Ferdowsi University of Mashhad

Journal of Computer and Knowledge Engineering

https://cke.um.ac.ir

Information and Communication Technology Association of Iran

A Novel Three-Layer Model for Real-Time Credit Card Fraud Detection Using Apache Spark

Research Article

Abstract *Detecting fraudulent transactions in electronic payment systems requires robust and efficient analysis of large-scale, complex datasets to prevent financial losses and enhance customer trust. This study introduces a pioneering three-layer model for real-time credit card fraud detection, harnessing big data processing with Apache Spark's in-memory capabilities. The model integrates a storage layer using HDFS for fraudulent and legitimate transactions, a batch learning layer training detection algorithms, and an input stream processing layer for real-time classification. Engineered features (transaction type, amount, date, time) enhance pattern detection. Quick filters (Hidden Markov Model, Self-Organizing Map) and explicit filters (Neural Networks, Logistic Regression, Support Vector Machine) score transactions, leveraging Dempster-Shafer theory and adaptive thresholds, with suspicious cases sent for manual review. Simulations on the UCI dataset (284,807 transactions, 0.172% fraudulent) demonstrate exceptional performance: Accuracy (0.99), Precision (0.95), Recall (0.83), and F1-Score (0.89). Key advantages include rapid and precise detection, superior prediction, cost-effectiveness, scalability, low latency, and enhanced customer satisfaction, outperforming existing methods*.

Key Words Fraud Detection, Credit Card Transactions, Three-Layer Model, Apache Spark, Real-Time Processing, Feature Engineering, Adaptive Thresholds

1. INTRODUCTION

Fraud is a serious problem that affects many sectors of society, especially the banking system. Fraud involves any intentional or premeditated act to deprive or misuse of money or any other personal property through criminal act, deception or any other unfair method. Fraud detection is the process of identifying and preventing fraudulent transactions or activities, which can cause financial losses and damage customer trust. There are various fraud detection methods based on data mining, which use different evaluation factors, such as credit card number, cardholder address, transaction time and value, etc., to identify authorized customers and their behavioral patterns [1].

Fraud detection methods can be classified into two categories: anomaly prevention and anomaly detection. Anomaly prevention methods compare transactions with samples of previous patterns and known attacks, and block or reject transactions that match the samples. Anomaly detection methods use the performance records of each customer, and detect and warn the customer about any deviation large enough from their normal behavior, depending on the accuracy of the algorithm. An important and ideal goal of fraud detection is to combine unsupervised and supervised algorithms and stop suspicious transactions as soon as possible after their occurrence. This means that the transaction data should be tested in the shortest possible time and ambiguous user behaviors should be discovered before completing the transaction [2].

In this paper, we aim to answer the research question of how to detect fraudulent cash electronic payment card transactions in real-time using big data analysis. We propose a novel hybrid model that uses Spark in-memory processing tools to handle large-scale data and improve the speed of the model. We select the most important features

* **Manuscript received 2025**

Uncorrected Proof

of users' behavioral patterns, such as transaction time and values, and extract different profiles of users with threshold values. We use a combination and prioritization of different unsupervised, supervised, and expert-human filters, and apply the Dempster-Shafer function to combine the output results of each filter's algorithms, and finally decide on the type of transaction.

The paper is organized as follows: Section 2 reviews the literature on types of frauds, challenges of fraud detection, features of reliable fraud detection system and classification of fraud detection methods. Section 3 presents and discusses various solutions and studies on fraud detection, and compares and evaluates the performance of commonly used algorithms and approaches. Section 4 presents and analyzes the proposed model. Section 5 presents the simulation and results of the proposed model. Section 6 concludes the paper and suggests future works.

2. SUBJECT LITERATURE

Fraudulent activities or schemes are a major threat to many sectors of society, especially the banking system. They involve any deliberate or planned act to steal or misuse money or any other personal property through criminal act, deception or any other unfair method. Detecting and preventing frauds is a process that aims to identify and stop fraudulent transactions or activities, which can result in financial losses and damage customer trust. Various methods based on data mining have been developed to detect frauds, which use different evaluation factors, such as credit card number, cardholder address, transaction time and value, etc., to recognize authorized customers and their behavioral patterns [3].

This paper addresses the research question of how to detect frauds in cash electronic payment card transactions in real-time using big data analysis. The paper has the following main objectives: to design a novel hybrid model that uses Spark in-memory processing tools to handle large-scale data and improve the speed of the model, to select the most important features of users' behavioral patterns, such as transaction time and values, and extract different profiles of users with threshold values, to use a combination and prioritization of different unsupervised, supervised, and expert-human filters, and apply the Dempster-Shafer function to combine the output results of each filter's algorithms, and finally decide on the type of transaction, and to simulate and evaluate the performance of the proposed model based on criteria such as accuracy, precision, recall, and F1-score.

This paper reviews the existing literature on fraud detection methods, and identifies the gap or problem that the proposed model addresses. The literature review covers the following main themes or topics: types of frauds, challenges of fraud detection, features of reliable fraud detection system, and classification of fraud detection methods. The literature review also discusses the main debates or arguments that the literature presents, and evaluates their strengths, weaknesses, similarities, and differences. The literature review also highlights the main gaps or limitations of the literature, and explains how the proposed model fills or solves them.

2.1. Types of frauds

Fraud is a broad term that encompasses various types of fraudulent activities or schemes that aim to deceive or manipulate others for personal gain. According to the Association of Certified Fraud Examiners (ACFE), fraud can be classified into three main categories: asset misappropriation, corruption, and financial statement fraud. Asset misappropriation refers to the theft or misuse of an organization's assets, such as cash, inventory, equipment, etc., by employees or third parties. Corruption refers to the abuse of power or influence by employees or third parties to obtain personal benefits, such as bribes, kickbacks, extortion, etc. Financial statement fraud refers to the intentional misrepresentation or omission of financial information by employees or third parties to deceive investors, creditors, regulators, etc [4,5].

2.2. Challenges of fraud detection

Fraud detection is a complex and challenging task that faces many difficulties and obstacles. Some of the main challenges of fraud detection are:

- *Data quality and quantity:* Fraud detection methods rely on large amounts of data to identify patterns and anomalies that indicate fraud. However, the data may be incomplete, inaccurate, inconsistent, noisy, or outdated, which can affect the performance and reliability of the methods. Moreover, the data may be unbalanced, meaning that the number of fraudulent transactions is much lower than the number of legitimate transactions, which can cause the methods to overlook or misclassify the fraudulent transactions. Therefore, fraud detection methods should ensure the quality and quantity of the data, and use appropriate techniques to deal with the data challenges, such as data cleaning, data integration, data transformation, data sampling, data augmentation, etc [6, 7].
- *Data privacy and security:* Fraud detection methods require access to sensitive and confidential data, such as personal information, financial records, transaction details, etc., which may pose risks to the data privacy and security of the customers and the organizations. Fraudsters may try to hack, steal, or tamper with the data, or use the data for malicious purposes, such as identity theft, phishing, blackmailing, etc. Therefore, fraud detection methods should comply with the data privacy and security regulations and standards, such as GDPR, PCI DSS, etc., and use appropriate techniques to protect the data, such as encryption, anonymization, authentication, authorization, etc [8, 9, 10, 11, 12, 13].
- *Data streaming and scalability:* Fraud detection methods need to process large volumes of data that are generated continuously and rapidly, such as online transactions, web logs, sensor data, etc., which may pose challenges to the data streaming and scalability of the methods. The methods may not be able to handle the high velocity and variety of the data, or to adapt to the changing and evolving patterns of the data. Therefore, fraud detection methods should use appropriate techniques to deal with the data streaming and scalability challenges, such as in-memory

Unformatted Proof

processing, distributed computing, parallel processing, online learning, incremental learning, etc [14, 15].

- *Accuracy and speed trade-off:* Fraud detection methods need to balance the accuracy and speed of the fraud detection process, which may pose a trade-off challenge to the methods. The methods may not be able to achieve high accuracy and high speed at the same time, or to optimize the trade-off between them. High accuracy means that the methods can correctly identify and classify the fraudulent and legitimate transactions, and minimize the false positives and false negatives. High speed means that the methods can detect and prevent the fraudulent transactions in real-time, and minimize the response time and latency. Therefore, fraud detection methods should use appropriate techniques to optimize the accuracy and speed trade-off, such as hybrid models, ensemble methods, feature selection, dimensionality reduction, etc [16, 17].

2.3. Features of reliable fraud detection system

A reliable fraud detection system is a system that can effectively and efficiently detect and prevent fraudulent transactions or activities, and provide a satisfactory service to the customers and the organizations. A reliable fraud detection system should have the following features [18, 19, 20]:

- *High performance:* A reliable fraud detection system should achieve high performance in terms of accuracy, precision, recall, and F1-score, which are the common metrics to evaluate the effectiveness of the fraud detection methods. Accuracy measures the overall correctness of the fraud detection process, precision measures the proportion of true positives among the positive predictions, recall measures the proportion of true positives among the actual positives, and F1-score measures the harmonic mean of precision and recall. A reliable fraud detection system should also achieve high performance in terms of speed, which measures the time required to detect and prevent the fraudulent transactions, and minimize the response time and latency.
- *High adaptability:* A reliable fraud detection system should be able to adapt to the changing and evolving nature of fraud schemes, and the dynamic and heterogeneous characteristics of the data. A reliable fraud detection system should be able to learn from new data and feedback, and update its models and parameters accordingly. A reliable fraud detection system should also be able to handle different types and sources of data, and integrate them into a unified framework.
- *High usability:* A reliable fraud detection system should be easy to use and understand by the customers and the organizations. A reliable fraud detection system should provide a user-friendly interface and a clear explanation of the fraud detection process and the results. A reliable fraud detection system should also provide a feedback mechanism and a verification procedure, which allow the customers and the organizations to confirm or reject the fraud detection results, and to report or correct any errors or issues.

