the encrypted transaction data (Enc_c) is decrypted on the receiver side (i.e., fraud detection system) and is equated as,

$$\tau\partial = (K2 - P) * K1 \tag{10}$$

Where, $\tau\partial$ represents the decrypted data. The pseudocode for KHCC is depicted as follows,

Pseudocode for KHCC

```
Input: Transaction data $(\tau\partial)$
Output: Encrypted data $(Enc_c)$
Begin
    Initialize $(\tau\partial)$
    For each $(\tau\partial)$
        Estimate Koblitz Zorro curve
            $(q^2)^2 + pq = (p^3)^2 + \upsilon p^2 + \kappa + 5$
        Generate public key
            $\Omega = P * \Gamma$
        Perform Encryption
            $K1 = rn \times \Gamma$
            $K2 = \tau\partial + rn \bullet \Omega$
        Implement decryption
            $\tau\partial = (K2 - P) * K1$
    End For
    Obtain Encrypted data $(Enc_c)$
End
```

Thus, the proposed KHCC excellently secured the transaction details from data breaches. Afterward, the (Enc_c) is decrypted at the fraud detection system.

3.5 Fraud Detection System

Here, the fraud detection system is trained in a local model and updated in a global model using FL. In the fraud detection system, the significant processes, such as advanced data integration, temporal behavior analysis, word embedding, feature extraction, and fraud detection, are performed to excellently identify the fraudulent transactions in U.S banks. This is explained as follows,

3.5.1 Advanced Data Integration

For advanced data integration, the ETL technique is employed. It is explained briefly as follows,

→ **Extract**

Firstly, in the extract process, the datasets, namely "Credit Card Transactions Fraud Detection Dataset" (Structured) and "Financial App Review" (Unstructured), are extracted from the database.

The “Fraud Detection” dataset that includes transaction details is indicated as Θ_z. Also, the “Financial App Review” that contains review contents and sentiment score is denoted as R_y.

→ **Transform**

Then, in the transform process, pre-processing is done for the Θ_z and R_y to prepare the data for further analysis.

Pre-Processing (Transaction data): Here, firstly, the missing values in Θ_z are imputed by employing K-Kernel-based Weighting Nearest Neighbors (2KW2N). Based on the local context, K-Nearest Neighbor (KNN) imputes values using the closest data points, thus providing more accurate imputations. However, KNN’s performance mainly depends on the choice of the number of neighbors (k). If the k value is small, then it can make the method sensitive to noise. If the k value is large, then it can oversmooth the data, resulting in less accurate imputations. To address this problem, the Kernel-Based Weighting (KW) is employed to select the k value. The mathematical expression of 2KW2N is described below,

Here, the KW technique is utilized to select the k-value (J) and is equated as,

$$J \xrightarrow{\Theta_z} \exp\left(-\frac{\|\Theta-\Theta_i\|}{2bd}\right) \tag{11}$$

Where, $\exp$ indicates the exponential function, Θ implies the query point, Θ_i exemplifies the neighbor data point, and bd defines the bandwidth parameter. After that, the distance (Dis) is estimated between each data point to identify the nearest k-neighbors regarding (J).

$$Dis(\Theta_1,\Theta_2) \xrightarrow{J} \sqrt{\sum(\Theta_1-\Theta_2)^2} \tag{12}$$

The closest k-neighbors are recognized according to the distance (Dis). Subsequently, based on the average value of the k-neighbors, the missing values are imputed. The missing value imputed data is specified as Ms_{im}. Thereafter, in the numeralization process, the Ms_{im} is converted into numerical form and is denoted as,

$$Nm(Ms_{im}) \Rightarrow [Nm_1, Nm_2, Nm_3, \ldots\ldots, Nm_{ab}] \tag{13}$$

Where, Nm defines the numeralized data and Nm_{ab} specifies the number of numeralized data. Next, Nm is normalized into 0s and 1s using the Z-score normalization technique. The pre-processed transaction details’ data $(\wp_{tran})$ is given as,

$$\wp_{tran} = \frac{Nm - m\zeta}{\sigma} \tag{14}$$

Where, σ signifies the standard deviation and $m\zeta$ represents the mean.

Pre-Processing (Review data): Likewise, the R_y are pre-processed. Initially, the R_y are converted into small tokens in the tokenization process. It is written as,

$$R_y \xrightarrow{tokenized} \varpi \tag{15}$$

The tokenized words are signified as ϖ. Afterward, the stop words, including “for”, “and”, “the”, “of”, etc., present in the ϖ are removed. It is given as,

$$\varpi \xrightarrow{remove} Stp \tag{16}$$

The stop word removed data is indicated as Stp. Subsequently, the stemming process is performed on the Stp to reduce the word to its root form and is represented as Tm. Thus, the pre-processed review data (pr_Z) is given as,

$$pr_Z(Tm) \rightarrow \{pr_1, pr_2, pr_3, \ldots\ldots, pr_{mn}\} \quad Here \quad Z = (1\ to\ mn) \tag{17}$$

Where, $Z = (1\ to\ mn)$ indicates the number of pre-processed review data.

→ **Load**

Afterward, the unstructured (pr_Z) are structured by using the Python library named “python-docx”. Generally, “python-docx” converts the unstructured report content into analyzable formats. The structured review data are specified as,

$$pr_Z \xrightarrow{structured} St_{rw} \tag{18}$$

Where, St_{rw} implies the structured review data. Subsequently, the $(\wp_{tran})$ and St_{rw} are integrated by using the schema matching algorithm named COMA++. Generally, COMA++ supports multiple matching strategies, thus increasing the accuracy and flexibility across various data sources. Initially, COMA++ loads and parses the schemas of $(\wp_{tran})$ and St_{rw}. Afterward, a time-based matcher is applied to compare elements from each schema. Thereafter, a similarity score is computed for each element pair. Next, the similarity scores are combined, and suggested mappings between related fields are generated. The integrated data is denoted as $A_{\partial t}$. Afterward, the $A_{\partial t}$ are loaded into the data warehouse. The loaded data (L_w) is defined as,

$$L_w(A_{\partial t}) \rightarrow \langle L_1, L_2, L_3, \ldots\ldots, L_l \rangle \tag{19}$$

Here, L_l indicates the l^{th} loaded data.

