International Journal of Scientific Research & Engineering Trends
Volume 12, Issue 2, Mar-Apr-2026, ISSN (Online): 2395-566X

Blockchain and AI-Based Fraud Detection System for Digital Payments

Avinash, Anshika, Shivam
Department of Computer Science and Engineering
(Blockchain Specialization) Chandigarh University Punjab, India.

Abstract- We know digital payment systems are growing faster all over the world, and as they grow, they have some consequences. One big problem among them is digital payment fraud, which rapidly increases as payment systems grow. Fraudsters can surpass rule-based detection systems as they adapt; they have new patterns for doing fraud. They find loopholes in the main architecture from where they manipulate data and do fraud. We study both problems and reach a very solid solution to track down all fraud patterns. We added artificial intelligence and the Hyperledger Fabric blockchain, which is used to detect the pattern of fraud, and a blockchain, which is used to make the payment system tamper-proof. All data related to the payment system are stored in a single system, which is very secure and not able to be encrypted. The detection system runs on four methods. For unstable workflow it uses LSTM networks. For rule-based classification, we used a random forest classifier. For fraud detection, we used a GraphSAGE network, and last, for any suspicious activity, we used an autoencoder. All the things are watched by a meta-learner, which analyzes and combines their output and provides data to trigger a smart contract response, which works automatically. Different financial institutions are used to train their systems without using shared row transaction data to make privacy learn their module detection. We concluded our study, but two public benchmarks are set by PaySim (6.35M transactions) and IEEE-CIS (590K transactions). In PaySim we succeed with up to 98.3% accuracy and an AUCROC of 0.991. Adversarial robustness testing shows the team requires 3.2 times larger to prevent any mistake for success for a single model. These results show much need of AI and blockchain. Using AI and blockchain is very efficient; they are better than anything else to detect fraud.

Keywords- Digital Payment Fraud, Blockchain, Hyperledger Fabric, LSTM, GraphSAGE, Random Forest, Autoencoder, Federated Learning, Differential Privacy, Smart Contracts, Fraud Detection.

I. INTRODUCTION

As we have the internet, many things have come into our lives because of it. One of them is digital payment, along with many other services connected to the internet. As the internet grows, digital payments grow with it — and like all things that exist, they have pros and cons.

Digital payment systems grew very rapidly due to their simplicity, speed, and security. The real boost came during COVID-19, when people were not able to meet each other and did not want to physically touch anything, so they needed the simplest and most efficient method available — which was digital payment. In India, with a population of 1.4 billion people, digital payment plays a very critical role in daily life. Many applications use UPI — Unified Payment Interface — to run platforms like Paytm, Google Pay, PhonePe, and many more. Globally, platforms like Stripe, PayPal, and Alibaba are widely used for digital transactions.

As digital payments increase, their cons also grow. Online fraud, phishing, hacking, and data theft are serious concerns, and as the system advances, so do the threats. In India, from FY 2014–15 to December 2024, over 63,000 fraud cases were reported, resulting in losses of C733.26 crore. In the capital city alone, approximately C1,717 crore was lost between 2023 and 2025, with only 10% recovered. Globally, digital payment fraud losses are projected to exceed $362 billion cumulatively between 2023 and 2028, with losses reaching $91 billion in 2028 alone.

© 2026 IJSRET

As we know, bank and payment apps follow rule-based systems that are fundamentally very complicated and depend on hard coding and have static logic. These rules are not able to adapt to new threats or complex input variables, and fraudsters run their programs under these old rules and find loopholes, which causes them to be undetected for fraud. There are some key limitations, such as inflexibility, security risks, operational inefficiency, and last, modernization barriers. Fraudsters also use new technologies and psychological tactics that are used to exploit human behavior rather than system loopholes. Some methods, like bot networks, SIM swapping, and SMS pumping, collect data from dark web markets, so they have a plan to execute and use those IDs that have not been used for multiple days and target specific vulnerabilities in businesses. Current AI systems stop learning once deployed, relying instead on offline.