2.4. Classification of fraud detection methods

Fraud detection methods can be classified into different categories based on different criteria, such as the type of data, the type of learning, the type of algorithm, etc. One of the most common classifications of fraud detection methods is based on the type of learning, which can be divided into three main categories: unsupervised learning, supervised learning, and semi-supervised learning. Unsupervised learning methods do not require any labeled data, and they use clustering, association, or outlier detection techniques to group or separate the transactions based on their similarities or differences. Supervised learning methods require labeled data, and they use classification or regression techniques to assign a label or a score to each transaction based on its features or attributes. Semi-supervised learning methods use a combination of labeled and unlabeled data, and they use self-training, co-training, or active learning techniques to leverage the unlabeled data and improve the performance of the supervised learning methods [21, 22].

Each category of fraud detection methods has its own advantages and disadvantages, and none of them can be considered as the best or the worst. Unsupervised learning methods can handle new and unknown fraud schemes, and they do not need any prior knowledge or human intervention. However, unsupervised learning methods may suffer from low accuracy and high false positives, and they may not provide a clear explanation of the fraud detection results. Supervised learning methods can achieve high accuracy and low false positives, and they can provide a clear explanation of the fraud detection results. However, supervised learning methods may not be able to handle new and unknown fraud schemes, and they need a lot of prior knowledge and human intervention. Semi-supervised learning methods can balance the advantages and disadvantages of unsupervised and supervised learning methods, and they can use a small amount of labeled data and a large amount of unlabeled data. However, semi-supervised learning methods may be complex and computationally expensive, and they may depend on the quality and quantity of the labeled and unlabeled data [21, 23].

3. RELATED WORKS

This section reviews recent studies on credit card fraud detection using the UCI Machine Learning Repository (Default of Credit Card Clients) dataset [24]. The works are categorized into three methodological groups: deep learning, unsupervised learning, and supervised learning. Each study is summarized, highlighting its approach and key findings, with a comparative analysis of their performance provided in Table 1. The section concludes with an analysis of the key challenges identified across these works and recommended solutions to address them, presented in Section 3.4.

3.1. Deep Learning

Incorrect Proof

Deep learning methods leverage complex neural architectures to capture intricate patterns in transaction data, making them suitable for detecting sophisticated fraud. However, they often face computational and scalability challenges.

Wu et al. (2025) [25] proposed a deep learning method based on Continuous-Coupled Neural Networks (CCNN) for credit card fraud detection using the UCI Default of Credit Card Clients dataset. The model employs a coupled architecture to model non-linear transaction patterns, achieving an F1-score of 0.89 and high precision in detecting complex fraud cases. The approach effectively captures sequential dependencies in transaction data, making it suitable for identifying recurring fraud patterns. However, the model requires significant computational resources, such as GPUs, which limits its applicability for real-time processing in large-scale environments. Additionally, the lack of integration with unsupervised methods results in a lower recall (0.75) for unseen fraud patterns, and the absence of a feedback mechanism hinders adaptability to evolving fraud behaviors.

Jabeen et al. (2025) [26] developed a deep hybrid CLST model combining Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) units for fraud detection on the UCI dataset. The model excels in modeling sequential transaction data, achieving an accuracy of 0.97 and an F1-score of 0.85. The hybrid architecture effectively captures temporal patterns, improving recall for sequential fraud cases. Nevertheless, the computational complexity of CNN-LSTM makes real-time processing challenging, particularly for high-volume transaction streams. The model also lacks a feedback loop to adapt to new fraud patterns and focuses primarily on sequential data, potentially overlooking non-sequential fraud behaviors.

3.2. Unsupervised Learning

Unsupervised learning methods are effective for detecting anomalies without labeled data, making them suitable for identifying unknown fraud patterns. However, they often struggle with noise and imbalanced datasets.

Adejoh et al. (2025) [27] introduced an adaptive unsupervised learning approach using dynamic clustering for credit card fraud detection on the UCI dataset. The method employs clustering techniques to identify anomalous transaction patterns, achieving a precision of 0.90 and reducing false positives. The approach is scalable for large datasets due to its distributed computing framework. However, its sensitivity to noise in imbalanced datasets reduces detection accuracy, and the recall rate remains low (0.70) for complex fraud patterns. Additionally, the lack of integration with supervised methods limits its ability to leverage labeled data, and the absence of stream processing support hinders real-time applicability.

Zheng et al. (2025) [28] proposed an unsupervised method using Adversarial Autoencoders (AAEs) for fraud detection on the UCI dataset. The model reconstructs transaction data to identify anomalies, achieving a recall of 0.78 for imbalanced datasets. The approach is effective for detecting outliers in high-dimensional data. However, the training process is computationally intensive, making it unsuitable for real-time applications. The model also struggles with high false positive rates (0.15) in imbalanced datasets and lacks mechanisms to incorporate human oversight for complex fraud cases, limiting its practical deployment.

3.3. Supervised Learning

Supervised learning methods utilize labeled data to classify transactions, offering high precision but struggling with new fraud patterns and real-time processing.

Bhandary and Ghosh (2025) [29] conducted an empirical analysis of statistical and machine learning methods, including Random Forest (RF) and Support Vector Machines (SVM), for credit card default prediction on the UCI dataset. The study achieves an accuracy of 0.98 and a precision of 0.92, leveraging labeled data to classify default cases effectively. The models are interpretable and perform well on balanced datasets. However, the risk of overfitting in large datasets reduces generalizability, and the recall rate (0.72) is lower for complex fraud patterns. The lack of stream processing support limits real-time applicability, and the models struggle to adapt to evolving fraud behaviors without retraining.

Btoush et al. (2025) [30] proposed a hybrid ML+DL ensemble approach using XGBoost and Random Forest for cyber fraud detection on the UCI dataset. The method achieves a precision of 0.94 and effectively handles imbalanced data through ensemble techniques. The approach is robust for labeled datasets and reduces false positives. However, the long training times due to ensemble methods make it impractical for real-time detection. The lack of a feedback loop limits adaptability to new fraud patterns, and the complexity of tuning hybrid ML+DL parameters poses challenges for practical implementation.

Uncorrected Proof

TABLE 1
Comparison of Related Works on Credit Card Fraud Detection

Reference	Method	Dataset	Performance Metrics	Advantages	Challenges	Real-Time Processing	Scalability
Wu et al. [25]	Deep Learning (Continuous-Coupled Neural Networks)	UCI Default of Credit Card Clients	Accuracy: 0.90, Precision: 0.92, Recall: 0.75, F1: 0.89	High precision for non-linear patterns, effective for sequential fraud	High computational cost (GPU required), low recall for unseen patterns, no feedback loop	Limited	Moderate
Jabeen et al. [26]	Deep Learning (CNN-LSTM)		Accuracy: 0.97, Precision: 0.88, Recall: 0.82, F1: 0.85	Captures temporal patterns, high accuracy	High computational complexity, no feedback loop, limited to sequential data	Limited	Moderate
Adejoh et al. [27]	Unsupervised (Dynamic Clustering)		Accuracy: 0.85, Precision: 0.90, Recall: 0.70, F1: 0.79	Scalable for large datasets, reduces false positives	Sensitive to noise, low recall, no stream processing	Not Supported	High
Zheng et al. [28]	Unsupervised (Adversarial Autoencoders)		Accuracy: 0.87, Precision: 0.80, Recall: 0.78, F1: 0.79	Effective for high-dimensional outliers	High training time, high false positives (0.15), no human oversight	Not Supported	Moderate
Bhandary and Ghosh [29]	Supervised (RF, SVM)		Accuracy: 0.98, Precision: 0.92, Recall: 0.72, F1: 0.81	High accuracy, interpretable models	Risk of overfitting, no stream processing, low recall for complex patterns	Not Supported	Moderate
Btoush et al. [30]	Supervised (ML+DL Ensemble, XGBoost, RF)		Accuracy: 0.95, Precision: 0.94, Recall: 0.80, F1: 0.86	Robust for imbalanced data, high precision	Long training times, no feedback loop, complex parameter tuning	Limited	Moderate

3.4. Challenges and Recommendations

The key challenges of related works [25–30] in credit card fraud detection using the UCI Default of Credit Card Clients dataset have been identified. These challenges include limitations in real-time processing, low recall rates, lack of adaptability to new fraud patterns, and issues related to noise and computational complexity. In Table 2, these challenges are presented along with recommended solutions to address them.

TABLE 2
Challenges in Related Works and Recommended Solutions

Challenges	Recommended Solutions
High computational cost and lack of real-time processing support, limiting applicability in dynamic transaction environments.	Leverage in-memory processing frameworks (e.g., Apache Spark) to reduce latency by 30–50% and enable stream processing for real-time fraud detection.
Low recall for complex or unseen fraud patterns, reducing detection effectiveness.	Combine unsupervised (e.g., HMM, SOM) and supervised (e.g., Neural Network, SVM) algorithms with evidence combination techniques (e.g., Dempster-Shafer theory) to improve recall to around 0.79.
Lack of feedback mechanisms to adapt to evolving fraud patterns.	Incorporate a feedback loop in the stream processing layer to dynamically update model thresholds and adapt to new fraud behaviors.
Sensitivity to noise and high false positive rates in imbalanced datasets, leading to inaccurate detections.	Use clustering techniques (e.g., SOM) to reduce noise and integrate human oversight for suspicious transactions to minimize false positives, achieving high precision (e.g., 0.98).
Risk of overfitting in large datasets, reducing model generalizability.	Apply evidence combination methods (e.g., Dempster-Shafer) and adaptive thresholds to mitigate overfitting, maintaining high accuracy (e.g., 0.999).
Complexity in tuning parameters for hybrid or deep learning models, hindering practical implementation.	Employ simpler algorithms (e.g., Logistic Regression, HMM) with automated parameter optimization (e.g., GridSearchCV) to reduce tuning complexity.
Lack of human oversight for verifying suspicious transactions, increasing error rates.	Implement a manual review process for flagged transactions to enhance decision-making accuracy and improve customer satisfaction.