3.5.2 Temporal Behavior Analysis

By utilizing DSHCJWT, the temporal behavior analysis for (L_w) is done to capture the temporal dynamics of transactions over time. In general, the Discrete Wavelet Transform (DWT) can adapt to diverse frequency components at various times. Therefore, it is well-suited for analyzing the temporal behavioural data. Yet, DWT involves decomposition into multiple levels, which may result in information loss. To address this problem, Jordan Decomposition is used. But, the Jordan Decomposition is sensitive to round-off errors and small perturbations in the input matrix. To overcome this issue, the Six-Hump Camel technique is included. The working process of DSHCJWT is described as follows,

In the first step, by passing (L_w) through the series of filters, the DWT (β) of (L_w) is detected. The (L_w) are passed via the low-pass filter and are expressed as,

$$\beta(s) = (L_w * imp)[s] \rightarrow \sum_{\ddot{n}=-\infty}^{\infty} L_w[\ddot{n}] \cdot imp[(s - \ddot{n}) \tag{20}$$

Here, s implies the level count, imp denotes the impulse response, and $\ddot{n}$ depicts the shift parameter. After that, by using Jordan Decomposition, the (L_w) are decomposed. Also, the Six-Hump Camel technique is included in Jordan Decomposition to avoid the round-off errors and is given as,

$$j\gamma(L_w) \xrightarrow{L_w} Mt \Im Mt^{-1} + \chi \tag{21}$$

$$\chi = \left(4 - 2.1(L_1)^2 + \frac{(L_1)^4}{3}\right)(L_1)^2 + L_1L_2 + \left(-4 + 4(L_2)^2\right)(L_2)^2 \tag{22}$$

Where, Mt defines the matrix of generalized eigenvectors, Mt^{-1} elucidates the inverse of Mt, $\Im$ depicts the Jordan Matrix, and χ exemplifies the Six-Hump Camel technique. Next, the low-pass filter outcomes and decomposition outcomes are provided to the new low-pass filter and decomposition. It is equated as,

$$\beta_{low}[s] = \sum_{\ddot{n}=-\infty}^{\infty} L_w[\ddot{n}] \cdot imp(2s - \ddot{n}) \tag{23}$$

$$\beta_{high}[s] = \sum_{\ddot{n}=-\infty}^{\infty} L_w[\ddot{n}] \cdot \hbar c(2s - \ddot{n}) \tag{24}$$

Here, $\hbar c$ indicates the coefficient. Subsequently, in the low-pass filter and decomposition process, the sub-sampling operator is computed and applied. It is represented as,

$$(\beta \downarrow \ddot{n})[s] = \beta(\ddot{n}s) \tag{25}$$

$$\beta_{low} = (L_w \times imp) \downarrow 2 \tag{26}$$

$$\beta_{high} = (L_w \times \hbar c) \downarrow 2 \tag{27}$$

Eventually, the obtained temporal behavior analysis outcomes are denoted as T_{an}.

3.5.3 Word Embedding

Next, word embedding is performed by using Bidirectional Representation from Transformers (BERT) to convert the (L_w) into a vector format. Generally, BERT generates different representations for the same word depending on its context. Also, it captures nuances and improves accuracy in various NLP tasks. The working of BERT is explained as follows,

❄ Firstly, (L_w) is converted into a vector representation. The final input embedding (IME) is defined as,

$$IME(L_w) = Tk + Pi + Sm \tag{28}$$

Here, Tk, Pi, and Sm demonstrates the token, positional, and segment embeddings, respectively.

❄ Next, each Tk is converted into Query (Qr), Key (Ky), and Value (Vlu) matrices as follows,

$$Qr = TkW^{Qr} \tag{29}$$

$$Ky = TkW^{Ky} \tag{30}$$

$$Vlu = TkW^{Vlu} \tag{31}$$

Here, W defines the weight matrix.

❄ After that, the inputs are passed via the stacked transformer layer, where each layer encompasses a multi-head self-attention and feed-forward network. In multi-head self-attention, the attention score (Ath) is estimated in terms of $(Qr),(Ky)$, and (Vlu) for each attention head. It is mathematically expressed as,

$$Ath \xrightarrow{IME} soft\left(\frac{QrKy^{Tr}}{\dim}\right)Vlu \tag{32}$$

Where, $soft$ indicates the softmax function, $\dim$ depicts the dimension of the embeddings, and Tr is the transpose matrix.

❄ In the next step, for obtaining multiple attention heads, the outcomes of all heads are summed. Then, through a feed-forward layer, each token is passed.

$$Fed \Rightarrow E(IMEW_1 + b_1)W_2 + b_2 \tag{33}$$

Here, Fed depicts the feed-forward layer outcomes, E elucidates the Rectified Linear Unit (ReLU) function, and b_1 and b_2 demonstrates the bias terms. The word embedding outcomes are denoted as W_ϕ.

3.5.4 Feature Extraction

From (L_w), the features, such as id, trans_date_trans_time, cc_num, merchant, category, amt, first, last, gender, street, city, state, zip, lat, long, city_pop, job, dob, trans_num, unix_time, merch_lat, merch_long, is_fraud, id, content, created_at, updated_at, language, lead_time, score, thumbsUpCount, sentiment, etc., are extracted. Also, from T_{an}, the temporal features, including mean, median, standard deviation, variance, entropy, moving averages, transaction frequency, session duration, etc., are extracted. The extracted features are signified as F_d.

3.5.5 Fraud Detection

Based on the sentiment score (SC), word embedding outcomes (W_ϕ), and extracted features (F_d), the fraud transactions are detected by using FL-ZSL-GLNRP3N. Normally, a Recurrent Neural Network (RNN) can capture long-term dependencies in sequential data. Also, RNN automatically learns more informative and abstract features. Nevertheless, RNN had vanishing gradient issues owing to the ineffective activation. Likewise, RNN had a high training time due to improper hyperparameter (weight and bias) tuning. To address the vanishing gradient problem, the Global-Local Neuron activation function is employed, which upgrades the learning efficiency. Also, the Path Norm regularization is used to initialize the hyperparameters, thus minimizing the time complexity. Likewise, the FL technique is included to avoid the data latency. Similarly, Zero Shot Learning (ZSL) is added to learn new or unknown fraud patterns. The diagrammatic representation of the proposed FL-ZSL-GLNRP3N is displayed in Figure 2.

