Journal of Harbin Engineering University
ISSN: 1006-7043

Vol 46 No. 11
November 2025

An Intelligent ABCBNLS-Based Context-Aware Credit Card Fraud Detection with User Spending Trajectory Analysis Using HFMM

Farha Anjum [1]

[1] Research Scholar, Srinivas university, SUIET, Mukka, Mangalore, Karnataka, India
Assistant Professor, Department of Intelligent Computing and Business Systems, St Joseph Engineering College, Mangalore, Karnataka, India

Abstract

Robust Credit Card Fraud Detection (CCFD) is fundamental in the context of the digital Financial Sector (FS) for averting fraudulent transactions. Nevertheless, none of the existing CCFD studies concentrated on the Category Sequence Deviation (CSD) detection and User Spending Trajectory (UST) investigation, thereby overlooking the hidden fraudulent behaviours. Therefore, a robust context-aware CCFD with UST analysis and CSD detection utilizing Adaptive Bounded Cubic Bump Neutrosophic Logic System (ABCBNLS) and Hidden Faddeeva Markov Model (HFMM) is proposed in this paper. Primarily, the user registers in the financial application, followed by digital signature creation. Then, to initiate the transaction, the registered users log in to the application. Here, the user authentication is carried out via Digital Signature Verification (DSV). Afterward, the transaction attributes are extracted; further, the extracted transaction attributes are subjected to the trained proposed CCFD model. The CCFD dataset is gathered, followed by data pre-processing, contextual feature extraction, and correlation investigation. Thereafter, the context-aware pattern is identified. Next, by utilizing HFMM, the UST analysis and CSD detection are performed. In the meantime, to examine the behavioural risk, the proposed ABCBNLS is utilized based on the contextual features. Lastly, by using the Transfer Deep Long Adaptive Hyper-Tangent Rademacher Short Term Memory (TDLAHTRSTM), the credit card fraud is detected with an accuracy of 98.99748%. Hence, the normal transactions are completed and further stored on the Blockchain (BC) through the Proof-of-Stake (PoS) protocol. Overall, the hidden fraudulent behaviours were efficiently investigated by the proposed approach with a perplexity of 1.423.

Keywords: Credit Card Fraud Detection (CCFD), Cyber Security (CS), Financial Sector (FS), Block-Chain (BC), Context-Aware Pattern Recognition (CAPR), User Spending Trajectory (UST) analysis, and Latent Transactional Anomalies (LTA).

1. Introduction

FS plays a fundamental role in the era of digital platforms for enabling secure credit systems and financial transactions (Hafez et al., 2025). The utilization of credit cards has grown exponentially with the rapid advancement of online payment platforms and digital banking (Esenogho et al., 2022). A credit card permits users to carry out financial transactions in a rapid and convenient way (Peter et al., 2023) (Sulaiman et al., 2022). In addition, by rendering increased security and transparency, BC technology improves credit card transactions via an immutable ledger and decentralized architecture (Tien et al., 2024). Nevertheless, credit card fraud and illegitimate transactions have also significantly increased. This results in severe reputational and economic risks to both financial institutions as well as consumers (Mniai et al., 2023). Hence, precise and timely CCFD systems (Chung & Lee, 2023) have evolved as a potential solution for increasing the customer's trust and diminishing the financial losses (Ileberi et al., 2022).

In the prevailing studies, Artificial Intelligence (AI) techniques, such as Deep Learning (DL) and Machine Learning (ML), were established for detecting the fraudulent patterns centered on user behaviours (Mienye & Jere, 2024). To classify the financial transactions into fraudulent or legitimate, the ML algorithms, namely Naïve Bayes (NB), decision tree, and Random Forest (RF), were utilized in the existing works (Khalid et al., 2024) (Salekshahrezaee et al., 2023). Moreover, for performing robust CCFD, the prevailing works introduced the DL algorithms like Gated Recurrent Unit (GRU), Recurrent Neural Network

(RNN), and Long Short Term Memory (LSTM) (Alarfaj et al., 2022). Also, a deep explainer was incorporated with a credit card anomaly detection model, thereby facilitating the system to interpret transactional behaviour in a human-like manner (Lebichot et al., 2021). But, the traditional models didn't capture subtle transactional anomalies and struggled to adapt to the emerging attack strategies (Yu et al., 2024).

The prevailing works failed to detect subtle anomalies owing to the absence of contextual information like transaction time, device, and location (Habibpour et al., 2023). However, none of the traditional works focused on the UST investigation and category pattern deviation detection, thus affecting the potential of the model to detect hidden fraudulent behaviour in the dynamic credit card environments. Thus, to adapt to the dynamic transaction environments, an intelligent and context-aware CCFD system with UST analysis and CSD detection is vital. This study is motivated by the need to provide an intelligent context-aware CCFD with UST investigation and CSD detection utilizing ABCBNLS and HFMM.

1.1. Problem Statement

The potential challenges faced by the existing works are provided as follows,

- None of the prevailing works focused on the UST analysis and CSD detection, thereby limiting the model's potential to understand how the spending patterns of users emerged over time across different categories. Thus, in the credit card usage environment, hidden fraudulent behaviours (legitimate-looking purchases made in unusual category sequences) might be unnoticed.
- The traditional study (Ding et al., 2023) mainly relied on previously observed fraud patterns and known users. Nevertheless, this framework wasn't generalized well to handle unseen attack patterns, thus increasing transaction misclassification.
- The conventional work (Dastidar et al., 2022) failed to perform user authentication (i.e., permitting unauthorized individuals to initiate transactions), compromising the overall security of the cybersecurity system.
- The prevailing work (Charizanos et al., 2024) didn't examine the spatial relationship between transactional factors, such as time, amount, region, and merchant category, thereby degrading the prediction rate of the detection model.
- Some of the existing methods failed to examine the user transaction behavioural boundaries (usual transaction pattern and unusual transaction pattern), degrading the anomaly detection performance.
- Due to the less efficient consensus mechanisms like Proof of Work (PoW), numerous prevailing BC-assisted approaches had slow transaction processing and high costs. Also, PoW required considerable energy to maintain security.

1.2. Research Objectives

The major objectives of the proposed method are given below,

- The proposed work establishes a novel HFMM to investigate the UST and detect the deviations in the user spending category sequence, thus capturing the latent transaction anomaly.
- To handle the unseen and emerging attack patterns, an adaptive Transfer Learning (TL) is incorporated with the proposed TDLAHTRSTM.
- During transaction initiation, the proposed Digital Sellmeier Signature Algorithm (D2SA) significantly authenticates the user, thereby reducing unauthorized user access.
- The proposed approach introduces the Pearson Correlation Coefficient (PCC) to capture the relationship betwixt the contextual factors, thus enhancing the fraud detection performance.
- To group the user's transaction patterns into usual and unusual, an effective Density Price-Based Sine-Gordan Clustering of Applications with Noise (DPBSGCAN) is used, ensuring significant anomaly detection efficiency.
- To improve the energy efficiency and scalability of the BC transaction creation and validation, a PoS consensus protocol is introduced in the research methodology.

The paper is structured as: The related works are examined in Section 2, the mathematical modelling of the proposed work is displayed in Section 3, the performance of the proposed work is evaluated in Section 4, and finally, the paper is winded up in Section 5 with future enhancements.

2. Literature Survey

(Charizanos et al., 2024) recommended an online fraud detection module centered on fuzzy logic for credit card transactions. Here, for classifying the credit card transactions into fraudulent and non-fraudulent, a robust fuzzy Logistic Regression (LR) model was used. Non-stationary changes in fraud behaviour and the imbalance issue were proficiently handled by this approach. Nevertheless, this model didn't examine the relationship among the contextual factors, thus affecting the detection rate.

(Ding et al., 2023) employed a CCFD utilizing an enhanced Variational Autoencoder Generative Adversarial Network (VAGAN). Initially, the historical fraud detection dataset was collected and further fed into data balancing. Here, to perform CCFD, an enhanced VAGAN was established. This approach had high reliability and robustness in CCFD. But, it didn't handle the unseen attack patterns, thereby depicting less generalizability.

(Mienye & Sun, 2023) presented an ensemble DL-powered data resampling approach for CCFD. Here, by utilizing a hybrid Synthetic Minority Oversampling TEchnique and Edited Nearest Neighbor (SMOTE-ENN), the gathered input dataset was balanced. Afterward, to identify the credit card fraudulent transactions, the ensemble classifiers, namely LSTM, Multi-Layer Perceptron (MLP), and GRU, were used. However, due to the utilization of a stacked ensemble approach, the framework had considerable computational overhead.