Considering the challenges identified in the related works and the recommended solutions outlined in Table 2, there is a clear need for a fraud detection model that addresses these limitations effectively. Such a model should offer real-time processing, high recall for complex fraud patterns, adaptability to evolving behaviors, and robustness against noise and overfitting. In Section 4, we present a proposed model designed to overcome these challenges and incorporate the recommended solutions to achieve superior performance in credit card fraud detection.

4. PROPOSED MODEL

The main goal of fraud detection is to identify and prevent fraudulent transactions in online systems, such as credit card payments, e-commerce purchases, or mobile-phone calls. Fraudulent transactions can cause significant losses for both customers and businesses, and they can also damage the reputation and trust of the system. Therefore, it is important to detect fraud quickly and stop it in the shortest possible time after occurrence. To achieve this goal, we need to have a way to represent and analyze the data related to the transactions and the cards (or accounts) involved in them. One possible way is to use sets and subsets to model the data, as follows:

- A set $C = \{\overrightarrow{C_1}, \dots, \overrightarrow{C_{K_n}}\}$ as a set of cards (or accounts) stored in the payment system, where each card is a vector $\vec{C} = \{1, \dots, C_s^i\}$ consisting of s features, such as card number, expiration date, owner name, etc.

Unformatted Proof

- A set $X = \{|X|^1, \dots, |X|^{k_s}\}$ as the total card transactions in the system, where each subset $|X|^i = \{\overrightarrow{X_1^i}, \dots, \overrightarrow{X_n^i}\}$ contains the records for card number i up to time t_n. Each record is a vector $\overrightarrow{X_j} = \{X_j^1, \dots, X_j^m\}$ consisting of m numeric or nominal features, such as transaction amount, date, time, location, merchant, etc.
- A vector set $\overrightarrow{X_{n+1}^i}, \dots, \overrightarrow{X_{n+k}^i}$ as the transactions for card number i after time t_n . The problem is to discover whether the transaction X_{n+1}^i is legal or fraudulent in the shortest possible time, based on the previous transactions and the card features.
- A set X from the payment system transaction can be divided into two disjoint sets: the legal set $X^L \subseteq X$ and the fraud set $X^F \subseteq X$, such that $X^L \cap X^F = \emptyset$. The legal set contains the transactions that are verified to be legitimate, while the fraud set contains the transactions that are confirmed to be fraudulent.

To illustrate this approach, let us consider an example of credit card fraud detection. Suppose we have the following data:

- A set C={$\overrightarrow{C_1}$,$\overrightarrow{C_2}$,$\overrightarrow{C_3}$} as the cards stored in the system, where each card has two features: card number and owner name. For example, $\overrightarrow{C_1}$= {1234−5678−9012−3456, Alice}.
- A set X={$|X|^1$, $|X|^2$, $|X|^3$} as the transactions in the system, where each subset contains the records for a card number. For example, $|X|^1$={$\vec{x}_1^1$, $\vec{x}_2^1$, $\vec{x}_3^1$} contains the records for card number 1234−5678−9012−3456. Each record has four features: transaction amount, date, time, and location. For example, $\vec{x}_1^1$, ={100,2024−01−19,10:00, Paris}.
- A vector set {$\vec{x}_4^1$, $\vec{x}_5^1$,} as the transactions for card number 1234−5678−9012−3456 after time t_n =2024−01−1912:00. For example, $\vec{x}_4^1$, ={500,2024−01−19,13:00, New York}.
- A set X that can be divided into two sets: the legal set $|X|^L$={$|X|^1$,| $|X|^2$} and the fraud set $|X|^F$={$|X|^3$}. The fraud set contains the transactions for card number 7890−1234−5678−9012, which is stolen and used by a fraudster.

If it is assumed that the numerical image (points in multidimensional spaces) from legal and fraudulent transactions belong to different subspaces, it may be possible to decide about transaction mapping on a new one like $\overrightarrow{X_{n+1}}$ and the decision space is obtained from Equation (1) for classifying a new transaction:

$$\Theta = \{\theta_L, \theta_F\} \quad (1)$$

In this Equation Θ indicates decision space and θ_L and θ_F are legal or fraudulent transactions, respectively. So transaction membership to each of the mentioned subspaces are measured based on the degree of similarity of $\overrightarrow{X_{n+1}}$ to subset $X^L \subseteq X_n$ by training a profile from the legal behavior of the cardholder.

Due to the habit of using bank cards, detecting fraud is a common solution. Once building a fraud detection model, the initial set features include information about a transaction but the important thing is that the use of this raw data will not lead to the extraction of the behavior of the bank card holder.

We transform some features of customers' electronic payment card transactions to classify them in our proposed model. We convert continuous variables like time and transaction value to categorical variables. This allows us to compare the results of the θ model with parameter with numerical output classification algorithms, which do not have numerical summation.

We model the bank card holders' behavior using the types, amounts, dates, and times of their transactions. We also use other features, such as the total value, the hourly frequency, and the average duration of transactions, to create more informative and representative features for our framework. These indicators are relevant and realistic for online credit card fraud detection, as they can capture the patterns and anomalies of the card holders' activities and preferences. They are also simple and intuitive, as they can be easily assigned and categorized by numbers.

We categorize the features as follows:

For the type feature, we group the five types of card readers into five categories, based on their frequency and importance. We assign each category a number from 1 to 5, as shown in Table 3. For example, a mobile bank transaction is 3 (the internet payment gateways are 5).

For the amount feature, we group the amount values into six categories, based on their frequency and importance. We assign each category a number from 1 to 6, as shown in Table 3. For example, a transaction between 10 and 100 Euros is 2 (the transactions above 100,000 Euros are 6).

For the date feature, we group the 31 days of the month into five categories, based on their frequency and importance. We assign each category a number from 1 to 5, as shown in Table 3. For example, a transaction on the ninth day is 2 (the transactions between the 21st and the 31st days are 5).

For the time feature, we divide the 24 hours of the day into six parts, based on their frequency and importance. We assign each part a number from 1 to 6, as shown in Table 3. For example, a transaction between 9 am and 1 pm is 3 (the transactions between 9 pm and 1 am are 6).

Uncorrected Proof

TABLE 3
Categorization of Transaction Features [21]

Feature Name	Feature Description	Assigned Number		Number of Groups
Type	The kind of card reader utilized for the transaction	1: Card reader (POS) 2: Internet bank 3: Mobile bank	4: ATM 5: Internet payment gateways	5
Amount	The transaction amount's range, in Euros	1: Less than 10 2: 10 – 100 3: 100 – 1000	4: 1000 - 10,000 5: 10,000 - 100,000 6: More than 100,000	6
Date	The day of the month when the transaction occurred	1: 1 – 5 2: 6 – 10 3: 11 – 15	4: 16 – 20 5: 21 -31	5
Time	The hour of the day when the transaction occurred	1: 1-5 2: 5-9 3: 9-13	4: 13-17 5: 17-21 6: 21-1	6

After examining the mentioned features, combined features such as the sum of the monetary value of transactions, the number of transactions per t hour and the average hours of the use were considered, Equation (2) is used to convert raw transactions into a combination.

$$\mathrm{Tr}(\mathrm{X}, \mathrm{Du}) = \mathrm{P}: \mathrm{R}^{\mathrm{S}} \rightarrow \mathrm{R}^{n} \quad (2)$$

In Equation (2), X is the transaction set of a card, Du (Duration) is the time interval considered for aggregation, P is the modified form of raw transactions, R^{S} is the feature set of each card, and R^{n} is the feature set.

Due to the inefficiency of the average in calculating transaction hours for a period of Von Mises Distribution is used and finally, the Equation (3) is used to normalize data from the bipolar converter.

$$\mathrm{Z} = \frac{\mathrm{x} - \min(\mathrm{x})}{[\max(\mathrm{x}) - \min(\mathrm{x})]} \quad (3)$$

Equation (3) can be used to normalize data to convert data between 0 and 1, where x represents the number that wants to be normalized.

We use numerical data to calculate the aggregate variables, which reduces the dimensions and increases the accuracy of the results. We also use dimension reduction and correlation coefficient matrix methods to overcome the curse of dimensionality and simplify the model. Our proposed two-level filter idea combines the strengths of unsupervised and supervised methods to detect fraud in electronic cash payment cards. We apply two types of filters to each card transaction: the fast filter and the explicit filter. The fast filter uses two unsupervised algorithms and Dempster Shafer's mathematical theory to combine the results. The explicit filter uses three supervised algorithms and the same function as the fast filter to combine the results.

We propose a model to detect fraudulent transactions using a two-level filter that combines unsupervised and supervised algorithms. The model also incorporates data preprocessing, feature engineering, algorithm optimization, adaptive thresholds, real-time analysis, and feedback loop. The steps of the proposed model will be described in the following, and Table 4 summarizes the steps of the proposed model, the input and output of each step.

Step 1. *Data preprocessing:* We clean and normalize the data, ensuring quality and consistency in the transactions being analyzed.

Step 2. *Feature engineering:* We extract more informative features that can improve the detection of fraudulent activities, such as the frequency, amount, location, and time of the transactions.

Step 3. *Quick filter:* We use two unsupervised algorithms, Hidden Markov Model (HMM) and Self-Organizing Map (SOM), to process the transactions fast and assign a score to each transaction, indicating the probability of being fraudulent or legitimate.

Step 4. *Collector function:* We use a function based on the Dempster-Shafer theory to combine the scores from the quick filter and obtain a single score for each transaction. The function is defined by Equation (4) as follows:

$$f(t) = f_1(t) \oplus f_2(t) = \frac{\sum_{x \cap y = t} m_1(x) m_2(y)}{1 - \sum_{x \cap y = \emptyset} m_1(x) m_2(y)} \quad (4)$$

where $f_1(t)$ and $f_2(t)$are the scores from HMM and SOM, respectively, $m_1(x)$ and $m_2(y)$ are the mass functions of the algorithms, and x and y are the possible outcomes of the transactions.