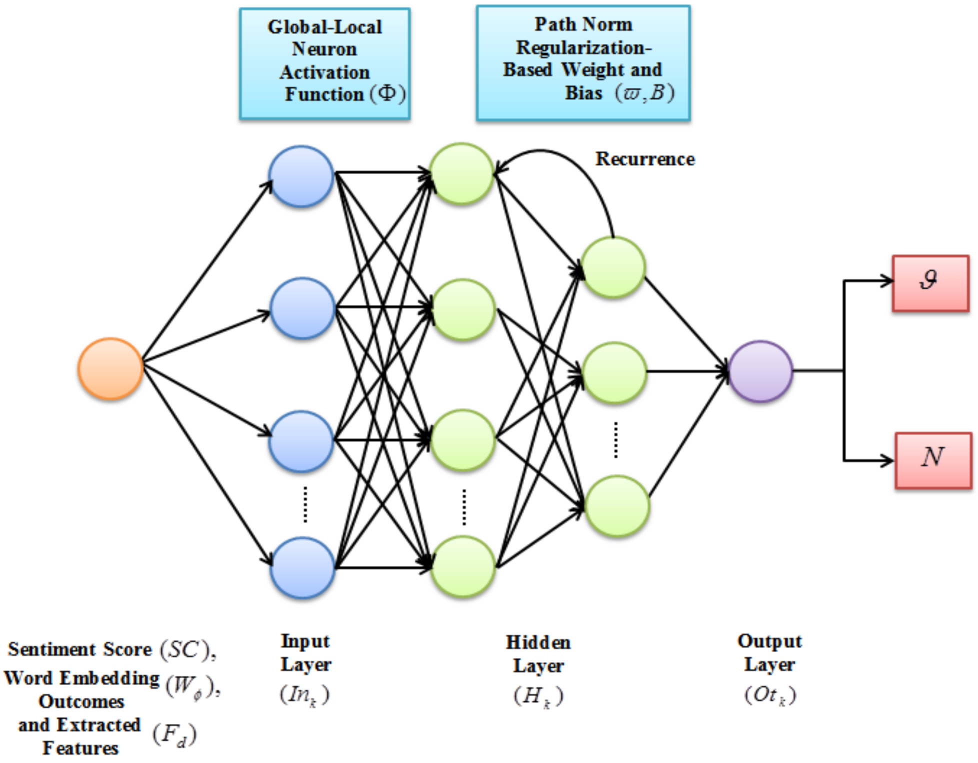

Global-Local Neuron Activation Function (Φ)
Path Norm Regularization-Based Weight and Bias (ϖ, B)
Recurrence
ϑ
N
Sentiment Score (SC), Word Embedding (W_ϕ), Outcomes and Extracted (F_d) Features
Input Layer (In_k)
Hidden Layer (H_k)
Output Layer (Ot_k)

Figure 2: Diagrammatic representation of the proposed FL-ZSL-GLNRP3N

FL-ZSL-GLNRP3N comprises an input layer, a hidden layer, and an output layer. The working process of the FL-ZSL-GLNRP3N classifier is described as follows,

☑ ***Input Layer***

Initially, the input layer gathers the sentiment score (SC), word embedding outcomes (W_ϕ), and extracted features (F_d) as input. Here, the input layer operation (In_k) is written as,

$$(SC, W_\phi, F_d) \rightarrow In_k \tag{34}$$

Next, the input layer data (In_k) is fed to the hidden layer state for performing further processes.

☑ ***Hidden Layer State***

Then, the past information about the previous state in sequence is indicated by the hidden layer state. Regarding the current input and previous hidden state, the hidden layer state is updated. The hidden layer state operation (H_k) is mathematically expressed as,

$$H_k = \Phi \cdot [(H_{k-1}, In_k) * \varpi + B] \tag{35}$$

Where, H_{k-1} implies the previous hidden state, Φ represents the Global-Local Neuron activation function, B and ϖ denotes the bias value and weight, which are initialized by using the Path Norm regularization technique. Here, Φ, B, and ϖ are formulated as follows,

$$\Phi = \delta(u) \cdot glo(In_k) + (1 - \delta(u)) \cdot loc(In_k) - v \tag{36}$$

$$(\varpi, B) \xrightarrow{In_k} \sum_{ph} \left(\prod_{(c', d') \in ph} (In_k)^2 \right) \tag{37}$$

Where, δ defines the sigmoid function, u and v demonstrates the trainable weights, glo and loc specifies the global and local neurons, respectively, ph depicts the path from input to output, and (c', d') signifies the neurons.

☑ ***Output Layer***

In the next step, the decisions about the fraudulent transactions are computed by the output layer (Ot_k). The output layer operation is written as,

$$Ot_k = \Phi(\varpi * H_k + B) \tag{38}$$

Also, the FL is employed for avoiding data latency and improving the timely detection. Likewise, ZSL is included here to learn the unknown fraud patterns, thus improving the performance. Eventually, the fraud detection outcomes (FD) are delivered by the FL-ZSL-GLNRP3N, and it is expressed as follows,

$$FD = \langle \vartheta, N \rangle \tag{39}$$

Where, ϑ denotes the fraudulent transaction and N defines the normal transaction. The pseudocode for FL-ZSL-GLNRP3N is given as,

Pseudocode for FL-ZSL-GLNRP3N

```
Input: Sentiment score $(SC)$, Word embedding outcomes $(W_\phi)$, and Extracted features $(F_d)$
Output: Fraud detection outcomes $(FD)$
Begin
    Initialize $(SC)$, $(W_\phi)$, and $(F_d)$
    For each $(SC)$, $(W_\phi)$, and $(F_d)$
        Estimate input layer operation $(In_k)$
            $(SC, W_\phi, F_d) \rightarrow In_k$
        Compute hidden layer state
            $H_k = \Phi \cdot [(H_{k-1}, In_k) * \varpi + B]$
        Find Global-Local Neuron activation function
            $\Phi = \delta(u) \cdot glo(In_k) + (1 - \delta(u)) \cdot loc(In_k) - \nu$
        Discover weight and bias based on Path Norm regularization
            $(\varpi, B) \xrightarrow{In_k} \sum_{ph} \left( \prod_{(c',d') \in ph} (In_k)^2 \right)$
        Perform output layer operation
            $Ot_k = \Phi(\varpi * H_k + B)$
    End For
    Obtain Fraud detection outcomes $FD = \langle \vartheta, N \rangle$
End
```

Thus, the proposed FL-ZSL-GLNRP3N excellently detected the fraudulent transactions. If the transaction is fraudulent, then it is stopped; otherwise, the transaction is completed.