(Belle et al., 2023) propounded a robust framework called CCFD utilizing node representation learning. Initially, the CCFD dataset was collected and further fed into node representation learning. Here, for enhancing the learning efficiency of the classifier algorithm, an inductive pooling operator was established. Nevertheless, owing to the dynamic nature of fraud patterns, the model had poor classifier performance.

(Xie et al., 2023) developed a time-aware attention-assisted gated network for CCFD by extracting transactional behaviours. Here, to extract the behavioural information, a time-aware attention module was used based on the user's historical transactions. Thereafter, for learning the long-term and short-term transactional nature of the users, an RNN was utilized. The subtle fraudulent anomalies in the credit card transactions were effectively identified by this framework. However, the framework had poor learning efficiency due to the random classifier parameter initialization.

(Ghaleb et al., 2023) presented an efficient CCFD framework utilizing an ensemble-synthesized minority oversampling-centric RF algorithm and generative adversarial networks. In this work, steps like data collection, data pre-processing, data augmentation, and CCFD were included. The fraud and normal transactions were significantly predicted by the suggested approach. This approach had higher efficiency. Nevertheless, owing to the trial-and-error method, the framework had considerable time complexity.

(Raval et al., 2023) introduced a reliable, explainable LSTM approach to perform credit card fraud transactions grounded on the fraud patterns. Primarily, the historical dataset was pre-processed, followed by feature extraction and feature selection. Subsequently, to classify the credit card frauds, the LSTM was utilized. In addition, the outcome of the model was interpreted. Finally, in the BC, the final outcome was stored in a smart contract. This model had high reliability and trustworthiness. But, in real-time financial transactions, the model had less feasibility due to the enormous amount of transactions.

(Salam et al., 2024) recommended a federated learning framework for CCFD utilizing data balancing techniques. Here, in the input dataset, data balancing techniques, namely undersampling and oversampling, were applied. After that, to perform CCFD, the ensemble ML classifiers, including NB, RF, and LR, were used. The over-fitting problems were significantly reduced by this model. However, owing to the augmented dataset, the model had maximum computational complexity.

(Dastidar et al., 2022) integrated a neural feature aggregation scheme for CCFD. In this, for identifying fraudulent transactions in the credit card usage environment, the neural feature aggregation classifier was established. Moreover, to capture the most relevant transactional features, neural feature aggregation was utilized, thus enhancing the context-awareness of the CCFD model. Nevertheless, due to the lack of user authentication, the approach had poor security.

(Wang et al., 2025) employed a robust fraud detection in credit card transactions centered on temporal heterogeneous graph contrastive learning. In this framework, the key steps, namely data collection, pre-processing, and fraud detection, were included. Here, to capture the most informative features, a dual-view contrastive learning mechanism was utilized, thus enhancing the efficacy of the model. However, this model had data sparsity problems, thereby degrading the robustness of the model.

3. Proposed Methodology for a Robust Context-Aware CCFD with UST Analysis Using HFMM

Here, by utilizing HFMM, an enhanced ABCBNLS-based context-aware CCFD with UST analysis and CSD detection is proposed. The proposed ABCBNLS acts as the user behaviour interpretation layer, which depicts the deviation level of the user's transaction behaviour. Credit card fraud is significantly detected by an efficient TDLAHTRSTM, thereby diminishing the financial losses and enhancing the model's robustness. Figure 1 displays the diagrammatic format of the proposed work.

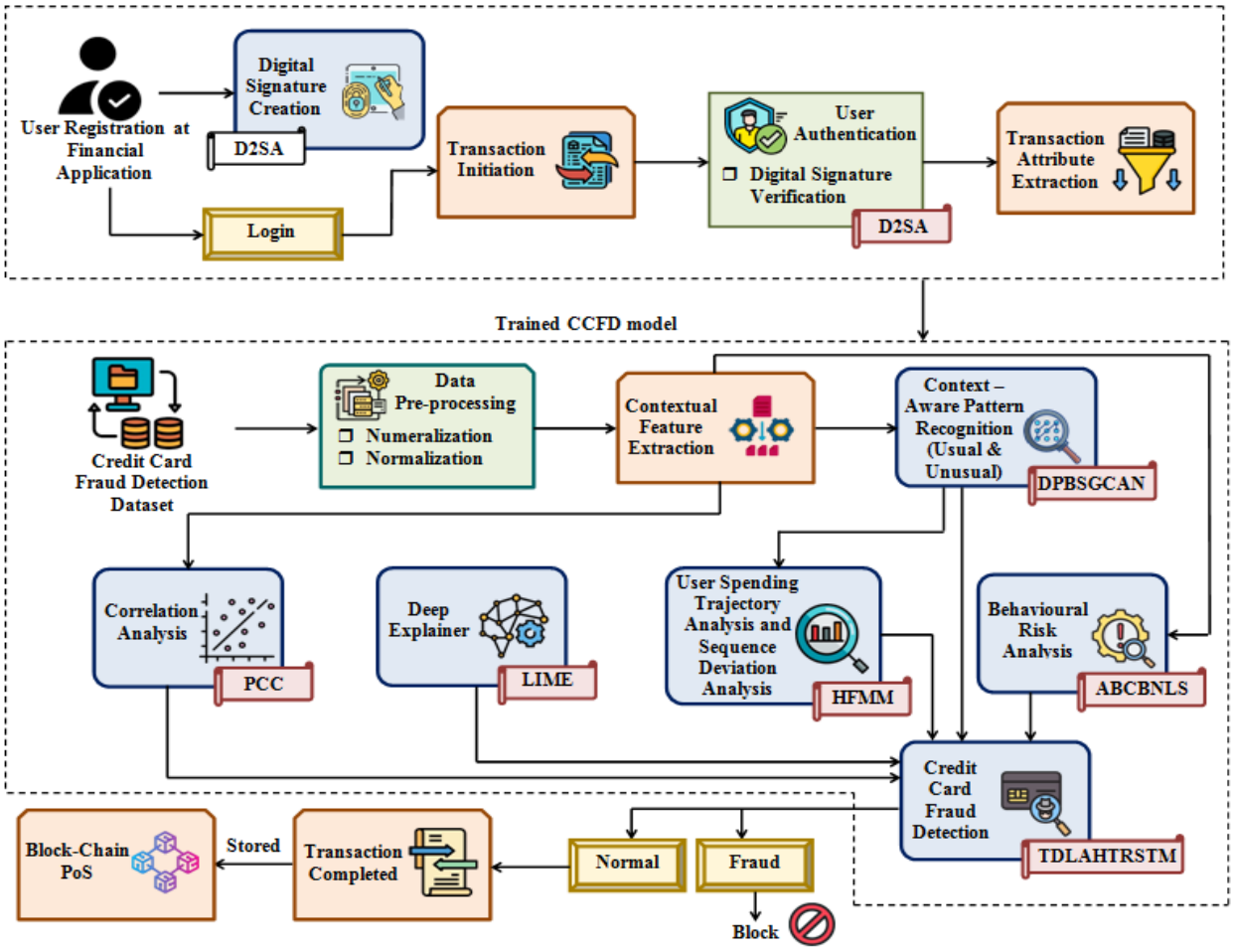

Figure 1. Pictorial depiction of the research methodology

The proposed method is generalized well enough to recognize the subtle or latent transaction anomalies via behaviour risk analysis, Context-Aware Pattern Recognition (CAPR), UST investigation, and CSD detection. The detailed overview of the proposed framework is depicted further.

3.1. User Registration at the Financial Application

Initially, the bank users register in the financial application by using their username and password.

$$\delta_u = (\delta_1, \delta_2, \ldots\ldots \delta_U)$$

Where, $u = 1 to U$ (1)

Where, the number of registered users δ_u is exemplified as U. User registration in a financial application involves verifying the identity of the user for ensuring that only the authorized user is permitted to access the financial transaction. The user's credential information (φ) (user name and password) is digitally signed during registration for future authentication purposes.

3.2. Digital Signature Creation

Afterward, a digital signature of φ is efficiently created by the proposed D2SA, thereby ensuring high financial security. The prevailing Digital Signature Algorithm (DSA) permits variable key lengths, thus balancing performance and security. Furthermore, during authentication, the DSA reduces computational load for DSV. However, the DSA is sensitive to the choice of nonce (per-message secret random number) selection. Private key exposure and signature collisions are caused by an improper nonce selection. Thus, to select a suitable nonce, the proposed framework uses the Sellmeier Equation (SE), thereby diminishing the signature collision rate. By considering the variations in the computation parameters, the SE selects the optimal nonce value, thus ensuring balanced stability and randomness.