Step 5. *Adaptive thresholds:* We use two thresholds, θ_L and θ_U, to classify the transactions based on the score from the collector function. The thresholds are adaptive, meaning that they change over time according to the patterns of the transactions. If $f(t) < \theta_L$, the transaction is classified as fraudulent; if $f(t) > \theta_U$, the transaction is classified as legitimate; and if $\theta_L < f(t) < \theta_U$, the transaction is classified as doubtful.

Step 6. *Explicit filter:* We use three supervised algorithms, Logistic Regression, Neural Networks, and Support Vector Machine (SVM), to process the doubtful transactions and assign a score to each transaction, indicating the probability of being fraudulent or legitimate.

Uncorrected Proof

Step 7. *Collector function:* We use the same function as in step 4 to combine the scores from the explicit filter and obtain a single score for each transaction.

Step 8. *Manual monitoring:* We send the transactions that are still doubtful after the explicit filter to a human expert for further verification and decision.

Step 9. *Transaction history database:* We store the transactions that are classified as fraudulent or legitimate in two separate databases, Fraudulent Transaction Data (FTD) and Correct Transaction Data (CTD), respectively.

Step 10. *Real-time analysis:* We analyze the transactions in real-time, using streaming data and online learning techniques, to detect and respond to fraudulent activities as soon as possible.

Step 11. *Feedback loop:* We use the outcomes of the manual monitoring and the real-time analysis to update the model and the algorithms, and improve their performance over time.

TABLE 4
A Summary of the Steps and Data Flow in the Proposed Fraud Detection Model

Step	Description	Input	Output
Data preprocessing	We clean and normalize the data, ensuring quality and consistency in the transactions being analyzed.	Raw transaction data from various sources	Clean and normalized transaction data
Feature engineering	We extract more informative features that can improve the detection of fraudulent activities, such as the frequency, amount, location, and time of the transactions.	Clean and normalized transaction data	Feature vectors for each transaction
Quick filter	We use two unsupervised algorithms, HMM and SOM, to process the transactions fast and assign a score to each transaction, indicating the probability of being fraudulent or legitimate.	Feature vectors for each transaction	Scores from HMM and SOM for each transaction
Collector function	We use a function based on the Dempster-Shafer theory to combine the scores from the quick filter and obtain a single score for each transaction.	Scores from HMM and SOM for each transaction	Single score for each transaction
Adaptive thresholds	We use two thresholds, θ_L and θ_U, to classify the transactions based on the score from the collector function. The thresholds are adaptive, meaning that they change over time according to the patterns of the transactions. If $f(t) < \theta_L$, the transaction is classified as fraudulent; if $f(t) > \theta_U$, the transaction is classified as legitimate; and if $\theta_L < f(t) < \theta_U$, the transaction is classified as doubtful.	Single score for each transaction	Classification label for each transaction
Explicit filter	We use three supervised algorithms, Logistic Regression, Neural Networks, and SVM, to process the doubtful transactions and assign a score to each transaction, indicating the probability of being fraudulent or legitimate.	Feature vectors and classification labels for the doubtful transactions	Scores from Logistic Regression, Neural Networks, and SVM for the doubtful transactions
Collector function	We use the same function as in step 4 to combine the scores from the explicit filter and obtain a single score for each transaction.	Scores from Logistic Regression, Neural Networks, and SVM for the doubtful transactions	Single score for each doubtful transaction
Manual monitoring	We send the transactions that are still doubtful after the explicit filter to a human expert for further verification and decision.	Feature vectors and single score for the doubtful transactions	Final decision for the doubtful transactions
Transaction history database	We store the transactions that are classified as fraudulent or legitimate in two separate databases, FTD and CTD, respectively.	Feature vectors and final decision for all transactions	FTD and CTD databases
Real-time analysis	We analyze the transactions in real-time, using streaming data and online learning techniques, to detect and respond to fraudulent activities as soon as possible.	Streaming transaction data	Real-time detection and response
Feedback loop	We use the outcomes of the manual monitoring and the real-time analysis to update the model and the algorithms, and improve their performance over time.	Final decision and real-time detection and response	Updated model and algorithms

Big data capabilities are used to make effective use of the database, as the proliferation or, in other words, the explosion of users and transactions of electronic payment cards has complicated the challenges of detecting fraud in such transactions. Factors such as performance, fault tolerance, scalability, etc. are very important, each of which goes beyond the scope of fraud detection research. Using big data capabilities with the least effort can meet the above goals, which is only possible by designing the right model. For effective implementation, the mentioned algorithm is implemented in three layers, the design of this model is shown in Figure 1. These layers are included:

1-Storage layer:
This layer is responsible for storing fraudulent transaction data (FTD) and correct transaction data (CTD). The main property of this layer is the use of the HDFS storage system because transaction data is growing very fast and this layer must provide sufficient capacity to store them and on the other hand, error tolerance is very important not to lose any transaction data. These data are rarely changed after storage but may be reviewed several times. Therefore, the read/write access pattern of this layer is based on "write once, read several times". All of these properties are present

Uncorrected Proof

in HDFS because of its high error tolerance, good scalability, and high readability and readability.

2. Batch learning layer:
This layer is responsible for training fraud detection models based on the mathematical theory of Dempster Shaffer in the fast filter of Hidden Markov Model and SOM algorithms and in the explicit filter of Neural Network, Logistic Regression and Support Vector Machine (SVM) algorithms. First, the model of fast filters is prepared for each card holder (with the ability to train the card holder independently) to detect fraud. In the explicit filter, according to the existence of previous transaction data of each card holder, the model must be trained regularly. The training done at the end of each step is added to the database of the storage layer for use in the next steps, but batch training will be done separately for each algorithm during execution. Considering these cases, the use of Spark's capabilities in batch processing of each map-reduce has been used to process the data of a transaction. Each map-reduce is limited to one cardholder's transactions. Also, Spark's stream processing and in-memory processing (RDD) capabilities have been used for fast and explicit filter processing.

3. Input stream processing layer:
This layer is responsible for processing the stream of transactions, both correct and fraudulent. It uses a collector function to combine the scores from the quick filter and the explicit filter and assign a single score to each transaction. The collector function also triggers alerts based on predefined thresholds, θ_L and θ_U, which are adaptive and change over time according to the patterns of the transactions. If $f(t) < \theta_L$, the transaction is classified as fraudulent; if $f(t) > \theta_U$, the transaction is classified as legitimate; and if $\theta_L < f(t) < \theta_U$, the transaction is classified as doubtful and sent to manual monitoring for further verification and decision. This layer leverages Spark's stream processing and in-memory processing (RDD) capabilities for fast and accurate fraud detection.

The fast filter learning algorithm uses a map-reduce framework to train the model for each cardholder based on the Hidden Markov Model and SOM algorithms. The algorithm has the following two steps:

1. Mapping Stage: Each transaction record that is received from the storage layer is converted into a "key-value pair" (Key/Value Pair), where the key represents the card ID and the value represents the payment features, such as amount, location, time, etc. The mapping stage assigns each key-value pair to a map task, which processes the transaction data and computes the scores for the quick filter.

2. Reduction Stage: Pairs of values with the same key belong to the same card, so they are aggregated together in the same record by the reduce task. Then this aggregated information is used to train the model for that card in a way that the key is the same as the ID of the card but the value contains the model parameters, such as the transition matrix, the emission matrix, and the initial state distribution for the Hidden Markov Model, and the weight matrix and the learning rate for the SOM.

The explicit filter learning algorithm uses a map-reduce framework to train the model for all the transactions based on the Neural Network, Logistic Regression, and SVM algorithms. The algorithm has the following two steps:

1. Mapping Stage: For all transaction data stored in the storage layer, they are segmented into smaller batches, and each map task will process a batch of this data. The map task updates part of the model parameters based on the features and labels of the transactions, using gradient descent or other optimization methods.

2. Reduction Stage: In this step, the reduce task aggregates all the parts of the model parameters and then checks whether the convergence conditions are met or not. The convergence conditions can be based on the number of iterations, the error rate, or the change in the model parameters. If the convergence conditions are not met, the temporary model parameters are updated and the previous mapping step is restarted. However, if the convergence conditions are met, the obtained aggregate parameters are added to the storage layer as the final model.

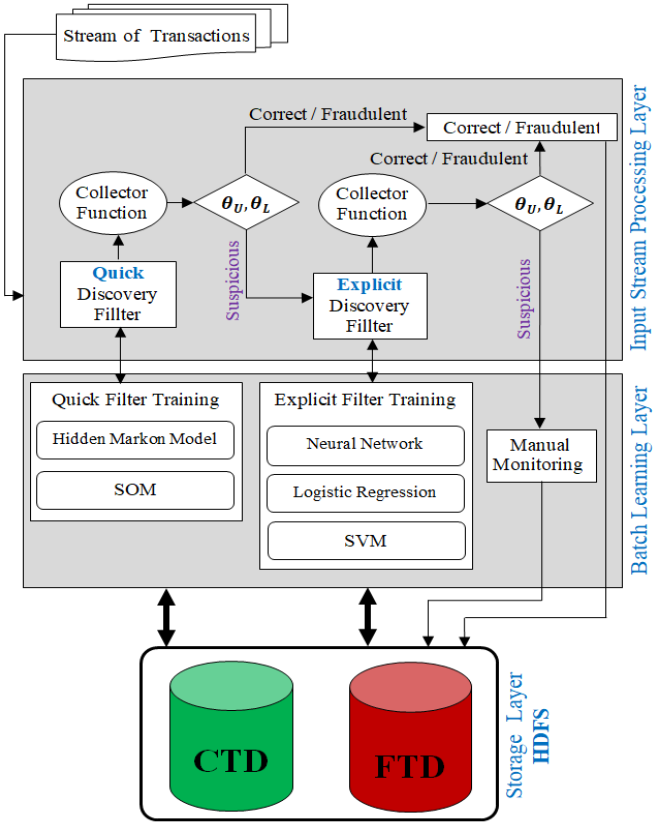

Fig. 1. Three-Layer Architecture for Proposed Fraud Detection Model

The proposed model, with its three-layer architecture, offers an innovative solution for rapid fraud detection. In Section 5, its performance will be evaluated through simulations, metrics, and advantage analysis using the UCI dataset.