3.5.6 DeepXplainer

Afterward, deep explanations about the FD are provided by employing LIMQE. Generally, Local Interpretable Model-agnostic Explanation (LIME) provides valuable insights into fraud detection outcomes. However, in LIME, an inappropriate kernel may not capture the underlying structure of the data effectively, leading to suboptimal results. To overcome this issue, the Qing function is introduced to assign the weight value. The working of LIMQE is explained below,

Firstly, the FD are chosen to initiate the process. Next, the perturbed samples are generated around the original instance and are denoted as (Ψ). Afterward, for each instance, the weight value $(\varpi\tau)$ is assigned using the Qing function.

$$\varpi\tau = \sum \left((FD)_f^i - f \right)^2 \tag{40}$$

Where, f indicates a parameter. Subsequently, the local model is trained using the weighted dataset, thus interpreting the complex model. After that, the coefficients of the local model demonstrate which features are most important for the FD. Thus, the LIMQE excellently provided the explanations about fraud detection.

During testing, the review is requested from the original users at the time of transaction initiation and given to fraud detection. Also, the encrypted transaction details $\left(Enc_c\right)$ are decrypted $\left(\tau\partial\right)$ using KHCC and given to the fraud detection system. Then, the fraud detection system detects the transaction as fraud or normal. Thus, the proposed methodology excellently performed fraud detection in U.S financial transactions by integrating data from multiple sources.

4. Result And Discussion

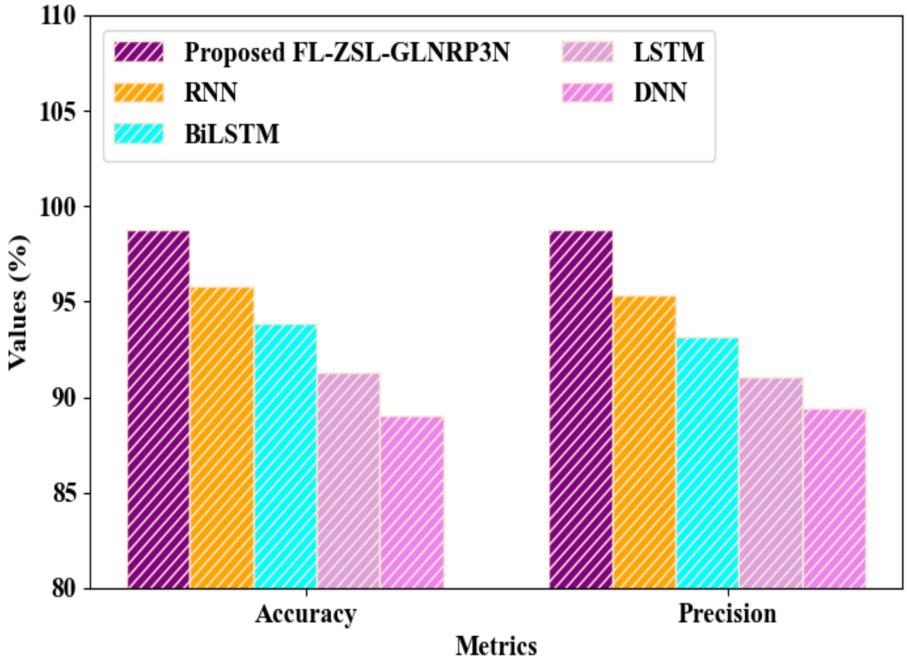
In this section, the comparative analysis and performance assessment of the proposed and prevailing techniques are carried out to prove the proposed model's reliability. Also, the proposed model is implemented in the working platform of PYTHON.

4.1 Dataset Description

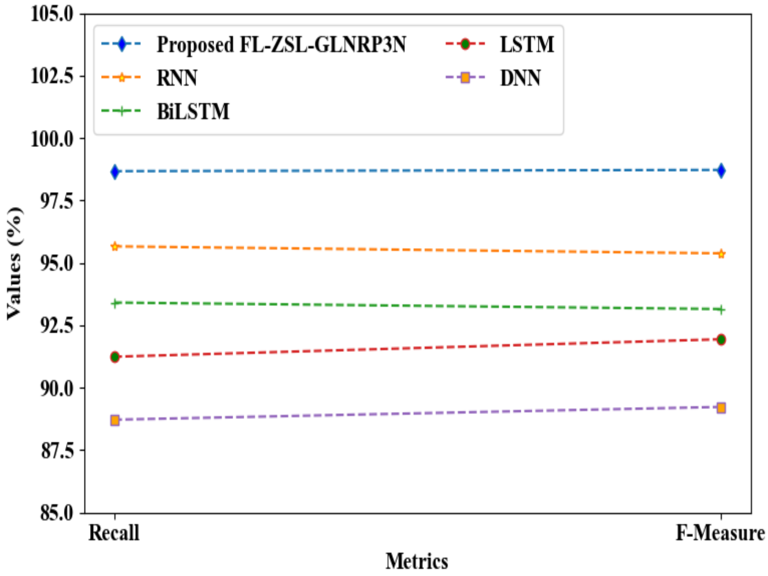
The proposed model employs the "Credit Card Transactions Fraud Detection Dataset" (Structured) for detecting fraudulent transactions. Also, the "Financial App Review" dataset (Unstructured), which is made by custom, is utilized. Here, the "Credit Card Transactions Fraud Detection Dataset" consists of 1604294 numbers of transaction details data. Also, "Financial App Review" dataset encompasses 388 numbers of review contents data. Among that, 80% of the data is employed for training and 20% of the data is utilized for testing purposes.

4.2 Performance Evaluation

Here, the performance of the proposed techniques is compared with prevailing techniques to show the proposed model's dependability.