Primarily, the prime divisor (160 bits) and the large prime number (1024 bits) are assumed to perform modular arithmetic operations for key generation.

$$|\alpha,\beta| = |1024,160| \tag{2}$$

Here, the prime number and prime divisor are notated as α and β, correspondingly. Then, centered on the selected prime number and prime divisor, the generator (γ) is computed, ensuring resistance against subgroup attacks.

$$\gamma = \varpi^{(\alpha-1)/\beta} \bmod \alpha \tag{3}$$

Subsequently, to choose an appropriate nonce value (integer) (ϖ), the proposed approach utilizes the SE.

$$\varpi(\alpha,\beta) = 1 + \sum_m \left(\frac{P_m \alpha^2}{\wp(\beta^2) - Q_m} \right) \tag{4}$$

Where, the modular function is notated as $\bmod$, the variations in the computation parameters are signified as $\wp$, the constant value is denoted as m, and the Sellmeier coefficients are exemplified as (P,Q). Thereafter, to sign the information, the private key (ρ) and public key (κ) are generated. Here, by considering the prime divisor, the private key $(1 < \rho < \beta)$ is randomly selected.

$$\kappa = \gamma^{\rho} \bmod \alpha \tag{5}$$

In the meantime, to create a hash value (λ) of the φ, the cryptographic hash function is applied. After that, the signature creation parameters (I,J) are calculated by utilizing the following expressions,

$$I = (\gamma^{l} \bmod \alpha) \bmod \beta \tag{6}$$

$$J = l^{-1}(\lambda + \rho I) \bmod \beta \tag{7}$$

Where, the random number is indicated as l. Next, to create the digital signature (τ), the I and J are paired.

$$\tau = (I,J) \cdot \varphi \tag{8}$$

When the user initiates the financial transaction, the digital signature is validated during user authentication based on the signature pair and the user's credential information. Conditions like if $(0 < I < \beta)$ and $(0 < J < \beta)$ are considered to check the validity of the signature components. If both conditions are true, then the signature components are within the valid range; or else, the signature components are rejected in the course of verification. Then, the digital signature is verified as,

$$x = J^{-1} \bmod \beta \tag{9}$$

$$y = (\lambda \cdot x) \bmod \beta \tag{10}$$

$$z = (I \cdot x) \bmod \beta \tag{11}$$

$$a = ((\gamma^{y} \cdot \kappa^{z}) \bmod \alpha) \bmod \beta \tag{12}$$

Here, the signature verification parameters are implied as (x,y,z,a).

$$\begin{cases} \text{If}(a == I), & \text{valid} \\ \text{Else}, & \text{invalid} \end{cases} \quad (13)$$

If a is equal to I, then the signature is valid; or else, the signature is invalid. Once DSV is successfully done, the user is permitted to proceed with the transaction.

3.3. Login

After registering in the financial application, the user logs in to the application for initiating the financial transaction across numerous categories.

3.4. User Authentication

By utilizing the proposed D2SA, the DSV is carried out during transaction initiation for authenticating the legitimacy of the user. The steps involved in the proposed D2SA are discussed in Section 3.2. The process of verifying the authenticity of the user is termed user authentication, which permits legitimate users to access the financial application. Moreover, user authentication assists in reducing fraudulent transactions.

3.5. Transaction Attributes Extraction

To detect the fraudulent activity, the transaction attributes (ν), such as category, amount, first name, last name, transaction number, and Unix time, are extracted from the authenticated users. By using a trained model, the proposed CCFD is tested centered on the extracted transaction attributes ν.

3.6. Credit Card Fraud Detection Framework

The historical data is gathered to train the proposed CCFD. Moreover, the proposed CCFD involves various processes, namely pre-processing, contextual feature extraction, CAPR, UST analysis and CSD detection, and CCFD. The CCFD is generalized well enough to predict credit card fraud, thereby enhancing financial security.

3.6.1. Credit Card Fraud Detection Dataset

Initially, to train the proposed CCFD, the historical CCFD dataset is collected from publicly available sources.

$$\psi_f = [\psi_1, \psi_2, \ldots\ldots \psi_F] \text{ Here, } f = 1,2,\ldots F \quad (14)$$

Here, the number of collected input data ψ_f is notated as F.

3.6.2. Data Pre-Processing

For numeralization and normalization, the ψ_f is pre-processed to enhance the computational efficiency and quality of the data. The procedure of transforming the non-numerical values into numerical vectors for better investigation is referred to as numeralization. By utilizing min-max normalization, the numerical values (ς_{num}) are represented within the range of 0 and 1 in normalization.

$$\eta = \frac{\varsigma_{\text{num}} - \min(\varsigma_{\text{num}})}{\max(\varsigma_{\text{num}}) - \min(\varsigma_{\text{num}})} \quad (15)$$

Where, the pre-processed data (normalized data) is signified as η.

3.6.3. Contextual Feature Extraction

After that, from η, the contextual features (χ), namely category, amount, gender, first name, last name, street, state, city, zip, longitude, latitude, transaction number, date of birth, job, and unix time, are extracted for enhancing the model's potential to recognize the hidden transaction anomalies.

3.6.4. Context-Aware Pattern Recognition

Next, to categorize the transaction patterns as usual and unusual, the χ is subjected to the proposed DPBSGCAN, thus improving the context-awareness of the proposed CCFD model. The Density-Based Spatial Clustering of Applications with Noise (DBSCAN) automatically groups similar patterns without requiring the number of clusters in advance. Also, it has adaptability and high robustness. However, DBSCAN is sensitive to the choice of clustering parameters, such as the minimum number of points (MinPts) and epsilon. Improper parameter selection leads to suboptimal clustering results. Hence, in the proposed work, the Price Equation (PE) is used to select the epsilon parameter. By quantifying variation in the cluster compactness, the PE adaptively determines the epsilon parameter. The research method establishes the Sine-Gordon Formula (SGF) to choose the proper minpts. By mathematically balancing density and distribution, the SGF selects an optimal minpts. Thus, the clustering outcomes are enhanced.

The clustering parameters, such as epsilon and MinPts, are calculated for initiating the cluster process. MinPts refers to the minimum number of points required within the epsilon's neighbourhood, whereas epsilon

signifies the maximum distance betwixt two points that refers to the neighborhood radius. Further, the PE is used to set the epsilon parameter (∂).

$$\partial = \frac{1}{O}\mathrm{cov}(\hbar_e, \chi) + \frac{1}{O} S(\hbar_e \sigma \chi) \tag{16}$$

Here, the average fitness score for all clusters is implied as O, the covariance between data points is notated as cov, the clustering quality score is exemplified as $\hbar_e$, the weight factor is indicated as S, and the variation factor is represented as σ.

Likewise, centered on the SGF, the MinPts (υ) is selected.

$$\upsilon = \chi_{aa} - \chi_{bb} + \sin \chi \tag{17}$$

Where, the space and time coordinates are represented as (a, b), correspondingly, and the sine function is denoted as $\sin$. Thereafter, grounded on the calculated epsilon and MinPts, the points are classified into core points (μ), border points (b), and noise (V). A point that has enough neighbours around it within the epsilon radius is termed a core point, whereas a noise point is a point that is too far away from the others. Moreover, a border point refers to the point that doesn't have enough neighbours by itself; however, it is close enough to a core point.

By adding the closest data points, a new cluster is created based on the core point. Subsequently, the epsilon neighbourhood is examined for every single point added to the cluster. If a neighbour (ϑ) is also a border point or core point, then the neighbour is included in the same cluster; otherwise, the noise points are isolated from the clustering space.

$$\begin{cases} \mathrm{If}(\vartheta = \mu \mid b), & \mathrm{add} \\ \mathrm{If}(\vartheta = V), & \mathrm{isolate} \end{cases} \tag{18}$$

Therefore, the clustering process is iteratively repeated till all the data points are marked as visited. The contextual behaviors, which are isolated from dense clusters, are assumed as unusual patterns. The proposed work facilitates precise anomaly detection modeling and subtle fraudulent behaviour identification by grouping the user's transaction patterns as usual and unusual. Here, (Ξ) represents the transaction pattern grouped features.

3.6.5. UST Analysis and CSD Detection

Then, to capture hidden fraudulent behaviours in the dynamic credit card usage environments, the Ξ is fed into the UST analysis and CSD detection. Generally, a consistent category sequence (home – utilities – groceries – travel) or spending pattern is maintained by each credit card user. Nevertheless, to avoid detection, the fraudsters perform legitimate-looking purchases in usual categories. The deviation in the category sequence of the user (hidden anomaly) signifies a sign of an emerging fraud pattern (account takeover or account misuse). The reliability of the context-aware CCFD is degraded by the hidden anomalous nature. Thus, to reveal the latent behavioural shifts, UST analysis and CSD detection are carried out, thereby improving the context-awareness in fraud detection. In the proposed work, the HFMM is introduced to examine UST and CSD. The Hidden Markov Model (HMM) efficiently models sequential dependencies in user transactions, thus capturing typical category sequences over time. It also accommodates uncertainty and variability in spending behavior, thereby rendering probabilistic insights for robust trajectory investigation. But, slow convergence and a local optimal solution are caused by the random initial state probability assignment. By providing a smooth and probabilistic mapping from feature scores to valid probability values, the Faddeeva Function (FF) assigns initial state probabilities. Hence, for assigning the initial state probability, the proposed work utilizes the FF.