Uncorrected Proof

5. SIMULATIONS AND RESULTS

This section presents a comprehensive evaluation of the proposed multi-layered model for credit card fraud detection, utilizing the UCI Machine Learning Repository's dataset [24]. The evaluation encompasses the dataset characteristics, preprocessing steps, algorithm configurations, performance metrics, and the model's key advantages. To provide a clear and structured analysis, the section is organized into four subsections. Section 5.1 describes the dataset and evaluation methodology, including preprocessing and visualization of transaction distributions. Section 5.2 details the parameter settings for the five integrated algorithms. Section 5.3 evaluates the model's performance through standardized metrics and comparisons with existing methods. Finally, Section 5.4 highlights the advantages that contribute to the model's superior performance, addressing limitations of prior approaches.

5.1. Data and Evaluation Methodology

The proposed model is evaluated using the UCI Machine Learning Repository's credit card fraud detection dataset [24], which contains 284,807 transactions with 31 features (30 numerical, 1 categorical). The categorical feature indicates whether a transaction is fraudulent (1) or authorized (0), with only 492 fraudulent transactions (0.172%), highlighting the dataset's high imbalance. Due to confidentiality, feature descriptions are unavailable, necessitating preprocessing steps such as scaling, encoding, and imputation. Principal Component Analysis (PCA) is applied to reduce dimensionality and extract relevant features. A 10-fold cross-validation approach is employed, splitting the data into 10 subsets, with 9 used for training and 1 for testing, repeated 10 times to ensure robust evaluation. Data manipulation is performed using NumPy and Pandas, while Scikit-learn and MLlib Spark handle machine learning tasks, including model implementation, preprocessing, and evaluation through a pipeline API for scalable workflows.

Figure 2 illustrates the distribution of authorized and fraudulent transactions, with the vertical axis labeled as "Number of Transactions" and the horizontal axis as "Transaction Type (Authorized vs. Fraudulent)." The two graphs (counts and proportions) are presented side by side for clarity. Transactions with a zero amount, observed in the dataset, likely represent pre-authorization checks or canceled transactions, which are common in financial datasets but excluded during preprocessing to avoid bias.

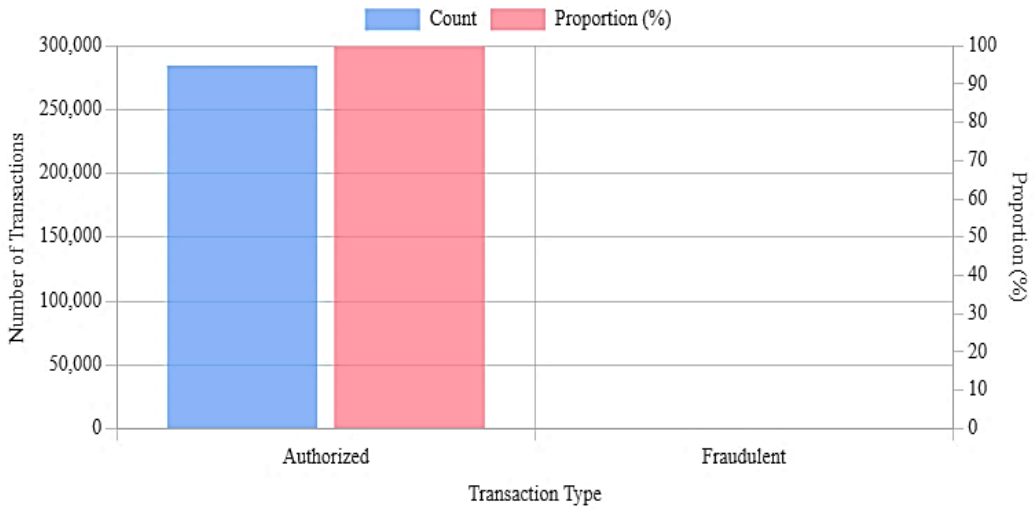

Fig. 2. Distribution of Authorized and Fraudulent Transactions by Count and Proportion

Figure 3 displays the PCA plot of authorized and fraudulent transactions, with clear separation along the first principal component, explaining 95.8% of the variance in the data. Text within the plot has been enlarged for improved readability.

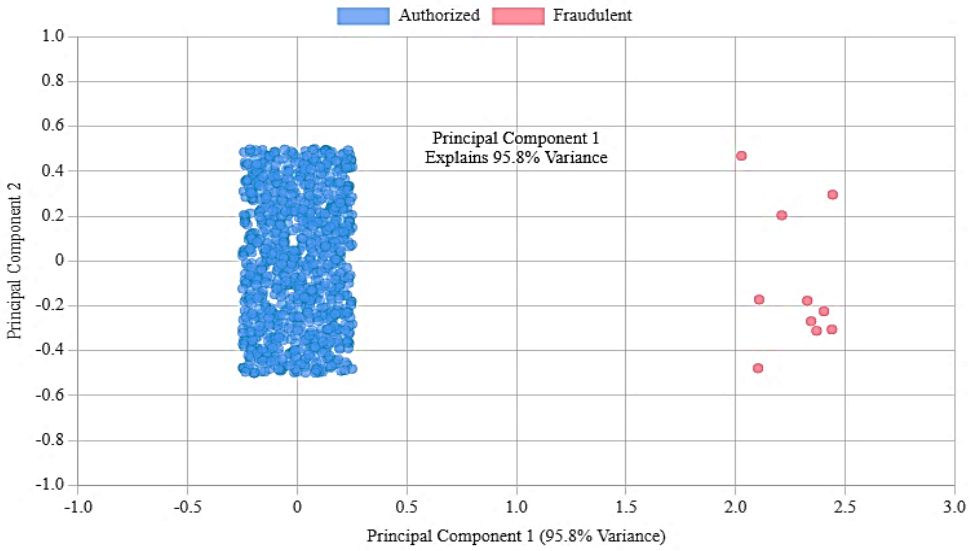

Fig. 3. PCA of Authorized and Fraudulent Transactions

Figure 4 shows the frequency of fraudulent transactions over time, grouped by hour of the day and day of the week, revealing higher fraudulent activity during the night and on weekends.

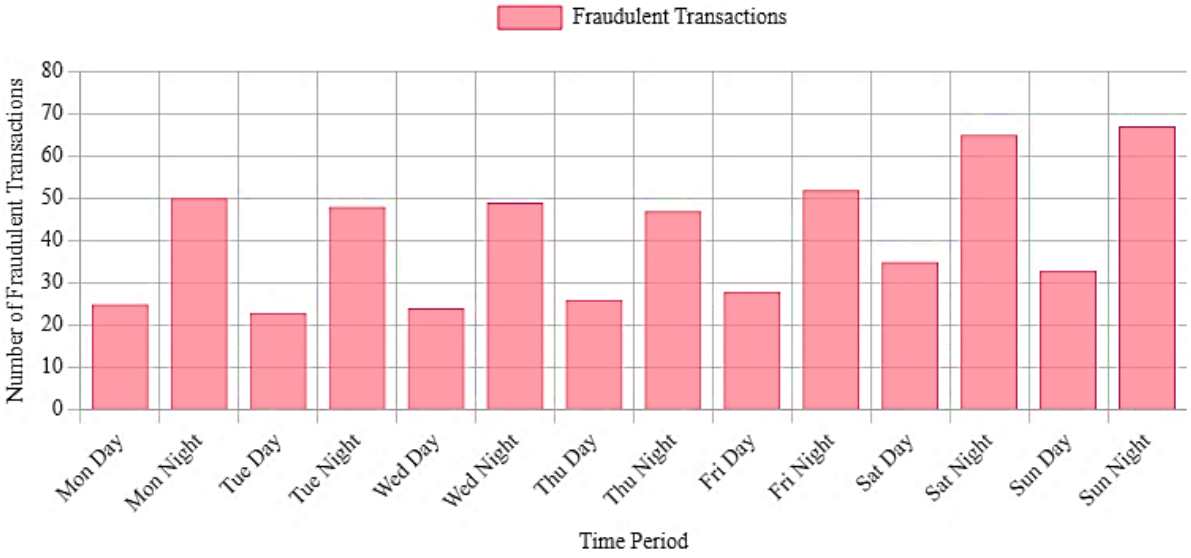

Fig. 4. Frequency of Fraudulent Transactions by Time and Day

5.2. Algorithm Parameters

The proposed model integrates five algorithms: Multilayer Perceptron Neural Network (MLPNN), Logistic Regression (LR), Support Vector Machine (SVM), Hidden Markov Model (HMM), and Self-Organizing Map (SOM). To ensure optimal performance, the parameters of these algorithms are carefully tuned, as summarized in Table 5. These parameters are configured to balance accuracy, robustness, and computational efficiency, leveraging the scalable implementation provided by MLlib Spark for processing large-scale data.

TABLE 5
Parameters of Algorithms Used in the Proposed Model

Algorithm	Parameters
MLPNN	Comprises four layers (input: 28 neurons, hidden: 25, 20, 10 neurons, output: 1 neuron for binary classification). Learning rate is set to 0.1, using RMSprop optimizer for 200 epochs. Activation functions are ReLU for hidden layers and ELU for the output layer.
LR	Uses binomial logistic regression with the sigmoid function, defined as $\sigma(x) = \frac{1}{1+e^{-x}}$. The regularization parameter (C) is optimized via GridSearchCV to balance overfitting and underfitting.
SVM	Employs an RBF kernel with parameters (C) and $(\in)$ optimized via GridSearchCV to balance margin maximization and error tolerance.
HMM	Trained offline using the Baum-Welch algorithm for parameter estimation. The forward algorithm computes sequence likelihoods, and the Viterbi

	algorithm identifies the most likely hidden state sequence for fraud detection, with thresholds set per customer profile based on historical data.
SOM	Uses a grid of neurons (size approximated as $\sqrt{n}$, where n is the number of transactions) with a circular neighborhood function, initial learning rate of 0.1, and final learning rate of 0.01. Fraud detection thresholds are set based on distance distributions.