MLOps pipelines where human experts curate data and rebuild models, leaving them unable to adapt to new, real-world environments. Modern AI lacks three critical autonomous capabilities: active data selection, flexible switching between learning modes, and self-performance monitoring. Centralized databases can leak full data. Some major breaches at companies like Equifax (affecting 147.7 million Americans), Facebook (540 million records exposed), and Yahoo (3 billion accounts) illustrate how centralized targets are frequently hacked multiple times a year. Databases are exposed to threats like SQL injection, denial of service, and misconfigured public exposures that allow unauthorized access to sensitive data like bank statements and medical records. While organizations attempt to mitigate these risks with strong authentication and regular audits, they always have security concerns, with some issues regarding digital payment fraud and financial crimes including industrial espionage and sabotage detection challenges.

Artificial Intelligence (AI) and Machine Learning (ML) have been revolutionary in fraud detection in terms of creating intelligent solutions capable of discovering even non-linear relationships between transaction parameters in large datasets. At the same time, the use of blockchain technology as a means of creating an immutable and transparent ledger could solve some of the problems related to maintaining database integrity. The synergy effect of combining those technologies allows for developing a modernized approach to creating a fraud detection system.

Currently, there are some studies that explore hybrid approaches, addressing some issues arising while designing an intelligent solution to detecting fraudulent transactions. However, none of those studies has provided a holistic solution that can be implemented in production because current hybrids do not solve all existing problems. Most of the existing solutions tend to focus on one problem while overlooking other essential aspects such as the need for maintaining integrity and the necessity of protecting the privacy of data during model training.

This study contributes to the area by making the following points:

- A novel approach to designing a hybrid AI-blockchain system that solves all the major problems mentioned above.
- Four models (LSTM, Random Forest, Graph Neural Network, and autoencoder) are integrated into an ensemble to provide better accuracy when detecting different kinds of fraud.
- Federated learning combined with the mechanism of ensuring differential privacy allows solving the issue of conducting privacy-preserving collaborative learning among financial organizations.
- Smart contracts are used to provide a multi-stage response to detected fraud with full transparency due to being deployed on the Hyperledger Fabric platform.
- Security analysis demonstrates the resilience of the proposed framework to adversarial attacks.
- Extensive experiments were conducted on the datasets of two financial organizations. Results demonstrate high performance with statistical proof of the obtained findings.

The structure of this paper goes as follows: Section II describes related works comprehensively; in Section III the system architecture and its features are explained in detail. Section IV describes the methodology of research. Experiments and their results will be discussed in Section V. Security and privacy of our system will be assessed in Section VI; Section VII will be about implementation issues.

II. RELATED WORK

A. Machine Learning Methods for Fraud Detection

Initially, efforts in applying machine learning to fraud detection have primarily involved supervised classification methods on

© 2026 IJSRET

credit card transaction data. Dal Pozzolo et al. [1] have demonstrated the efficacy of ensembling techniques in credit card fraud detection, including algorithms such as Random Forest and Gradient Boosting, which yielded impressive AUCROC scores on class-imbalanced fraud data and implemented importance sampling as a method to overcome class imbalance issues. In a comprehensive comparison study between SVM, logistic regression, and Random Forest, Bhattacharyya et al. [2] concluded that ensemble algorithms tend to perform better than single classification methods.

The incorporation of deep learning architectures into fraud detection has further resulted in increased capability in terms of the ability to detect more sophisticated types of fraud. Specifically, Jha et al. [3] suggested employing LSTM to leverage the temporal dependency of transactions. The recurrent neural network is able to identify fraud based on patterns that are difficult for traditional classifiers because of the need to utilize the behavioral history in account takeover frauds. Recently, attention-based architectures, such as the Transformer, have been employed on transaction sequences, which offer greater parallelization capabilities compared to RNNs.

Fraud detection using graph-based neural networks has become popular due to their superior capacity in identifying fraud rings and collusive fraud networks. Liu et al. [4] employed a Graph Convolutional Network (GCN) to learn the relationship between accounts, merchants, and transactions through constructing a heterogeneous graph representation of each entity in the dataset. Dou et al. [12] further proposed an advanced version called CARE-GNN to handle the issue of camouflaging behavior by fraudsters in their fraudulent graphs. There have been unsupervised and semi-supervised approaches used to combat the difficulty in obtaining adequate labeled fraud data. Initially, autoencoder-based anomalous pattern detection was employed to identify anomalies in network intrusions but has now found its application in financial fraud detection. Variational autoencoders and generative adversarial networks have been considered as methods to synthesize additional fraud samples to overcome the problem of class imbalance.