For every single user, the user's spending pattern or category is obtained from Ξ. Initially, to learn the underlying behavioural patterns, the user's transaction sequences from Ξ are signified as an observation (D).

$$D = (D_1, D_w, \ldots\ldots D_W) \tag{19}$$

Here, the number of observations at time w is notated as D_w and the final time step is implied as W. After that, the hidden behavioural states (R) of D_w are defined to signify a spending intention of the user at a time w, thus capturing the category choice of the user.

$$R(D_w) \rightarrow \sum_{w=1}^{W} R_w \tag{20}$$

Where, the hidden state (routine, seasonal, and unusual) at a time w is represented as R_w.

Then, in the proposed work, the FF is utilized to evaluate the initial state probability $\left(\widetilde{Y}_o\right)$, thus revealing the likelihood of the category sequence at the initial state.

$$\widetilde{Y}_o\left(R_w\right)=\exp^{-w^2}\mathrm{er}\left(-\ddot{U}R_w\right)\cdot\hat{\Im}\left(R_w=\hat{n}\right) \quad (21)$$

Here, the likelihood factor is indicated as $\hat{\Im}$, the exponential function is notated as $\exp$, and the complementary error function with $\ddot{U}$ imaginary argument is implied as er. Next, to learn the probability of a user's sequence moving from one state to another, the transition probabilities are calculated. The transition probabilities depict how the user's behaviour emerges over time, capturing the trajectory of behavioural transitions.

$$\widetilde{Y}_{\hat{m},\hat{n}}=\widetilde{Y}_o\cdot\hat{\Im}\left(R_{w+1}=\hat{n}\mid R_w=\hat{m}\right) \quad (22)$$

Where, the probability of transitioning from state $\hat{m}$ to state $\hat{n}$ is denoted as $\widetilde{Y}_{\hat{m},\hat{n}}$. Also, regarding the state $\hat{m}$, the emission probabilities $\left(E'''_{\hat{m}}\right)$ are calculated to identify the chance of a category sequence. For instance, travel and luxury are common during the unusual state, and groceries and fuel are common during the routine state. The visible spending category, with a certain probability, assists in observing the hidden behaviours.

$$E'''_{\hat{m}}\left(D_w\right)=\hat{\Im}\left(D_w\mid R_w=\hat{m}\right) \quad (23)$$

Temporal evolution analysis is performed by considering the probability values to recognize the most likely state path that signifies the user's emerging behaviour over time.

Afterward, by analogizing the probability values of the new category sequence and the learned category sequence, the deviation detection is done. If a new category sequence doesn't fit the learned transition pattern, then the model assigns a low probability. Here, the low probability patterns represent an abnormal category sequence (hidden fraud).

By learning the hidden spending behaviors, the proposed HFMM investigates the emerging category sequence over time. The proposed HFMM identifies the abnormal evolution pattern (unusual category sequence), which deviates from a user's learned trajectory, by tracking the behavioural transitions.

Here, $\left(\psi''\right)$ represents the outcome of the proposed HFMM.

3.6.6. Behavioural Risk Analysis

Thereafter, by utilizing an intelligent ABCBNLS, the behavioural risk analysis is carried out based on the contextual features χ for recognizing the latent fraudulent behaviours. The prevailing Neutrosophic Logic System (NLS) automatically learn patterns from uncertain and vague inputs. It also renders a transparent and rule-centric structure that gives insight into the behavioural deviation level evaluation process. By facilitating human-like simulation (cognitive/ intelligent), the fuzzy logic component makes the system's behaviour interpretable. However, due to the improper membership function, the NLS has tuning difficulty. Thus, to diminish the tuning problems in the fuzzy logic system, the NLS introduces the Bounded Cubic Bump (BCB) membership function. By rendering a smooth and finite-support shape, the BCB enhances fuzzy performance. In this, a smooth and finite-support shape reduces abrupt transitions.

To calculate the Behavioural Risk Score (BRS), the factors, such as transaction amount $\left(T_{\mathrm{amt}}\right)$, transaction frequency $\left(T_{\mathrm{fq}}\right)$, transaction time $\left(T_{\mathrm{time}}\right)$, and transaction region, are considered from contextual features χ. The normalized transaction amount score $\left(\ddot{T}_{\mathrm{amt}}\right)$, normalized transaction frequency score $\left(\ddot{T}_{\mathrm{fq}}\right)$, normalized transaction time score $\left(\hat{T}_{\mathrm{time}}\right)$, and normalized transaction region score $\left(\hat{T}_{\mathrm{reg}}\right)$ are calculated by varying the ranges (low, medium, and high) of the contextual factors.

$$\ddot{T}_{\mathrm{amt}}=\begin{cases}\mathrm{if}\left(T_{\mathrm{amt}}=0-5000\right), & T_{\mathrm{amt}}/5000\\ \mathrm{if}\left(T_{\mathrm{amt}}=5000-20{,}000\right), & 0.25+0.75*\left(T_{\mathrm{amt}}-5000\right)/15000\\ \mathrm{if}\left(T_{\mathrm{amt}}>20000\right) & 1\end{cases} \quad (24)$$

$$\ddot{T}_{\mathrm{fq}}=\begin{cases}\mathrm{if}\left(T_{\mathrm{fq}}=0-3\right), & T_{\mathrm{fq}}/3\\ \mathrm{if}\left(T_{\mathrm{fq}}=3-7\right), & 0.25+0.75*\left(T_{\mathrm{fq}}-3\right)/4\\ \mathrm{if}\left(T_{\mathrm{fq}}>7\right), & 1\end{cases} \quad (25)$$

$$\hat{T}_{\text{time}}=\begin{cases}\text{if}\,(T_{\text{time}}=0-2), & T_{\text{time}}/2\\ \text{if}\,(T_{\text{time}}=2-5), & 0.25+0.75*(T_{\text{time}}-2)/3\\ \text{if}\,(T_{\text{time}}>5), & 1\end{cases} \tag{26}$$

$$\hat{T}_{\text{reg}}=\begin{cases}\text{if}\,(T_{\text{reg}}=0-50), & T_{\text{reg}}/50\\ \text{if}\,(T_{\text{reg}}=50-200), & 0.25+0.75*(T_{\text{reg}}-50)/150\\ \text{if}\,(T_{\text{reg}}>200), & 1\end{cases} \tag{27}$$

The aforementioned normalization scores are obtained by considering the varying ranges of the contextual factors, such as low range, medium range, and high range. Moreover, the normalization score is varied for every single range. If the T_{amt} is within the range of 0 to 5000, then the $\ddot{T}_{\text{amt}}$ is obtained by utilizing $T_{\text{amt}}/5000$. Further, by summing the normalized score values, the weighted BRS score $(\tilde{\beta})$ is computed.

$$\tilde{\beta}=0.25*\left(\ddot{T}_{\text{amt}}+\ddot{T}_{\text{fq}}+\hat{T}_{\text{time}}+\hat{T}_{\text{reg}}\right) \tag{28}$$

The behavioural risk level is evaluated into three categories, such as low, medium, and high, centered on $\tilde{\beta}$. Primarily, by utilizing the BCB membership function (Ω_{mem}), the crisp inputs $\tilde{\beta}$ are transformed into neutrosophic triples (falsity-membership, indeterminacy-membership, and truth-membership). The falsity-membership is the degree to which $\tilde{\beta}$ is not a member of the set, whereas indeterminacy-membership is the degree of uncertainty. Furthermore, truth-membership reveals the actual membership degree to the fuzzy set.

$$\Omega_{\text{mem}}(\tilde{\beta})=\begin{cases}1-\left(\dfrac{\tilde{\beta}-\text{ct}}{\tilde{\lambda}}\right)^3, & \text{if}\left(\left|\tilde{\beta}-\text{ct}\right|\le\tilde{\lambda}\right)\\ 0, & \text{else}\end{cases} \tag{29}$$

Where, the neutrosophic triples of Ω_{mem} are exemplified as $(\tilde{\beta}_1,\tilde{\beta}_2,\tilde{\beta}_3)$, the center of the curve is implied as ct, and the half-width of the curve is notated as $\tilde{\lambda}$. Subsequently, to evaluate the behavioural risk level, if-then rules (Rl) are framed utilizing neutrosophic terms.