(a)



(b)

Figure 3: Graphical representation regarding (a) Accuracy and Precision, (b) Recall and F-Measure

Figures 3 (a) and (b) depict the graphical representation of the proposed FL-ZSL-GLNRP3N and existing techniques in terms of accuracy, precision, recall, and F-Measure. Here, the proposed FL-ZSL-GLNRP3N achieved high accuracy, precision, recall, and F-Measure of 98.76%, 98.81%, 98.67%, and 98.72%, respectively. However, the existing techniques, including RNN, Bidirectional Long Short Term Memory (BiLSTM), Long Short Term Memory (LSTM), and Deep Neural Network (DNN), obtained low average accuracy, precision, recall, and F-Measure of 92.49%, 92.25%, 92.25%, and 92.42%, correspondingly. With the usage of Global-Local Neuron activation function and Path Norm regularization, the proposed FL-ZSL-GLNRP3N accurately detected the fraud transactions. Thus, the effectiveness of the proposed model was proved.

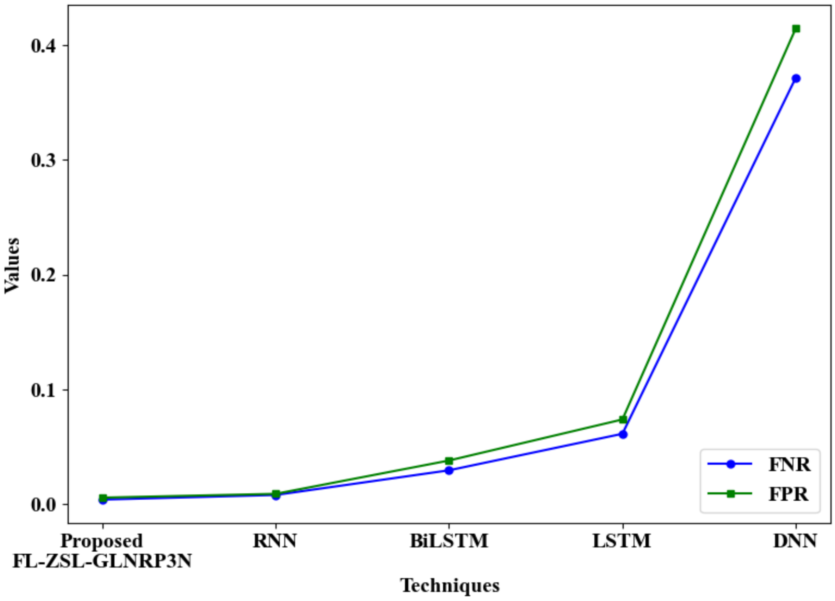


Figure 4: FNR vs FPR analysis

False Negative Rate (FNR) vs False Positive Rate (FPR) analysis of the proposed FL-ZSL-GLNRP3N and prevailing techniques is shown in Figure 4. Here, the proposed model employed Global-Local Neuron activation function and Path Norm regularization for accurate decision-making about fraud transactions. Also, ZSL is included to learn unknown fraud patterns. The proposed FL-ZSL-GLNRP3N obtained a low FNR (0.004138) and FPR (0.0058198). But, the prevailing RNN, BiLSTM, LSTM, and DNN attained high FNR of 0.008153, 0.029646, 0.0614537, and 0.371665, respectively. Also, the prevailing techniques attained high FPR values. Thus, the results proved that the proposed model was superior to the prevailing techniques.

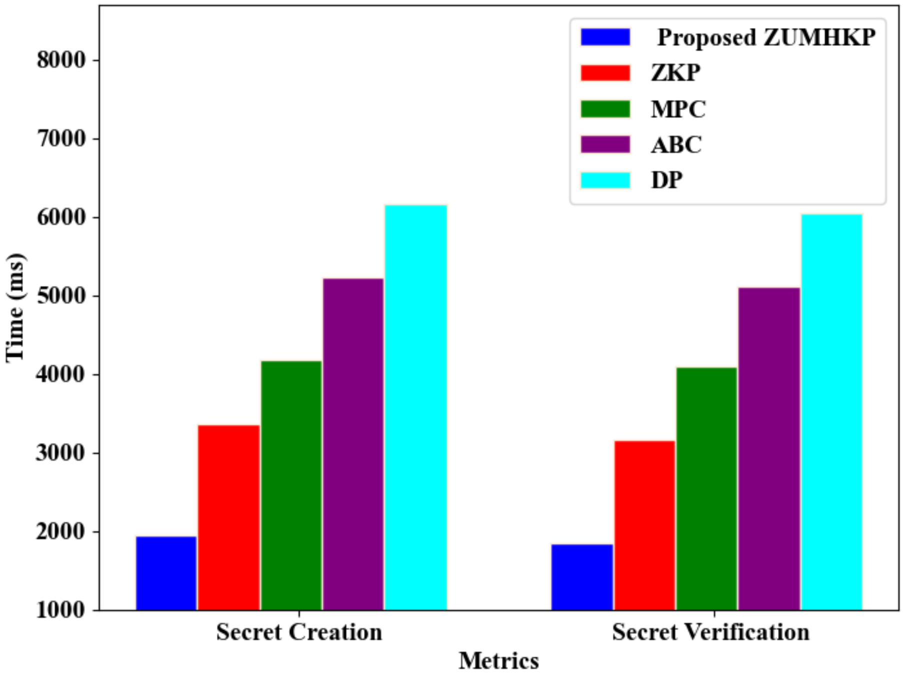


Figure 5: Performance validation regarding secret creation and verification time

To avoid identity theft, the proposed ZUMHKP was established, which verified the secrets of users and provided enhanced authentication. Figure 5 depicts the performance validation of the proposed ZUMHKP and conventional methods in terms of secret creation time and secret verification time. Here, the proposed ZUMHKP took less time of 1956ms for secret creation and 1854ms for secret verification. But, the conventional ZKP and Multi-Party Computation (MPC) took a maximum tree creation time of 3372ms and 4178ms, respectively. Also, the conventional Attribute-Based Credentials (ABC) and Differential Privacy (DP) took a maximum secret verification time of 5124ms and 6055ms, respectively. Here, the proposed ZUMHKP excellently mitigated the identity theft due to the usage of UMH.