B. Blockchain for Financial Security

Numerous studies have examined the use of blockchain technology in securing financial transactions since the pioneering Bitcoin paper by Nakamoto [13]. Guo and Liang [5] reviewed applications of blockchain technology in finance, focusing on the roles of decentralization, immutability, and transparency as important tools for transaction management in a secure manner. The shift from permissionless to permissioned financial blockchains (e.g., Bitcoin and Ethereum to Hyperledger Fabric and R3 Corda) has been vital for addressing security and privacy concerns.

Christidis and Devetsikiotis [6] investigated smart contractbased financial automation in the form of a proof-ofconcept implementation for automated fraud detection based on the Ethereum platform. In response to scaling problems associated with the Ethereum network, Layer-2 protocols and permissioned blockchains became prevalent. Androulaki et al.

[7] designed a modular permissioned blockchain architecture for enterprise financial applications in the form of Hyperledger Fabric with its built-in pluggable consensus protocol and channel-based privacy.

Recently, the use of blockchain technology for fraud management has expanded to include its direct integration with fraud detection systems. Patel and Sharma [14] designed a framework for blockchain-enabled fraud reporting between financial institutions, allowing the parties to share fraud intelligence in real time. However, this work lacks the aspect of integrating AI-based methods for fraud detection and relies solely on the existing process of fraud reporting among institutions.

C. Federated Learning in Fraud Detection

The problem of detecting fraud among several financial institutions in a privacy-preserving way motivated numerous studies investigating Federated Learning (FL). FedAvg, one of the first federated learning algorithms proposed by McMahan et al. [8], allows collaboratively updating the weights of a neural network across institutions while avoiding the need for centralized storage of sensitive client data. The method achieves high communication efficiency by periodically averaging the gradients.

More recently, Zheng et al. [9] applied FL to fraud detection on simulated bank networks and found that FL provides results within 2% accuracy of the central training. However, the privacy risks of FL include the possibility of gradient inversion and membership inference attacks, motivating the use of differential privacy in federated learning. Geyer et al. [15]

© 2026 IJSRET

provided an implementation of user-level differential privacy in federated learning by applying gradient clipping and adding Gaussian noise.

D Hybrid AI/Blockchain Systems

A number of recent studies are investigating how hybrid AI and blockchain technology can be used to provide improved security solutions. For instance, Salah et al. [16] conducted a survey on how AI/blockchain systems could be applied to different sectors and determined that fraud detection and financial security were among the most important application domains. Huang et al. [17] developed a preliminary framework for detecting insurance fraud, using artificial neural networks (ANNs) as the AI component and Ethereum smart contracts as the blockchain component. However, this framework is limited in scope because of scalability constraints on the underlying Ethereum infrastructure, and no evaluation has been conducted using known benchmark datasets. Our research expands on this area of literature by providing an experimental framework that can be used to investigate how to integrate these two technologies within a specific use case — high-volume digital payments.

III. SYSTEM ARCHITECTURE

The system architecture is comprised of five separate but interrelated layers: the Data Ingestion Layer, the Feature Engineering Layer, the AI Detection Layer, the Blockchain Integrity Layer, and the Smart Contract Response Layer. The architecture supports horizontal scaling, regulatory compliance, and the ability to process transactions in real-time at rates up to 10,000 transactions per second (TPS). Each layer of the architecture is designed to operate independently of one another while also providing the ability for each layer to interact with other layers through well-defined interfaces, thereby supporting modular deployment and ongoing maintenance of each layer.

A. Data Ingestion Layer

The Data Ingestion Layer is where all transaction data that enters the fraud detection pipeline first gets ingested. Transaction events come into the fraud detection pipeline from several payment channels, including mobile banking APIs, payment gateway webhooks, card authorization networks, and exchanges for cryptographic currency via a message queue system distributed across multiple servers using Apache Kafka to ingest these transaction events into the fraud detection pipeline in real-time. The ability to maintain transaction sequences (in chronological order) at the encrypted and unencrypted levels by maintaining a partitioned and ordered log structure within Apache Kafka will ensure the accurate and timely development of temporal models in the detection layer using Long Short-Term Memory (LSTM) functions. In addition to being able to support the distribution of transaction events from the varied payment channels, the ingestion layer is also responsible for validating the integrity of transaction event data through the use of validation, deduplication, and normalization of record schemas across all transaction data sources through various types of heterogeneous data.