$$\text{Rl}=\begin{cases}\text{If}(\tilde{\beta}_1,\tilde{\beta}_2,\tilde{\beta}_3)<0.33, & \text{then low}\\ \text{If}(\tilde{\beta}_1,\tilde{\beta}_2,\tilde{\beta}_3)=0.34\,\text{to}\,0.66, & \text{then medium}\\ \text{If}(\tilde{\beta}_1,\tilde{\beta}_2,\tilde{\beta}_3)>0.66, & \text{then high}\end{cases} \tag{30}$$

If $(\tilde{\beta}_1,\tilde{\beta}_2,\tilde{\beta}_3)$ is less than 0.33, then behavioural risk is low. If $(\tilde{\beta}_1,\tilde{\beta}_2,\tilde{\beta}_3)$ is within the range of 0.34 to 0.66, then behavioural risk is medium. If $(\tilde{\beta}_1,\tilde{\beta}_2,\tilde{\beta}_3)$ is greater than 0.66, then behavioural risk is high. Here, in the decision engine, the fuzzy operations are done. Then, the multiple input triples per rule are aggregated; also, the outcomes of all the rules are combined into a single outcome. Lastly, in the deneutrosophication, the fuzzy data is converted into a crisp value. The pseudo-code of the proposed ABCBNLS is provided as follows,

Input: Contextual features χ

Output: Behavioral risk level

Begin

Initialize χ, T_{amt}, T_{fq}, T_{time} and T_{reg}

For 1 to each χ do,

Compute normalized score values

Calculate weighted BRS score

$$\tilde{\beta}=0.25*\left(\ddot{T}_{\text{amt}}+\ddot{T}_{\text{fq}}+\hat{T}_{\text{time}}+\hat{T}_{\text{reg}}\right)$$

#neutrosophication

Apply BCB membership function

$$\Omega_{\text{mem}}(\tilde{\beta})=\begin{cases}1-\left(\dfrac{\tilde{\beta}-\text{ct}}{\tilde{\lambda}}\right)^3, & \text{if}\left(\left|\tilde{\beta}-\text{ct}\right|\le\tilde{\lambda}\right)\rightarrow(\tilde{\beta}_1,\tilde{\beta}_2,\tilde{\beta}_3)\\ 0, & \text{else}\end{cases}$$

Generate fuzzy rules

If $(\tilde{\beta}_1,\tilde{\beta}_2,\tilde{\beta}_3)$ < 0.33

{

Low

}

If $(\tilde{\beta}_1,\tilde{\beta}_2,\tilde{\beta}_3)$ = 0.34-0.66

{

Medium

}

If $\left(\tilde{\beta}_1, \tilde{\beta}_2, \tilde{\beta}_3\right)_{>}$ 0.66

{

High

}

End If

Aggregate fuzzy outcomes

Perform deneutrosophication

End For

Return behavioural risk level

End

Therefore, $\left(\kappa_{\text{out}}\right)$ represents the behavioural risk analyzed outcome.

3.6.7. Correlation Analysis

In the meantime, by using the PCC, the correlation analysis is performed in χ, thus capturing the meaningful relationship among the contextual features. The PCC is simple and computationally effective. The correlation investigation assists in upgrading the proposed CCFD model's learning efficiency.

$$C_{\otimes} = \frac{\sum_{a=1}^{A}\left(\chi_1 - \Re''\right)\left(\chi_A - \wp''\right)}{\sqrt{\sum_{a=1}^{A}\left(\chi_1 - \Re''\right)^2}\sqrt{\sum_{a=1}^{A}\left(\chi_2 - \wp''\right)^2}} \tag{31}$$

Where, the correlated outcome is represented as $C_{\otimes}$, the number of contextual features is indicated as $a = 1 to A$, and the mean and standard deviation of the features are exemplified as $\left(\Re'', \wp''\right)$, correspondingly.

3.6.8. Credit Card Fraud Detection

Afterward, the Ξ, Ψ'', κ_{out}, and $C_{\otimes}$ are fed into the proposed TDLAHTRSTM. The credit card transaction is classified into legitimate and fraud by the proposed TDLAHTRSTM. The Deep Long Short-Term Memory (DLSTM) is efficiently designed to handle sequential data, thereby making it particularly useful for fraud detection. However, due to the basic activation function, DLSTM has vanishing and exploding gradient problems. Moreover, it is sensitive to overfitting problems owing to improper regularization of weight values. Hence, to enhance learning efficiency, the proposed model uses the Adaptive Slope Hyperbolic Tangent (ASHT) activation function. The ASHT adaptively adjusts its slope during training, thus facilitating effective gradient flow. Additionally, in the proposed work, the Rademacher Complexity Regularization (RCR) is employed for addressing the overfitting issue. By penalizing overly complex weight configurations, the RCR selects optimal weight values, thereby reducing overfitting problems. Also, to adapt to the dynamic and emerging attack patterns, the TL is incorporated with the proposed neural network, depicting the generalizability of the model. Figure 2 displays the diagrammatic illustration of the proposed TDLAHTRSTM.

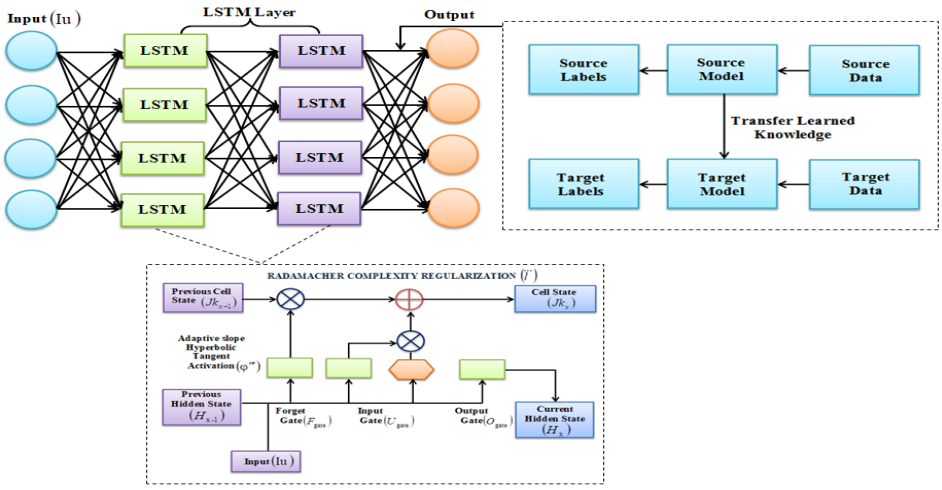

Figure 2. Structural design of the proposed TDLAHTRSTM

The proposed TDLAHTRSTM involves multiple LSTM layers for capturing the long-term dependencies and sequential information. Initially, the inputs (Iu), namely Ξ, ψ'', κ_{out}, and $C_{\otimes}$, and the previous hidden state (H_{x-1}) are transferred to the forget gate (F_{gate}) for determining what information is eradicated from the previous cell state (Jk_{x-1}).

$$F_{\mathrm{gate}} = \phi'''(\mathrm{Iu}, H_{x-1}) \cdot \dddot{\ell} + \nabla_{\mathrm{bias}} \tag{32}$$

$$\phi'''(\mathrm{Iu}) = \tanh(\mathrm{tr} \cdot \mathrm{Iu}) \tag{33}$$

Here, the ASHT activation function is exemplified as ϕ''', the tangent activation function is represented as $\tanh$, the bias of the model is denoted as ∇_{bias}, and the trainable parameter is notated as tr. Furthermore, to regularize the weight value $(\ddot{\ell})$, the proposed work utilizes the RCR.

$$\ddot{\ell} = \mathrm{Ls} + \tilde{\upsilon}\mathrm{Iu}(H_{x-1}) \tag{34}$$

Where, the loss value is implied as Ls and the regularization parameter is indicated as $\tilde{\upsilon}$. Moreover, the input gate (U_{gate}) decides the information that needs to be added to the cell state.

$$U_{\mathrm{gate}} = \phi'''(\dddot{\ell} \times (\mathrm{Iu}, H_{x-1})) + \nabla_{\mathrm{bias}} \tag{35}$$

Grounded on the input gate, the candidate cell state $(\vec{\mathrm{Jk}}_x)$ and the cell state are calculated to hold the information for a long time.

$$\vec{\mathrm{Jk}}_x = \tanh(\dddot{\ell} * (\mathrm{Iu}, H_{x-1}) + \nabla_{\mathrm{bias}}) \tag{36}$$

$$\mathrm{Jk}_x = F_{\mathrm{gate}} * \mathrm{Jk}_{x-1} + U_{\mathrm{gate}} * \vec{\mathrm{Jk}}_x \tag{37}$$