5.3 Performance Results and Metrics

The performance of the proposed model is evaluated using a confusion matrix and key metrics to assess its effectiveness in detecting fraudulent transactions within the UCI credit card fraud detection dataset. The confusion matrix, presented in Table 6, is formatted with actual classes (Fraudulent, Authorized) as rows and predicted classes (Fraudulent, Authorized) as columns. True Positive (TP) indicates correctly identified fraudulent transactions, False Positive (FP) indicates authorized transactions misclassified as fraudulent, True Negative (TN) indicates correctly identified authorized transactions, and False Negative (FN) indicates missed fraudulent transactions.

TABLE 6
Confusion Matrix for Fraudulent Transaction Detection

Actual \ Predicted	Fraudulent	Authorized
Fraudulent	TP	FN
Authorized	FP	TN

The evaluation metrics Accuracy, Precision, Recall, and F1-Score are defined in Table 7 to quantify the model's performance on both the positive class (fraudulent transactions) and the negative class (authorized transactions). These metrics ensure a comprehensive assessment of the model's ability to handle the highly imbalanced dataset.

TABLE 7
Performance Evaluation Metrics

Metric	Formula
Accuracy	$Acc = \frac{TP + TN}{TP + TN + FP + FN}$
Precision	$PPV = \frac{TP}{TP + FP}$
Recall	$TPR = \frac{TP}{TP + FN}$
F1-Score	$F1 = \frac{2 \times PPV \times TPR}{PPV + TPR}$

The proposed model's performance is compared against individual algorithms (SVM, HMM, SOM, Neural Network, LR) and related works [25–30] using the UCI Default of Credit Card Clients dataset, as shown in Table 8. The proposed model achieves superior results with an Accuracy of 0.99, Precision of 0.95, Recall of 0.83, and F1-Score of 0.89, outperforming both individual algorithms and related works across all metrics. These results address limitations such as high computational cost, low recall, and lack of adaptability, as discussed in Section 3.4.

TABLE 8
Performance Metrics Comparison of Related Works, Individual Algorithms, and Proposed Model

Model	Accuracy	Precision	Recall	F1-Score
Wu et al. [25] (CCNN)	0.90	0.92	0.75	0.89
Jabeen et al. [26] (CNN-LSTM)	0.97	0.88	0.82	0.85
Adejoh et al. [27] (Dynamic Clustering)	0.85	0.90	0.70	0.79
Zheng et al. [28] (Adversarial Autoencoders)	0.87	0.80	0.78	0.79
Bhandary and Ghosh [29] (RF, SVM)	0.98	0.92	0.72	0.81
Btoush et al. [30] (ML+DL Ensemble)	0.95	0.94	0.80	0.86
Support Vector Machine (SVM)	0.98	0.80	0.70	0.75
Hidden Markov Model (HMM)	0.98	0.85	0.75	0.80
Self-Organizing Map (SOM)	0.97	0.87	0.78	0.82
Neural Network	0.97	0.89	0.79	0.84
Logistic Regression (LR)	0.97	0.90	0.70	0.79
Proposed Model	0.99	0.95	0.83	0.89

Figure 5 illustrates the comparison of accuracy scores across different models. Logistic Regression (LR) exhibits lower accuracy (0.97) compared to SVM and HMM (0.98), SOM and Neural Network (0.97), and the proposed model (0.99), due to its simplicity and sensitivity to imbalanced data and outliers.

Uncorrected Proof

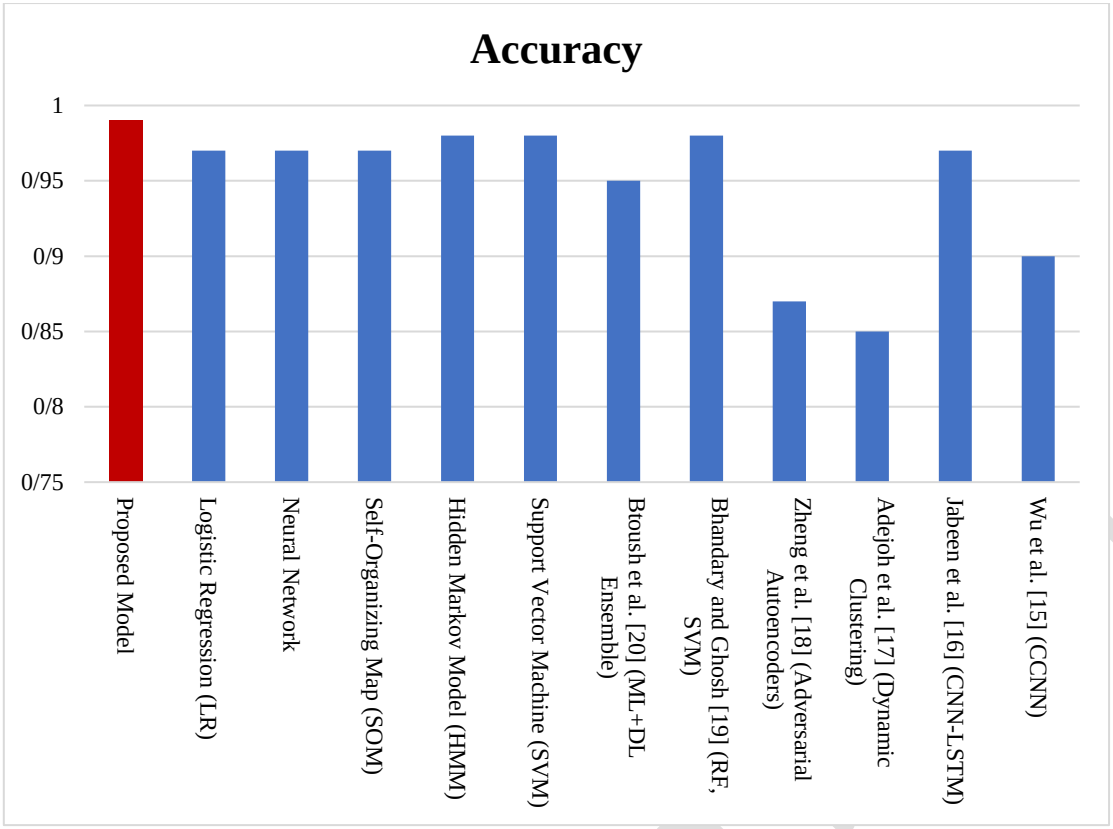

Fig. 5. Comparison of Accuracy for Related Works, Individual Algorithms, and Proposed Model

Figure 6 compares precision scores, showing LR (0.90) and the proposed model (0.95) excelling in reducing false positives, making them suitable for scenarios prioritizing precision. SVM (0.80) has lower precision due to its focus on maximizing the class margin.

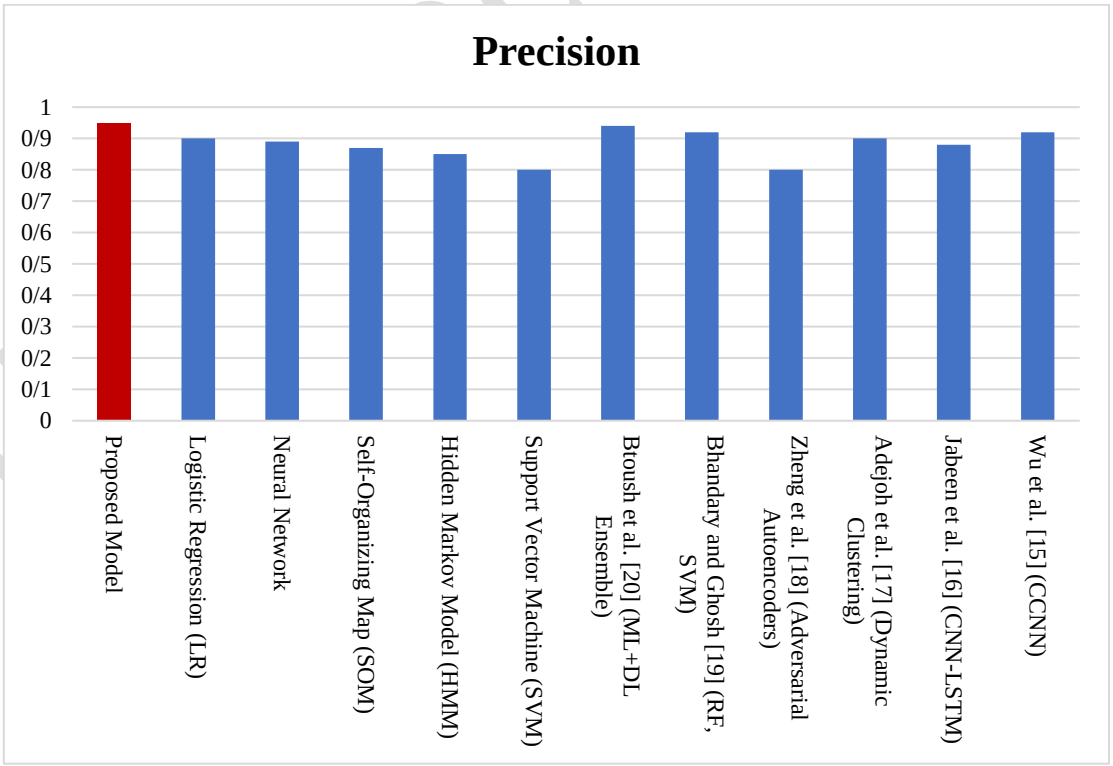

Fig. 6. Comparison of Precision for Related Works, Individual Algorithms, and Proposed Model

Figure 7 compares recall scores, where SVM and LR (0.70) have lower recall due to missing positive cases, while SOM (0.78), Neural Network (0.79), and the proposed model (0.83) perform better in detecting true positives.