B. Feature Engineering Layer

The raw transaction data transitions into a rich feature representation in the feature engineering layer of the fraud detection pipeline. Static features that can be derived directly from the data contained within the transaction record include transaction amount, Merchant Category Code (MCC) associated with the transaction, transaction type, payment instrument type, and geolocation data associated with the transaction. Using the transaction record as a reference point, dynamic behavioral features will be calculated in one of three rolling time frames (1-hour, 6-hour, and 24-hour) based on the following calculations: (i) transaction velocity (the number of transactions and total amount of transactions) during the time period, (ii) average transaction amount per Merchant Category Code, (iii) geographic displacement rate (the distance traveled between consecutive transactions divided by the elapsed time between consecutive transactions), and (iv) device-merchant affinity scores.

Network-based features capture relational patterns by modeling the transaction network as a directed graph. In this graph, nodes represent accounts, and edges represent transactions. For each account node, we calculate graph features like in-degree, out-degree, clustering coefficient, and PageRank score. These features highlight the structural signatures of money mule accounts and fraud rings that are not visible from individual transaction features. We then input these graph features into the graph neural network part of the detection ensemble.

C. AI Detection Layer

The AI Detection Layer uses four models to detect fraud. These models are combined to catch types of bad behavior.

- The LSTM network has hidden units of 128 and 64. It also has a dropout rate of 0.3. This model looks at 10

© 2026 IJSRET

transactions for each account. It tries to find changes in behavior over time.

- The Random Forest classifier is made up of 500 trees. Each tree has a depth of 20. This model uses all the features to classify transactions as good or bad.
- The Graph Neural Network has two layers. It uses 64 and 128 hidden units. This model looks at the connections between accounts to find groups of accounts.
- The Autoencoder has a design of 256–128–64–128–256. It finds transactions by checking if they can be rebuilt. If a transaction cannot be rebuilt, it might be fake.

All these models work together. Their outputs are combined using a model called a "stacking meta-learner." This is a logistic regression model. It is trained on the guesses from the four models. It learns how to combine the models using simple rules. It helps the models work better together.

The final output is a score between 0 and 1. This score shows how likely a transaction is to be fake. Institutions can choose a threshold to balance how precise and complete they want the results to be. They can pick a threshold based on how much risk they are willing to take.

D. Blockchain Integrity Layer

A blockchain integrity and smart contract response framework for AI-based fraud detection in financial institutions to become a fraud-free zone. Blockchain implementation in the Hyperledger Fabric consortium network of transaction records for fraud decisions on blockchain. Blockchain is implemented to record the structured asset per transaction to keep its states secure when retrieved from ledgers. The secured data, with SHA-256 hash, AI ensemble confidence score, and timestamp if a decision is made in fraud detection, then a cryptographic signature is applied. The blockchain stores the information and states regardless of node failure anyway, which in turn is a mutable audit trail. Ledger information holds audit evidence for PCI-DSS and AML directives, suggested to be used for posthoc forensic analysis after a security breach. Each bilateral institutional relationship has a channel, and if accessed by only that institution, then the ledger will all populate. A private channel allows sensitive cross-institutional fraud intelligence sharing for protection. Raft ordering service, suitable for small consortium nodes, has crash fault tolerance database. It supports (n−1)/2 node failures in an n-node ordering cluster; it already sent a request for the transaction, and it just needs an answer for the transaction already requested. Channel to allow access control of the participants that can access that ledger. A new ledger for the new channel is necessary only for the consortium and channel property.

E. Smart Contract Response Layer

With chaincode, supported smart contracts specify the business logic and tiered response protocol. With a transactional level smart contract within the channel, along with block information on how to apply the business rule. The transactional code is parsed in the chaincode and later directed as a transaction using information about the current block. A confidence score of 0.5–0.75 triggers additional authentication, where 0.75–0.9 confidence means automatic blocking with SMS and push notification, and above 0.9 means the full account is suspended for fraud investigation team reporting and an alert is broadcasted to institutions as cross-institutional fraud intelligence, in addition to the mutable audit trail requirement. All actions taken on the account are recorded on the ledger, with aid for auditing the automated fraud response.

IV. METHODOLOGY

A. Dataset Description

For the experimental evaluation, PaySim and IEEE-CIS Fraud Detection datasets have been utilized as benchmark datasets, and both datasets have been selected as two of the bestknown datasets in the field. The structure of these two datasets is different from each other, and on the other hand, the data collection method is also different. However, both of the datasets consist of transaction records in an e-commerce platform. Thus, although the datasets have different structures, they are similar in terms of the main feature.