Similarly, the output gate (O_{gate}) is utilized for deciding the outcome with the highest probability.

$$O_{\mathrm{gate}} = \phi''' \cdot ((\mathrm{Iu}, H_{x-1}) \times v + \nabla_{\mathrm{bias}}) \tag{38}$$

Finally, regarding the output gate and cell state, the hidden state (H_x) is evaluated.

$$H_x = O_{\mathrm{gate}} * \tanh(\mathrm{Jk}_x) \tag{39}$$

Next, to handle the unseen attack signatures, the TL is used by learning from the source model trained on the other task. Therefore, the transactions are efficiently classified into fraud and normal by the proposed TDLAHTRSTM. The pseudo-code of the proposed TDLAHTRSTM is provided as follows,

Input: Ξ, ψ'', κ_{out} and $C_{\otimes}$

Output: Credit card fraud detection

Begin

Initialize Iu, F_{gate}, U_{gate}, O_{gate} and ϕ'''

For 1 to each input do,

Transmit Iu to forget gate

Execute $F_{\mathrm{gate}} = \phi'''(\mathrm{Iu}, H_{x-1}) \cdot \dddot{\ell} + \nabla_{\mathrm{bias}}$

Activate ASHT function,

$$\phi'''(\mathrm{Iu}) = \tanh(\mathrm{tr} \cdot \mathrm{Iu})$$

Regularize weight value

$$\ddot{\ell} = \mathrm{Ls} + \tilde{\upsilon}\mathrm{Iu}(H_{x-1})$$

Perform input gate U_{gate}

Compute cell state

$$\mathrm{Jk}_x = F_{\mathrm{gate}} * \mathrm{Jk}_{x-1} + U_{\mathrm{gate}} * \vec{\mathrm{Jk}}_x$$

Evaluate

$$O_{\mathrm{gate}} = \phi''' \cdot ((\mathrm{Iu}, H_{x-1}) \times v + \nabla_{\mathrm{bias}})$$

Estimate hidden state H_x

Integrate transfer learning

Classify credit card fraud

End For

Return fraud or normal

End

Furthermore, the normal transaction is completed and further updated on the BC in an efficient manner, whereas the fraudulent transactions are blocked from the application.

Then, the outcome of the proposed TDLAHTRSTM is interpreted in the subsequent section.

3.6.9. Deep Explainer

In this section, the Local Interpretable Model-agnostic Explanation (LIME) is used for describing the prediction C^{cd} made by the proposed neural network, thus increasing the transparency and trust of the model. Primarily, from the input samples Iu, the instances are chosen. After that, to create a new dataset of perturbed samples, the selected instances are augmented. Furthermore, the weight values are assigned to every single instance. Thereafter, to train the local interpretable model, the weighted dataset is used. Also, for reflecting the informative features, the coefficient of the local model is interpreted.

3.7. Blockchain

Here, in BC, the completed legitimate transactions are updated via the PoS protocol. For financial transactions, BC offers decentralized and immutable storage. For verifying the legitimacy of the transaction, the proposed work utilizes the PoS consensus protocol. Moreover, to create new blocks, PoS utilizes a randomly selected validator's stake. Thereafter, the verified transaction is added to a BC. Hence, the distributed ledger is shared across the network.

Overall, in the proposed work, an intelligent and context-aware latent anomaly recognition and precise CCFD module is rendered, thereby enhancing the security of the dynamic credit card usage environments.

4. Results and Discussion

Here, to depict the model's reliability in CCFD, the proposed work's performance is assessed. Also, the proposed work is implemented on the working platform of PYTHON.

4.1. Dataset Description

By utilizing the Credit Card Transactions Fraud Detection Dataset (CCTFDD), the proposed CCFD framework is evaluated. Also, the source link of the CCTFDD is given in the reference section. In the CCTFDD, the crucial transaction information, namely transaction amount, time, region, and category, is included. Hence, credit card fraud is efficiently predicted. Table 1 depicts the data distribution of the CCTFDD.

Table 1. Data samples of the CCTFDD

Classes/samples	Training	Testing
Normal	1042569	553574
Fraud	6006	2145
Total (1604294)	1048575	555719

Concerning normal and fraud, the CCTFDD includes the training set and testing set.

4.2. Performance Validation for UST Analysis and CSD Detection

Regarding UST analysis and CSD detection, the proposed HFMM's performance is assessed by comparing it with numerous existing algorithms, such as HMM, Conditional Random Field (CRF), Maximum-Likelihood Continuity Mapping (MLCM), and Kalman Filter (KF).

Table 2. Performance analysis of the proposed HFMM

Techniques	Perplexity	Kullback-Leibler (KL) divergence	Mean Absolute Percentage Error (MAPE) (%)
Proposed HFMM	1.423	0.284	4.522
HMM	5.533	0.378	8.578
CRF	7.482	0.697	9.243
KF	8.578	0.773	10.4323
MLCM	9.749	0.874	14.5903

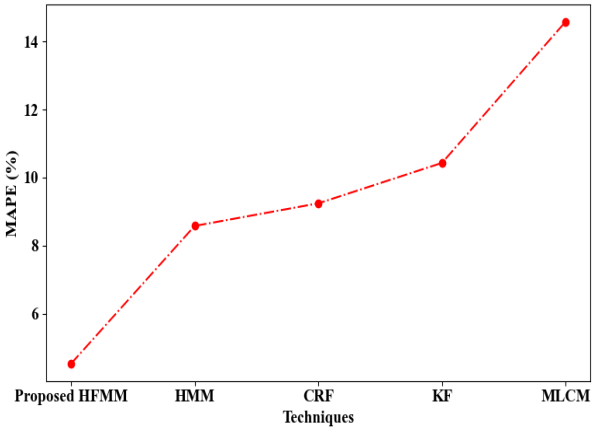

Figure 3. MAPE analysis

The performance validation of the proposed HFMM and conventional algorithms regarding the perplexity, KL-divergence, and MAPE is depicted in Table 2 and Figure 3. A perplexity of 1.423, a KL-divergence of 0.284, and an MAPE of 4.522% were attained by the proposed HFMM. Moreover, the traditional MLCM obtained perplexity, KL-divergence, and MAPE of 9.749, 0.874, and 14.5903%, respectively. However, due to the improper initial probability assignment, the prevailing approaches had suboptimal outcomes. Therefore, the efficiency of the proposed methodology was enhanced by the presence of FF-centric initial probability assignment.

4.3. Numerical Investigation for CCFD

In this section, the performance analysis is carried out regarding several quality factors, namely accuracy, precision, recall, and computational complexity, for proving the supremacy of the model in CCFD.

Table 3. Empirical evaluation for CCFD

Algorithms	Accuracy (%)	Precision (%)	Recall (%)	F-measure (%)	Training time (ms)	Cohen's kappa
Proposed TDLAHTRSTM	98.9974	98.9867	98.9974	98.9897	59308	0.9887
DLSTM	96.4672	95.7946	96.8694	95.8907	87362	0.9578
GRU	94.2646	90.4792	94.8923	92.7663	94892	0.9248
RNN	92.7582	89.4822	92.5789	90.7883	104222	0.9053
Probabilistic Neural Network (PNN)	90.5893	85.2904	90.7648	89.6754	113022	0.7522

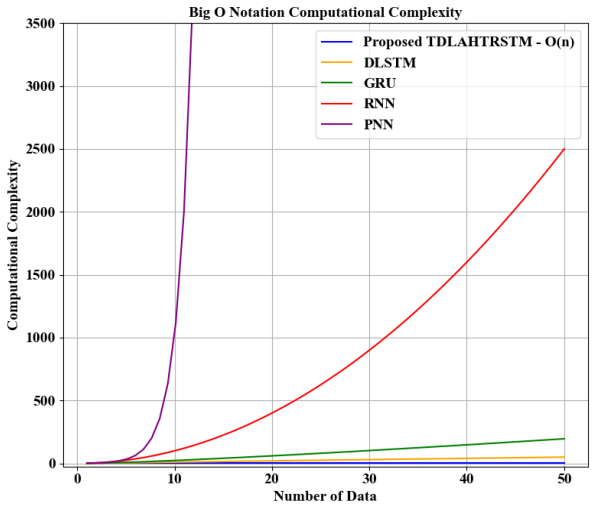

Figure 4. Computational complexity

In Table 3 and Figure 4, the proposed TDLAHTRSTM's performance is estimated by analogizing it with existing classifiers, such as DLSTM, PNN, GRU, and RNN. The learning efficiency of the classifier neurons was enhanced by the inclusion of the ASHT activation function. The proposed TDLAHTRSTM obtained an accuracy of 98.9974%, recall of 98.9974%, precision of 98.9867%, f-measure of 98.9897%, training time of 59308ms, and Cohen's kappa of 0.9887. Nevertheless, the conventional algorithms attained an average accuracy, recall, precision, f-measure, training time, and Cohen's kappa of 93.5198%, 93.7763%, 90.2616%, 98.2802%, 99874ms, and 0.8850, respectively. As the number of data increased, the computational complexity of the proposed TDLAHTRSTM remained stable. However, the prevailing approaches had a sudden increase in complexity. Here, due to the vanishing gradient problems, the existing classifiers had poor performance. Hence, the proposed method's supremacy in CCFD was depicted.