Table 1: Comparative Evaluation with respect to MSE and PSNR

Techniques	MSE	PSNR (dB)
Proposed DSHCJWT	0.0328	33.516
DWT	0.1439	28.934
LWT	0.4987	24.845
CWT	0.8562	21.502
SWT	1.5681	17.587

Table 1 shows the comparative evaluation of the proposed DSHCJWT and existing methods regarding Mean Squared Error (MSE) and Peak-Signal-to-Noise Ratio (PSNR). Here, the proposed DSHCJWT achieved a low MSE (0.0328) and high PSNR (33.516 dB). Likewise, the existing methods, such as DWT, Lifting Wavelet Transform (LWT), Continuous Wavelet Transform (CWT), and Stationary Wavelet Transform (SWT), attained a high average MSE of 0.76672 and a low average PSNR of 23.217 dB. Here, the Six-Hump Camel Jordan technique was modified with DWT for effectively capturing the temporal patterns over time.

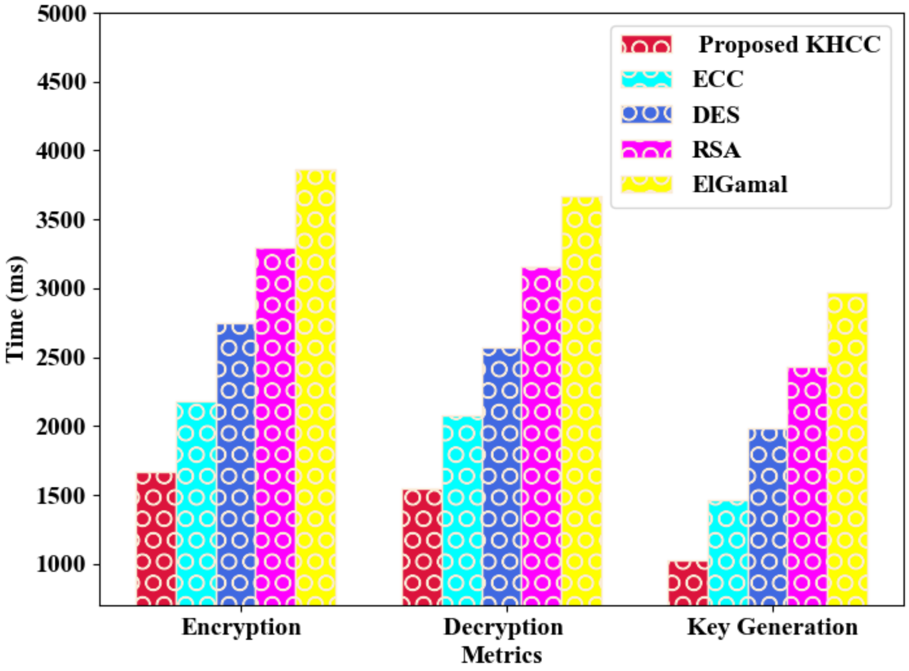


Figure 6: Pictorial analysis of the proposed and conventional methods

Pictorial analysis of the proposed KHCC and conventional methods in terms of encryption time, decryption time, and key generation time is depicted in Figure 6. Here, the KHCC used Koblitz Zorro Curve to avoid exposing sensitive information due to random key generation. The proposed KHCC took less encryption time, decryption time, and key generation time of 1672ms, 1544ms, and 1544ms, respectively. Also, the conventional ECC, Data Encryption Standard (DES), Rivest Shamir Adleman (RSA), and ElGamal took maximum average encryption time, decryption time, and key generation time of 3020.75ms, 2867.75ms, and 2214.5ms, respectively. Thus, the trustworthiness of the proposed model was demonstrated.

Table 2: Performance assessment in terms of fidelity and sparsity

Methods	Fidelity	Sparsity
Proposed LIMQE	0.981	0.984
LIME	0.794	0.732
SHAP	0.628	0.659
PDP	0.503	0.471
GS	0.397	0.356

Table 2 depicts the performance assessment of the proposed LIMQE and existing methods with respect to fidelity and sparsity. Here, the proposed LIMQE achieved a high fidelity and sparsity of

0.981 and 0.984. But, the existing LIME and SHapley Additive exPlanations (SHAP) obtained low fidelity of 0.794 and 0.628, respectively. Also, the existing Partial Dependence Plot (PDP) and Global Surrogate attained low sparsity of 0.471 and 0.356, respectively. Here, the Qing function was modified with LIME for assigning the weight value, thus enhancing the performance.

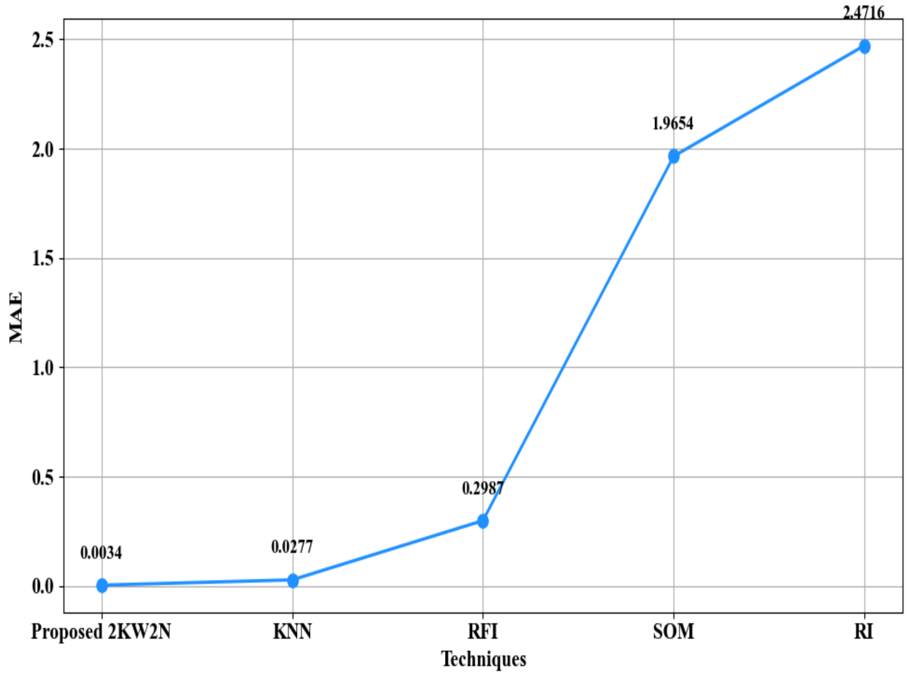


Figure 7: MAE validation

Mean Absolute Error (MAE) validation of the proposed 2KW2N and existing methods is displayed in Figure 7. Here, the proposed 2KW2N obtained a low MAE of 0.0034 owing to the inclusion of KW, whereas the existing KNN, Random Forest Imputation (RFI), Self Organizing Maps (SOM), and Regression Imputation (RI) attained high MAEs of 0.0277, 0.2987, 1.9654, and 2.4716, respectively. Thus, the efficacy of the proposed model was proved.