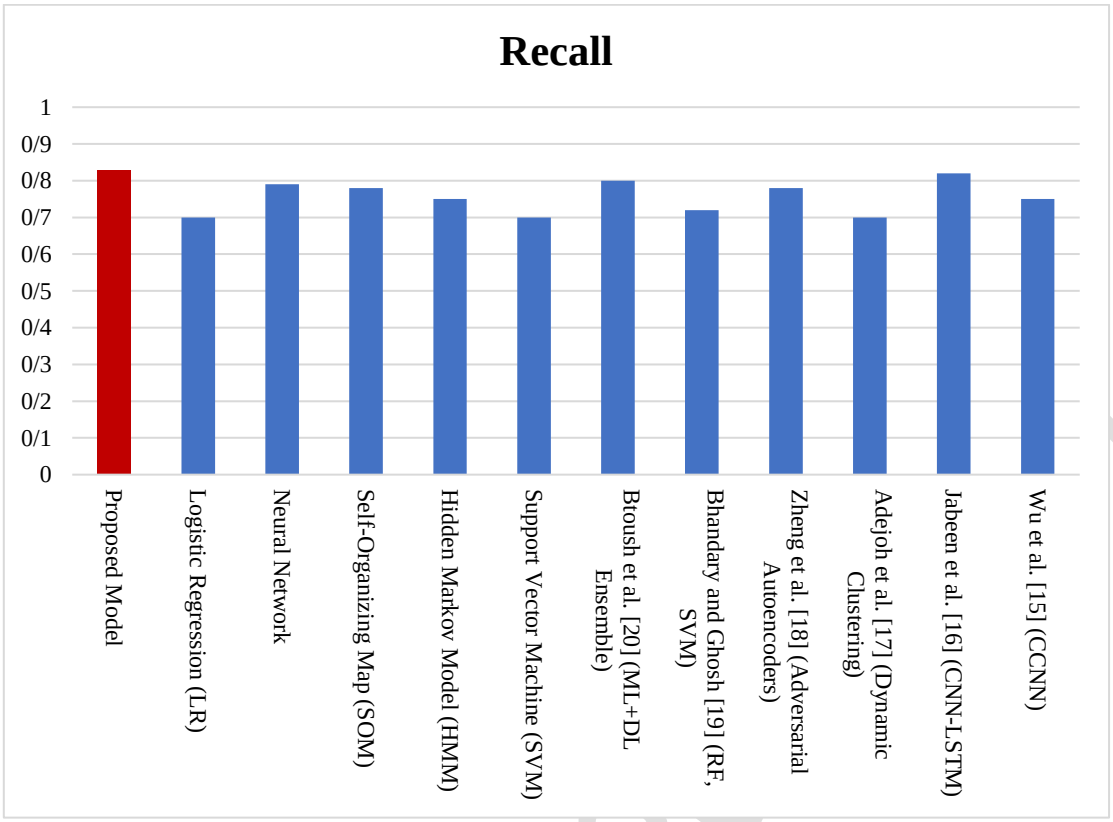

Fig. 7. Comparison of Recall for Related Works, Individual Algorithms, and Proposed Model

Figure 8 compares F1-Scores, with the proposed model (0.89) achieving the highest score, followed by Neural Network (0.84) and SOM (0.82), reflecting their balanced performance in precision and recall.

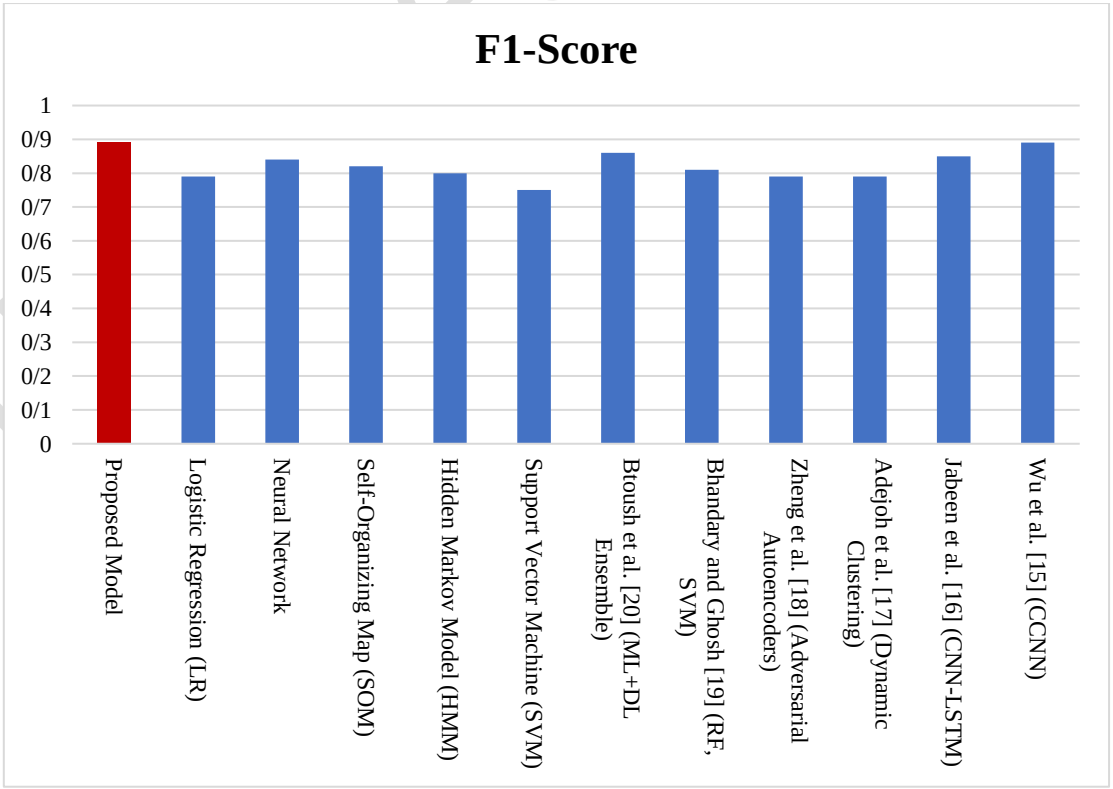

Fig. 8. Comparison of F1-Score for Related Works, Individual Algorithms, and Proposed Model

These results underscore the proposed model's superior robustness and effectiveness in credit card fraud detection, achieving exceptional accuracy (0.99) while maintaining a high recall (0.83) to identify fraudulent transactions and a high precision (0.95) to minimize false positives. This balance, reflected in the F1-Score (0.89), ensures reliable detection of rare fraud cases in imbalanced datasets, surpassing individual algorithms and related works by effectively addressing their computational and adaptability limitations, as outlined in Section 3.4.

5.4. Advantages of the Proposed Model

The superior performance of the proposed model, as demonstrated in Table 8 (Accuracy: 0.99, Precision: 0.95, Recall: 0.83, F1-Score: 0.89), is attributed to its key advantages, summarized in Table 9. These advantages include faster and more accurate detection, improved forecasting with large datasets, cost-effectiveness, flexible analytics, scalability, low latency, enhanced customer satisfaction, a hybrid learning approach, the use of multiple algorithms, the Dempster-Shafer function for combining evidence, and human resources involvement for manual monitoring. These strengths enhance the model's robustness and reliability in fraud detection, addressing limitations of individual algorithms and related works, such as high computational cost and low adaptability, as discussed in Section 3.4.

TABLE 9
Advantages of the Proposed Multi-Layered Model for Fraud Detection

Advantage	Description
Faster and more accurate detection	The model can analyze current consumer patterns and transaction methods faster and more efficiently than any human analysis, allowing it to instantly spot deviations from typical behavior.
Large datasets yield better forecasts	The model can handle enormous volumes of data, which results in better predictions. The more data a model is fed, the more it can learn and make better predictions.
Cost-effective detection technique	The model can analyze data in milliseconds, meaning that team members aren't burdened with manual reviews and checks every time new data is received. This is great for companies that see seasonal highs and lows in traffic, checkouts, or signups.
Flexible analytics and process controls	The model can offer flexible analytics, process controls, screening engines, and experiences. This hones the platform's ability to correctly distinguish valid and fraudulent transactions. The intention is to optimize authentic customer sales and reduce false chargebacks.
Scalable and low latency solution	The model can handle streams of transactions that enter the system continuously and quickly. It can scale up or down according to the demand and reduce the delay in processing the data. This ensures high performance and reliability of the model.
Improved customer experience and satisfaction	The model can enhance the customer experience and satisfaction by reducing false positives, minimizing friction, and increasing trust and loyalty. Customers can enjoy a smooth and secure transaction process without unnecessary interruptions or delays.
Hybrid learning approach	The model can leverage the advantages of both supervised and unsupervised methods in fraud detection. Supervised methods can provide accurate and reliable predictions based on labeled data, while unsupervised methods can discover novel and unknown patterns based on unlabeled data.
Multiple algorithms for different filters	The model can use different algorithms for the quick and explicit filters, which can increase the accuracy and robustness of the fraud detection process. The quick filter uses Hidden Markov Model and SOM algorithms, which can capture the temporal and spatial features of the transactions. The explicit filter uses Neural Network, Logistic Regression, and SVM algorithms, which can learn complex and nonlinear relationships between the features and the labels.
Dempster-Shafer function for combining evidence	The model can use the Dempster-Shafer theory of evidence to combine the scores from the quick and explicit filters and assign a single score to each transaction. The Dempster-Shafer function can handle uncertain and imprecise information, as well as the degree of conflict between the sources of evidence.
Khabar's human resources for manual monitoring	The model can involve Khabar's human resources for manual monitoring of doubtful transactions, which can increase the accuracy and validity of the fraud detection process. The human resources can provide feedback and verification for the transactions that are not clearly classified by the model, and also help to improve the model's performance over time.