4.4. Experimental Assessment for Behavioural Risk Level Analysis

Here, to depict the model's efficiency in behavioural risk level analysis, the time complexity of the proposed ABCBNLS and prevailing techniques, namely NLS, triangular-fuzzy, trapezoidal-fuzzy, and sigmoid-fuzzy, is estimated.

Table 4. Performance analysis for behavioural risk level assessment

Algorithms	Neutrosophication time (ms)	Deneutrosophication time (ms)	Rule generation time (ms)
Proposed ABCBNLS	423	431	872
NLS	642	652	997
Triangular-fuzzy	894	885	1104
Trapezoidal-fuzzy	911	922	1582
Sigmoid-fuzzy	1049	1065	1959

Table 4 presents the performance analysis for behavioural risk level analysis. The proposed ABCBNLS took neutrosophication time, deneutrosophication time, and rule generation time of 423ms, 431ms, and 872ms, correspondingly. However, due to the tuning problems of the membership function, the traditional algorithms had considerable time consumption. Therefore, owing to the presence of the BCB membership function, the proposed method had lower time complexity and higher supremacy.

4.5. Performance Analysis for Pattern Recognition

The performance of the proposed DPBSGCAN for pattern recognition is evaluated regarding metrics like clustering time, silhouette score, and Davies-Bouldin index.

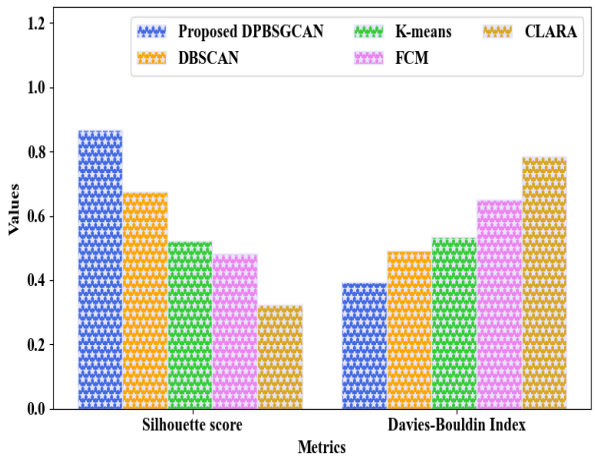

Figure 5. Experimental analysis for pattern recognition

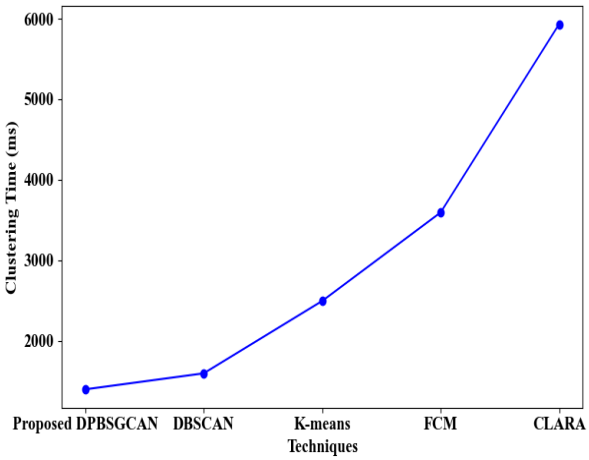

Figure 6. Clustering time

The performance of the proposed DPBSGCAN and traditional approaches, such as DBSCAN, Fuzzy C-Means (FCM), K-means, and Clustering of LARge Applications with Noise (CLARA), concerning pattern recognition is shown in Figures 5 and 6. The proposed DPBSGCAN had a silhouette score of 0.8675, Davies-Bouldin index of 0.3922, and clustering time of 1394ms. Likewise, the traditional DBSCAN had a silhouette score, Davies-Bouldin index, and clustering time of 0.6742, 0.4922, and 1593ms, respectively. The prevailing approaches had poor clustering efficiency due to improper clustering parameter initialization. Hence, owing to the utilization of SGF and PE, the proposed method had higher dominance in pattern recognition.

4.6. Empirical Validation for Digital Signature Creation

In this section, a performance evaluation is carried out for demonstrating the significance of the proposed D2SA in digital signature creation.

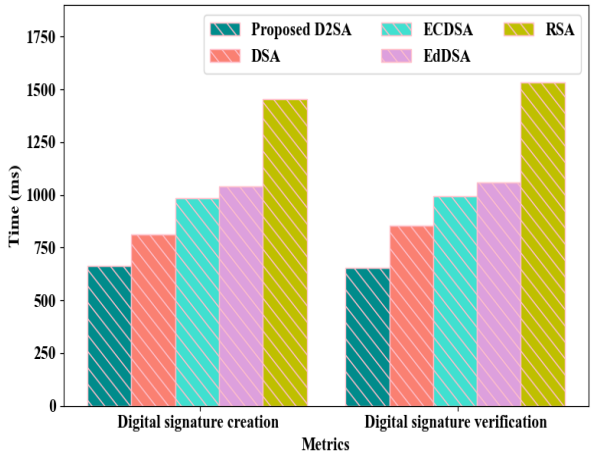

Figure 7. Performance assessment for digital signature creation

In Figure 7, the proposed D2SA's performance is analogized with conventional methods, namely Elliptic Curve DSA (ECDSA), Rivest-Shamir-Adleman (RSA), DSA, and Edwards-Curve DSA (EdDSA), to illustrate the model's reliability in digital signature creation. The proposed D2SA took a digital signature creation time of 663ms and a DSV time of 654ms. Nevertheless, owing to the inappropriate nonce selection, the traditional algorithms had high time complexity. Therefore, due to the usage of SE, the proposed method had superior reliability.

4.7. Comparative Validation of the Proposed Work

Here, to illustrate the model's efficacy, the performance of the proposed framework is analogized with numerous related studies.

Table 5. Comparative analysis of the proposed work

Authors' name	Techniques	Accuracy (%)	Precision (%)	F-measure (%)
Proposed work	TDLAHTRSTM	98.9974	98.9867	98.9974
(Singh et al., 2022)	Support Vector Machine (SVM)	85.65	-	85.00
(Baabdullah et al., 2024)	Convolutional Neural Network (CNN), RF, and LSTM	97.00	97.00	97.00
(Karthika & Senthilselvi, 2023)	Dilated CNN	97.39	36.82	37.80
(Berhane et al., 2023)	CNN and SVM	91.08	90.50	90.41
(Benchaji et al., 2021)	LSTM	97.48	97.69	-

In Table 5, the comparative investigation of the proposed work and related studies concerning CCFD is displayed. An accuracy of 98.9974%, f-measure of 98.9974%, and precision of 98.9867% were achieved by the proposed TDLAHTRSTM. The efficiency of the proposed TDLAHTRSTM was improved due to the utilization of ASHT and RCR. Moreover, for classifying the credit card transactions into normal and fraudulent, the conventional studies used techniques like SVM, CNN, RF, LSTM, and dilated CNN. But, the existing dilated CNN (Karthika & Senthilselvi, 2023) attained an accuracy of 97.39%, f-measure of 37.80%, and precision of 36.82%. The prevailing studies had poor performance owing to the improper activation function. Furthermore, the incorporation of TL assisted in upgrading the generalizability of the model. When contrasted with conventional studies, the proposed work was more efficient.

5. Conclusion

In this article, an intelligent ABCBNLS-based context-aware CCFD with UST analysis and CSD detection utilizing HFMM was proposed. The behaviour risk level was efficiently investigated by the proposed ABCBNLS, thus recognizing the subtle fraudulent behaviours. Hence, the proposed ABCBNLS took a rule generation time of 872ms, depicting low time complexity. To group the transaction patterns into usual and unusual, a novel DPBSGCAN was used, thereby facilitating precise anomaly detection. Here, the proposed HFMM efficiently investigated the UST and detected the CSD. Therefore, the reliability and context-awareness of the proposed CCFD were improved. Also, to upgrade the efficiency of the model, processes like user authentication, correlation analysis, and deep explainer were done. Lastly, the credit card transactions were significantly classified into fraud and normal by the proposed TDLAHTRSTM with an accuracy of 98.9974%. The proposed HFMM attained perplexity of 1.423 and

KL-divergence of 0.284, thereby depicting impressive performance. Moreover, the proposed D2SA took a digital signature creation time of 663ms, displaying financial security. Hence, the proposed work had higher reliability in CCFD, thus increasing financial security.