4.3 Comparative Analysis

Here, the comparative analysis of the proposed and related works is done as follows,

Table 3: Comparative Validation

Author's Name	Technique	Recall (%)	Precision (%)	F-Measure (%)
Proposed Model	FL-ZSL-GLNRP3N	98.67	98.81	98.72
(Malik et al., 2022)	Adaboost + LGBM	64	97	77
(Asha & Kumar, 2021)	SVM	89.76	97.43	-
(Habibpour et al., 2023)	EMCD	-	32	-

(Benchaji et al., 2021)	LSTM-Attention	97.22	97.69	-
(Alfaiz & Fati, 2022)	AllKNN-CatBoost	95.91	80.28	87.40

Table 3 depicts the comparative validation of the proposed and related works. Here, the proposed FL-ZSL-GLNRP3N achieved a high recall, precision, and F-Measure of 98.67%, 98.81%, and 98.72%, respectively, owing to the usage of Global-Local Neuron activation function and Path Norm regularization. However, the SVM and LSTM-Attention obtained low accuracy of 89.76% and 97.22%, respectively. Likewise, the conventional Ensemble Monte Carlo Dropout (EMCD) technique attained a very low precision of 32%. Also, the prevailing Adaptive Boosting + Light Gradient Boosting Machines (Adaboost + LGBM) and All K-Nearest Neighbors with CatBoost (AllKNN-CatBoost) obtained very low F-Measure of 77% and 87.40%, respectively. Thus, the results proved that the proposed model was better than the prevailing works.

5. Conclusion

This paper presented an effective framework named ETL and FL-ZSL-GLNRP3N-enabled advanced data integration from multiple sources for fraud detection. In the proposed methodology, significant processes, such as identity theft mitigation, data encryption, advanced data integration, temporal behavior analysis, word embedding, fraud detection, and DeepXplanation, were performed. Here, the proposed FL-ZSL-GLNRP3N achieved high accuracy and precision of 98.76% and 98.81%, respectively, which proved the effectiveness of the proposed model. Also, the proposed ZUMHKP took less time of 1956ms and 1854ms for secret creation and verification, respectively. Likewise, the proposed DSHCJWT had low MSE (0.0328) and high PSNR (33.516 dB), which demonstrated the reliability of the proposed model. Overall, the proposed model achieved high trustworthiness and efficacy. Although the proposed model integrated the structured and unstructured data for fraud detection, it didn’t consider the semi-structured data.

Future Scope

In the future, advanced techniques will be developed to integrate the semi-structured data (i.e., logs, XML, JSON, etc) with structured and unstructured data for fraud detection, thus further improving accurate decision-making.

References

Dataset link: https://www.kaggle.com/datasets/kartik2112/fraud-detection

[1] Abdul Salam, M., Fouad, K. M., Elbably, D. L., & Elsayed, S. M. (2024). Federated learning model for credit card fraud detection with data balancing techniques. *Neural Computing and Applications*, *36*(11), 6231-6256. https://doi.org/10.1007/s00521-023-09410-2

[2] Alfaiz, N. S., & Fati, S. M. (2022). Enhanced credit card fraud detection model using machine learning. *Electronics*, *11*(4), 662. https://doi.org/10.3390/electronics11040662

[3] Alharbi, A., Alshammari, M., Okon, O. D., Alabrah, A., Rauf, H. T., Alyami, H., & Meraj, T. (2022). A novel text2IMG mechanism of credit card fraud detection: A deep learning approach. *Electronics*, *11*(5), 756. https://doi.org/10.3390/electronics11050756

[4] Asha, R. B., & Kumar K. R. S (2021). Credit card fraud detection using artificial neural network. *Global Transitions Proceedings*, *2*(1), 35-41. https://doi.org/10.1016/j.gltp.2021.01.006

[5] Bello, O. A., Ogundipe, A., Mohammed, D., Adebola, F., & Alonge, O. A. (2023). AI-Driven Approaches for real-time fraud detection in US financial transactions: challenges and

opportunities. *European Journal of Computer Science and Information Technology*, *11*(6), 84-102. https://doi.org/10.37745/ejcsit.2013/vol11n684102

[6] Benchaji, I., Douzi, S., El Ouahidi, B., & Jaafari, J. (2021). Enhanced credit card fraud detection based on attention mechanism and LSTM deep model. *Journal of Big Data*, *8*, 1-21. https://doi.org/10.1186/s40537-021-00541-8

[7] Bin Sulaiman, R., Schetinin, V., & Sant, P. (2022). Review of machine learning approach on credit card fraud detection. *Human-Centric Intelligent Systems*, *2*(1), 55-68. https://doi.org/10.1007/s44230-022-00004-0

[8] Dileep, M. R., Navaneeth, A. V., & Abhishek, M. (2021). A novel approach for credit card fraud detection using decision tree and random forest algorithms. In *2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICI),* 1025-1028). IEEE. https://ieeexplore.ieee.org/abstract/document/9388431/

[9] Ghimire, A. (2024) Harnessing Big Data with AI-Driven BI Systems for Real-Time Fraud Detection in the US Banking Sector. *BULLET: Jurnal Multidisiplin Ilmu*, *3*(6), 731-743. https://www.researchgate.net/profile/Ashok-Ghimire-2/publication/389152498_Harnessing_Big_Data_with_AI-Driven_BI_Systems_for_Real-Time_Fraud_Detection_in_the_US_Banking_Sector/links/67b6edbbf5cb8f70d5b5d235/Harnessing-Big-Data-with-AI-Driven-BI-Systems-for-Real-Time-Fraud-Detection-in-the-US-Banking-Sector.pdf

[10] Habibpour, M., Gharoun, H., Mehdipour, M., Tajally, A., Asgharnezhad, H., Shamsi, A., ... & Nahavandi, S. (2023). Uncertainty-aware credit card fraud detection using deep learning. *Engineering Applications of Artificial Intelligence*, *123*, 106248. https://www.sciencedirect.com/science/article/pii/S0952197623004323