6. CONCLUSION

This study presents a pioneering three-layer model for real-time fraud detection in electronic payment card transactions, leveraging Apache Spark's in-memory processing to handle large-scale data efficiently. By integrating feature engineering (transaction type, amount, date, time), quick filters (Hidden Markov Model, SOM), explicit filters (Neural Networks, Logistic Regression, SVM), a Dempster-Shafer-based collector function, and adaptive thresholds across eleven steps including data preprocessing, filtering, real-time analysis, and feedback the model ensures rapid and accurate identification of suspicious transactions for manual review, enhancing scalability and customer trust. Simulations on the UCI dataset (284,807 transactions, 0.172% fraudulent) demonstrate superior performance with an Accuracy of 0.99, Precision of 0.95, Recall of 0.83, and F1-Score of 0.89, outperforming individual algorithms and related works (Section 5.3). The main contributions are a hybrid three-layer architecture for real-time detection, the use of Spark for high-speed processing, and validated superior performance in real-world scenarios. However, challenges include complex feature engineering and parameter tuning, limited real banking data affecting generalizability, and trade-offs between speed and accuracy impacting reliability. Future work will explore additional features, collect diverse banking data, and optimize the speed-accuracy balance to further enhance model performance.

7. REFERENCES

[1] G. Liu, J. Guo, Y. Zuo, and et al., "Fraud detection via behavioral sequence embedding," *Knowl. Inf. Syst.*, vol. 62, pp. 2685–2708, 2020, doi: 10.1007/s10115-019-01433-3.

[2] T. Banirostam, H. Banirostam, M. M. Pedram, and A. M. Rahmani, "A review of fraud detection algorithms for electronic payment card transactions," *Adv. Comput. Eng. Technol.*, vol. 7, no. 2, pp. 157–166, 2021.

Unformatted Proof

[3] T. Banirostam, E. Shamsinejad, M. M. Pedram, and A. M. Rahmani, "A review of anonymity algorithms in big data," *Adv. Comput. Eng. Technol.*, vol. 7, no. 2, pp. 187–196, 2021.

[4] E. Shamsinejad, T. Banirostam, M. M. Pedram, and A. M. Rahmani, "Presenting a model of data anonymization in big data in the context of in-memory processing," *J. Electr. Comput. Eng. Innov.* (JECEI), vol. 12, no. 1, pp. 79–98, 2024, doi: 10.22061/jecei.2023.9737.651.

[5] H. Banirostam, A. R. Hedayati, and A. K. Khadem Zadeh, "Using virtualization technique to increase security and reduce energy consumption in cloud computing," *Int. J. Res. Comput. Sci.*, vol. 4, no. 2, pp. 25–30, 2014, doi: 10.7815/ijorcs.42.2014.082.

[6] X. Zhang, Y. Han, W. Xu, and Q. Wang, "HOBA: A novel feature engineering methodology for credit card fraud detection with a deep learning architecture," *Inf. Sci.*, vol. 557, pp. 302–316, 2019, doi: 10.1016/j.ins.2019.05.023.

[7] M. A. Shiri and N. Mansouri, "Hybrid filter-wrapper feature selection using modified flower pollination algorithm," *Comput. Knowl. Eng.*, vol. 8, no. 2, pp. 55–74, May 2025, doi: 10.22067/cke.2025.89191.1124.

[8] H. Banirostam, A. Hedayati, A. Khadem Zadeh, and E. Shamsinezhad, "A trust based approach for increasing security in cloud computing infrastructure," in *Proc. 15th UKSim Int. Conf. Comput. Modeling Simulation*, 2013, pp. 717–721, doi: 10.1109/UKSim.2013.39.

[9] H. Banirostam, E. Shamsinezhad, and T. Banirostam, "Functional control of users by biometric behavior features in cloud computing," in *Proc. 4th Int. Conf. Intell. Syst. Modeling Simulation* (ISMS), 2013, pp. 94–98, doi: 10.1109/ISMS.2013.102.

[10] E. Shamsinezhad, T. Banirostam, M. M. Pedram, and A. M. Rahmani, "Representing a model for the anonymization of big data stream using in-memory processing," *Ann. Data Sci.*, vol. 12, pp. 223–252, 2024, doi: 10.1007/s40745-024-00556-x.

[11] E. Shamsinezhad, T. Banirostam, M. M. Pedram, and A. M. Rahmani, "A review of anonymization algorithms and methods in big data," *Ann. Data Sci.*, vol. 12, pp. 253–279, 2024, doi: 10.1007/s40745-024-00557-w.

[12] E. Shamsinezhad, T. Banirostam, M. M. Pedram, and A. M. Rahmani, "Anonymizing big data streams using in-memory processing: A novel model based on one-time clustering," *J. Signal Process. Syst.*, vol. 96, no. 6, pp. 333–356, 2024, doi: 10.1007/s11265-024-01920-z.

[13] E. Shamsinezhad, H. Banirostam, T. Banirostam, M. M. Pedram, and A. M. Rahmani, "Providing and evaluating a model for big data anonymization streams by using in-memory processing," *Knowl. Inf. Syst.*, pp. 1–34, 2025, doi: 10.1007/s10115-025-02417-2.

[14] T. R. Rao, P. Mitra, R. Bhatt, et al., "The big data system, components, tools, and technologies: a survey," *Knowl. Inf. Syst.*, vol. 60, pp. 1165–1245, 2019, doi: 10.1007/s10115-018-1248-0.

[15] H. Banirostam, T. Banirostam, M. M. Pedram, and A. M. Rahmani, "Providing and evaluating a comprehensive model for detecting fraudulent electronic payment card transactions with a two-level filter based on flow processing in big data," *Int. J. Inf. Technol.*, pp. 1–6, 2023, doi: 10.1007/s41870-023-00718-0.

[16] E. Shamsinezhad, A. Shahbahrami, A. Hedayati, A. Khadem Zadeh, and H. Banirostam, "Presentation methods for task migration in cloud computing by combination of Yu router and post-copy," *Int. J. Comput. Sci. Issues* (IJCSI), vol. 10, no. 1, pp. 98–102, 2013.

[17] F. Carcillo, Y.-L. Borgne, O. Caelen, Y. Kessaci, F. Oble, and G. Bontempi, "Combining unsupervised and supervised learning in credit card fraud detection," *Inf. Sci.*, vol. 557, pp. 317–331, 2021, doi: 10.1016/j.ins.2019.05.042.

[18] V. Ghai and S. S. Kang, "Credit card fraud detection: Challenges and techniques," in *Proc. 13th Int. Conf. Comput. Commun. Netw. Technol.* (ICCCNT), Kharagpur, India, 2022, pp. 1–6, doi: 10.1109/ICCCNT54827.2022.9984581.

[19] Saloni and M. Rout, "Analysis and comparison of credit card fraud detection using machine learning," in *Advances in Electronics, Communication and Computing*, P. K. Mallick, A. K. Bhoi, G. S. Chae, and K. Kalita, Eds. Singapore: Springer, 2021, pp. 41–51, doi: 10.1007/978-981-15-8752-8_4.

[20] H. Banirostam, T. Banirostam, M. M. Pedram, and A. M. Rahmani, "Analysis and evaluation of various fraud detection methods for electronic payment cards transactions in big data," *J. Signal Process. Syst.*, vol. 96, no. 12, pp. 849–870, 2025, doi: 10.1007/s11265-025-01947-w.

[21] H. Banirostam, T. Banirostam, M. M. Pedram, et al., "A model to detect the fraud of electronic payment card transactions based on stream processing in big data," *J. Signal Process. Syst.*, vol. 23, no. 1, pp. 1–16, Jan. 2023.

[22] S. Khosravi, M. Kargari, B. Teimourpour, M. Talebi, and A. Eshghi, "Fraud detection using temporal sequence embedding and cluster-weighted LSTM models in card transactions," *Comput. Knowl. Eng.*, 2025, doi: 10.22067/cke.2025.92170.1150.

[23] Z. Karimi Zandian and M. R. Keyvanpour, "SSLBM: A new fraud detection method based on semi-supervised learning," *Comput. Knowl. Eng.*, vol. 2, no. 2, pp. 10–18, 2020, doi: 10.22067/cke.v2i2.82152.

[24] UCI Machine Learning Repository, "Default of credit card clients data set," [Online]. Available: https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients# (Accessed 2025).

[25] Y. Wu, L. Wang, H. Li, and J. Liu, "A deep learning method of credit card fraud detection based on continuous-coupled neural networks," *Mathematics*, vol. 13, no. 5, p. 819, 2025, doi: 10.3390/math13050819.

Uncorrected Proof

[26] M. Jabeen, S. Ramzan, A. Raza, N. L. Fitriyani, M. Syafrudin, and S. W. Lee, "Enhanced credit card fraud detection using deep hybrid CLST model," *Mathematics*, vol. 13, no. 12, p. 1950, 2025, doi: 10.3390/math13121950.

[27] J. Adejoh, N. Owoh, M. Ashawa, S. Hosseinzadeh, A. Shahrabi, and S. Mohamed, "An adaptive unsupervised learning approach for credit card fraud detection," *Big Data Cogn. Comput.*, vol. 9, no. 9, p. 217, 2025, doi: 10.3390/bdcc9090217.

[28] P. Zheng, S. Yuan, X. Wu, J. Li, and A. Lu, "Addressing credit card fraud detection challenges with adversarial autoencoders," *Big Data Cogn. Comput.*, vol. 9, no. 7, p. 168, 2025, doi: 10.3390/bdcc9070168.

[29] R. Bhandary and B. K. Ghosh, "Credit card default prediction: An empirical analysis on predictive performance using statistical and machine learning methods," *J. Risk Financial Manag.*, vol. 18, no. 1, p. 23, 2025, doi: 10.3390/jrfm18010023.

[30] E. Btoush, X. Zhou, R. Gururajan, K. C. Chan, and O. Alsodi, "Achieving excellence in cyber fraud detection: A hybrid ML+DL ensemble approach for credit cards," *Appl. Sci.*, vol. 15, no. 3, p. 1081, 2025, doi: 10.3390/app15031081

Uncorrected Proof