Future Scope

For enhancing the significance of the dynamic credit card usage environments, future work will concentrate on rendering efficient and potential anomaly mitigation measures utilizing advanced context-aware techniques.

References

Dataset

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

[1] Alarfaj, F. K., Malik, I., Khan, H. U., Almusallam, N., Ramzan, M., & Ahmed, M. (2022). Credit Card Fraud Detection Using State-of-the-Art Machine Learning and Deep Learning Algorithms. *IEEE Access*, *10*, 39700–39715. https://doi.org/10.1109/ACCESS.2022.3166891

[2] Baabdullah, T., Alzahrani, A., Rawat, D. B., & Liu, C. (2024). Efficiency of Federated Learning and Blockchain in Preserving Privacy and Enhancing the Performance of Credit Card Fraud Detection (CCFD) Systems. *Future Internet*, *16*(6), 1–22. https://doi.org/10.3390/fi16060196

[3] 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

[4] Belle, R. Van, Baesens, B., & Weerdt, J. De. (2023). CATCHM: A novel network-based credit card fraud detection method using node representation learning. *Decision Support Systems*, *164*, 1–34. https://doi.org/10.1016/j.dss.2022.113866

[5] Berhane, T., Melese, T., Walelign, A., & Mohammed, A. (2023). A Hybrid Convolutional Neural Network and Support Vector Machine-Based Credit Card Fraud Detection Model. *Mathematical Problems in Engineering*, *2023*, 1–10. https://doi.org/10.1155/2023/8134627

[6] Charizanos, G., Demirhan, H., & İçen, D. (2024). An online fuzzy fraud detection framework for credit card transactions. *Expert Systems with Applications*, *252*, 1–15. https://doi.org/10.1016/j.eswa.2024.124127

[7] Chung, J., & Lee, K. (2023). Credit Card Fraud Detection : An Improved Strategy for High. *Sensors, MDPI*, *23*, 1–16. https://doi.org/10.3390/s23187788

[8] Dastidar, K. G., Jurgovsky, J., Siblini, W., & Granitzer, M. (2022). NAG: neural feature aggregation framework for credit card fraud detection. *Knowledge and Information Systems*, *64*, 831–858. https://doi.org/10.1007/s10115-022-01653-0

[9] Ding, Y., Kang, W., Feng, J., Peng, B., & Yang, A. (2023). Credit Card Fraud Detection Based on Improved Variational Autoencoder Generative Adversarial Network. *IEEE Access*, *11*, 83680–83691. https://doi.org/10.1109/ACCESS.2023.3302339

[10] Esenogho, E., Mienye, I. D., Swart, T. G., Aruleba, K., & Obaido, G. (2022). A Neural Network Ensemble with Feature Engineering for Improved Credit Card Fraud Detection. *IEEE Access*, *10*, 16400–16407. https://doi.org/10.1109/ACCESS.2022.3148298

[11] Ghaleb, F. A., Saeed, F., Al-Sarem, M., Qasem, S. N., & Al-Hadhrami, T. (2023). Ensemble Synthesized Minority Oversampling-Based Generative Adversarial Networks and Random Forest Algorithm for Credit Card Fraud Detection. *IEEE Access*, *11*, 89694–89710. https://doi.org/10.1109/ACCESS.2023.3306621

[12] Habibpour, M., Gharoun, H., Mehdipour, M., Tajally, A. R., Asgharnezhad, H., Shamsi, A., Khosravi, A., & Nahavandi, S. (2023). Uncertainty-aware credit card fraud detection using deep learning. *Engineering Applications of Artificial Intelligence*, *123*, 1–10. https://doi.org/10.1016/j.engappai.2023.106248

[13] Hafez, I. Y., Hafez, A. Y., Saleh, A., Abd El-Mageed, A. A., & Abohany, A. A. (2025). A systematic review of AI-enhanced techniques in credit card fraud detection. *Journal of Big Data*, *12*, 1–35. https://doi.org/10.1186/s40537-024-01048-8

[14] 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*, 1–17. https://doi.org/10.1186/s40537-022-00573-8

[15] Karthika, J., & Senthilselvi, A. (2023). Smart credit card fraud detection system based on dilated convolutional neural network with sampling technique. *Multimedia Tools and Applications*,

82(20), 1–19. https://doi.org/10.1007/s11042-023-15730-1

[16] 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*, 1–27. https://doi.org/10.1109/ICKECS61492.2024.10616809

[17] Lebichot, B., Verhelst, T., BORGNE, Y.-A. LE, He-Guelton, L., Oble, F., & Bontempi, G. (2021). Transfer Learning Strategies for Credit Card Fraud Detection. *IEEE Access*, *9*, 114754–114766. https://doi.org/10.1109/ACCESS.2021.3104472

[18] Mienye, I. D., & Jere, N. (2024). Deep Learning for Credit Card Fraud Detection: A Review of Algorithms, Challenges, and Solutions. *IEEE Access*, *12*, 96893–96910. https://doi.org/10.1109/ACCESS.2024.3426955

[19] Mienye, I. D., & Sun, Y. (2023). A Deep Learning Ensemble With Data Resampling for Credit Card Fraud Detection. *IEEE Access*, *11*, 30628–30638. https://doi.org/10.1109/ACCESS.2023.3262020

[20] Mniai, A., Tarik, M., & Jebari, K. (2023). A Novel Framework for Credit Card Fraud Detection. *IEEE Access*, *11*, 112776–112786. https://doi.org/10.1109/ACCESS.2023.3323842

[21] Peter, A., Manoj, K., & Kumar, P. (2023). Blockchain and Machine Learning Approaches for Credit Card Fraud Detection. *Proceedings - 5th International Conference on Smart Systems and Inventive Technology, ICSSIT 2023*, 1034–1041. https://doi.org/10.1109/ICSSIT55814.2023.10060999

[22] Raval, J., Bhattacharya, P., Jadav, N. K., Tanwar, S., Sharma, G., Bokoro, P. N., Elmorsy, M., Tolba, A., & Raboaca, M. S. (2023). RaKShA: A Trusted Explainable LSTM Model to Classify Fraud Patterns on Credit Card Transactions. *Mathematics*, *11*, 1–27. https://doi.org/10.3390/math11081901

[23] Salam, M. A., 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

[24] Salekshahrezaee, Z., Leevy, J. L., & Khoshgoftaar, T. M. (2023). The effect of feature extraction and data sampling on credit card fraud detection. *Journal of Big Data*, *10*, 1–17. https://doi.org/10.1186/s40537-023-00684-w

[25] Singh, A., Jain, A., & Biable, S. E. (2022). Financial Fraud Detection Approach Based on Firefly Optimization Algorithm and Support Vector Machine. *Applied Computational Intelligence and Soft Computing*, *2022*, 1–10. https://doi.org/10.1155/2022/1468015

[26] Sulaiman, R. Bin, Schetinin, V., & Sant, P. (2022). Review of Machine Learning Approach on Credit Card Fraud Detection. *Human-Centric Intelligent Systems*, *2*, 55–68. https://doi.org/10.1007/s44230-022-00004-0

[27] Tien, H. T., Tran-Trung, K., & Hoang, V. T. (2024). Blockchain-Data Mining Fusion for Financial Anomaly Detection: A Brief Review. *Procedia Computer Science*, *235*, 478–483. https://doi.org/10.1016/j.procs.2024.04.047

[28] Wang, J., Liu, J., Zheng, W., & Ge, Y. (2025). Temporal Heterogeneous Graph Contrastive Learning for Fraud Detection in Credit Card Transactions. *IEEE Access*, *13*, 145754–145771. https://doi.org/10.1109/ACCESS.2025.3599787

[29] Xie, Y., Liu, G., Yan, C., Jiang, C., & Zhou, M. C. (2023). Time-Aware Attention-Based Gated Network for Credit Card Fraud Detection by Extracting Transactional Behaviors. *IEEE Transactions on Computational Social Systems*, *10*(3), 1–13. https://doi.org/10.1109/TCSS.2022.3158318

[30] Yu, C., Xu, Y., Cao, J., Zhang, Y., Jin, Y., & Zhu, M. (2024). Credit Card Fraud Detection Using Advanced Transformer Model. *ArXiv*, 1–8. https://doi.org/10.1109/MetaCom62920.2024.00064