[11] Han, S., Zhu, K., Zhou, M., & Cai, X. (2021). Information-utilization-method-assisted multimodal multiobjective optimization and application to credit card fraud detection. *IEEE Transactions on Computational Social Systems*, *8*(4), 856-869. https://ieeexplore.ieee.org/abstract/document/9387114/

[12] 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. https://doi.org/10.1186/s40537-022-00573-8

[13] Islam, M. Z., Shil, S. K., & Buiya, M. R. (2023). AI-driven fraud detection in the US financial sector: Enhancing security and trust. *International Journal of Machine Learning Research in Cybersecurity and Artificial Intelligence*, *14*(1), 775-797. https://www.researchgate.net/profile/Md-Zahidul-Islam-22/publication/385817697_AI-Driven_Fraud_Detection_in_the_US_Financial_Sector_Enhancing_Security_and_Trust/links/67365b1a69c07a4114473ef7/AI-Driven-Fraud-Detection-in-the-US-Financial-Sector-Enhancing-Security-and-Trust.pdf

[14] Karthik, V. S. S., Mishra, A., & Reddy, U. S. (2022). Credit card fraud detection by modelling behaviour pattern using hybrid ensemble model. *Arabian Journal for Science and Engineering*, *47*(2), 1987-1997. https://doi.org/10.1007/s13369-021-06147-9

[15] Khalid, A. R., Owoh, N., Uthmani, O., Ashawa, M., Osamor, J., & Adejoh, J. (2024). Enhancing credit card fraud detection: an ensemble machine learning approach. *Big Data and Cognitive Computing*, *8*(1), 6. https://doi.org/10.3390/bdcc8010006

[16] Kumar, S., Gunjan, V. K., Ansari, M. D., & Pathak, R. (2022). Credit card fraud detection using support vector machine. In *Proceedings of the 2nd International Conference on Recent Trends in Machine Learning, IoT, Smart Cities and Applications: ICMISC 2021* (pp. 27-37). Springer Singapore. https://link.springer.com/chapter/10.1007/978-981-16-6407-6_3

[17] Langevin, A., Cody, T., Adams, S., & Beling, P. (2022). Generative adversarial networks for data augmentation and transfer in credit card fraud detection. *Journal of the Operational Research Society*, *73*(1), 153-180. https://doi.org/10.1080/01605682.2021.1880296

[18] Lebichot, B., Verhelst, T., Le Borgne, Y. A., He-Guelton, L., Oble, F., & Bontempi, G. (2021). Transfer learning strategies for credit card fraud detection. *IEEE access*, *9*, 114754-114766. https://ieeexplore.ieee.org/abstract/document/9512084/

[19] Malik, E. F., Khaw, K. W., Belaton, B., Wong, W. P., & Chew, X. (2022). Credit card fraud detection using a new hybrid machine learning architecture. *Mathematics*, *10*(9), 1480. https://doi.org/10.3390/math10091480

[20] Razaque, A., Frej, M. B. H., Bektemyssova, G., Amsaad, F., Almiani, M., Alotaibi, A., ... & Alshammari, M. (2022). Credit card-not-present fraud detection and prevention using big data analytics algorithms. *Applied Sciences*, *13*(1), 57. https://doi.org/10.3390/app13010057

[21] Sánchez-Aguayo, M., Urquiza-Aguiar, L., & Estrada-Jiménez, J. (2021). Fraud detection using the fraud triangle theory and data mining techniques: a literature review. *Computers*, *10*(10), 121. https://doi.org/10.3390/computers10100121

[22] Schneider, M., & Brühl, R. (2023). Disentangling the black box around CEO and financial information-based accounting fraud detection: machine learning-based evidence from publicly listed US firms. *Journal of Business Economics*, *93*(9), 1591-1628. https://doi.org/10.1007/s11573-023-01136-w

[23] Seera, M., Lim, C. P., Kumar, A., Dhamotharan, L., & Tan, K. H. (2024). An intelligent payment card fraud detection system. *Annals of operations research*, *334*(1), 445-467. https://doi.org/10.1007/s10479-021-04149-2

[24] Singh, A., Ranjan, R. K., & Tiwari, A. (2022). Credit card fraud detection under extreme imbalanced data: a comparative study of data-level algorithms. *Journal of Experimental & Theoretical Artificial Intelligence*, *34*(4), 571-598. https://doi.org/10.1080/0952813X.2021.1907795

[25] Strelcenia, E., & Prakoonwit, S. (2023). A survey on gan techniques for data augmentation to address the imbalanced data issues in credit card fraud detection. *Machine Learning and Knowledge Extraction*, *5*(1), 304-329. https://doi.org/10.3390/make5010019

[26] Strelcenia, E., & Prakoonwit, S. (2023a). Improving classification performance in credit card fraud detection by using new data augmentation. *AI*, *4*(1). https://doi.org/10.3390/ai4010008

[27] Tanouz, D., Subramanian, R. R., Eswar, D., Reddy, G. P., Kumar, A. R., & Praneeth, C. V. (2021). Credit card fraud detection using machine learning. In *2021 5th international conference on intelligent computing and control systems (ICICCS)*, 967-972. IEEE. https://ieeexplore.ieee.org/abstract/document/9432308/

[28] Van Belle, R., Baesens, B., & De Weerdt, J. (2023). CATCHM: A novel network-based credit card fraud detection method using node representation learning. *Decision Support Systems*, *164*, 113866. https://doi.org/10.1016/j.dss.2022.113866

[29] Wu, T. Y., & Wang, Y. T. (2021). Locally interpretable one-class anomaly detection for credit card fraud detection. In *2021 International Conference on Technologies and Applications of Artificial Intelligence (TAAI)* (pp. 25-30). IEEE. https://ieeexplore.ieee.org/abstract/document/9778002/

[30] Yu, C., Xu, Y., Cao, J., Zhang, Y., Jin, Y., & Zhu, M. (2024). Credit card fraud detection using advanced transformer model. In *2024 IEEE International Conference on Metaverse Computing, Networking, and Applications (MetaCom)* (pp. 343-350). IEEE. https://ieeexplore.ieee.org/abstract/document/10740150/