- **PaySim [10] –** This is a synthetic mobile money dataset that has been generated taking inspiration from the real logs of mobile payment transactions that are based on the African continent. The dataset contains a total of 6,354,407 transactions, of which 8,213 are fraudulent. Transactions can be divided into five different types as follows: CASH-IN, CASH-OUT, DEBIT, PAYMENT, and TRANSFER. The dataset is obtained from the Kaggle machine learning repository, and it has 11 features that are generated only for 30 days.
- **IEEE-CIS Fraud Detection [11]** – As a second dataset, the transactions that are collected from Vesta Corporation's payment protection system on e-commerce transactions have been utilized. This dataset is one of the

© 2026 IJSRET

most demanding datasets as real-world big data, and it contains a total of 590,540 e-commerce transactions, of which 394 are fraudulent.

B. Preprocessing and Feature Engineering

A great number of missing values exist in the columns of the IEEE-CIS dataset. On the other hand, some columns have a sparsity of approximately 40%. For this reason, before applying the main data processing technique, including feature engineering and class imbalance handling, basic data imputation has been applied to handle the problem of missing values.

The numerical features are imputed by the median of the entire column in order to handle the missing values in those numerical columns. In addition, the categorical data columns are imputed by the mode of their columns to avoid loss of information. Furthermore, the value of the categorical features is encoded in the training dataset by using target encoding, and they are scaled by using robust scaling. It should be noted that for the test dataset, the same scaling parameters have been applied directly without any new data transformation.

1) **Class Imbalance Handling:** The dataset is highly imbalanced, as it contains a huge number of fraudulent instances while the rest of the data are all non-fraudulent. Therefore, class imbalance handling techniques have been applied to the training set.

Synthetic Minority Over-sampling Techniques (SMOTE) are used for data over-sampling in order to create synthetic instances for a large number of targeted values of class attribute labels. In this study, SMOTE is used to create synthetic instances for the minority instances, which are fraudulent instances, by taking the non-fraudulent instances into consideration. However, it might not be efficient to predetermine the desired class ratio. In this study, the fraud ratio in the "Fraud Transactions in the Original Dataset" section has been utilized to decide the actual ratio of fraud by oversampling the fraud instances.

On the other hand, a number of Tomek link elimination methods have been applied to the training dataset. By the use of this method, the noise, which is caused by the synthetic instances, has been filtered in the training dataset.

It should be noted that the original fraud ratio in the original dataset is preserved in the testing dataset to evaluate the performance of the proposed algorithms.

C. Model Architecture and Training Details

The LSTM model architecture is built using TensorFlow 2.x and includes two LSTM layers stacked together of type LSTM with sizes of 128 and 64 hidden units, a layer of batch normalization, a dense layer containing 32 hidden units with ReLU activation, a dropout rate of 0.3, and a unit output layer using the sigmoid activation function. The input to the LSTM model is ten consecutive transactions made by one account. If the account does not have ten historical transactions, then we zero-padded the input in order to meet this requirement. The LSTM was trained with an Adam optimizer (learning rate of 0.001, $\beta 1 = 0.9$, and $\beta 2 = 0.999$), a loss function of binary crossentropy, and with early stopping based on validation set AUCROC to determine when training should be stopped (with a patience of five epochs).

GraphSAGE builds a transaction graph where nodes represent a user's account and where directed edges represent transactions between users' accounts. The GraphSAGE model initializes node features with account-level aggregated statistics. Two GraphSAGE convolutional layers, each using mean aggregation (with the first layer having 64 hidden features and the second layer having 128), will produce node embeddings and, once combined with the transaction-level features (for each transaction), will provide the information necessary for making classification decisions.

The autoencoder is trained only against legitimate transactions in order to learn the condensed representation of normal behavior; therefore, when the reconstruction error for a transaction exceeds the 99th percentile of the reconstruction error distribution generated by the validation set, that transaction will be designated as an anomalous transaction.

D. Federated Learning Configuration

In this study, we created a federated learning experiment that simulates 10 locations of financial institutions (nodes) with their own unique training data partitions (non-IID) to represent real-world instances of how the different demographics, transaction patterns, and types of fraud experienced by these institutions affect the operation of the institution. We will use FedAvg to train our model through a total of 20 rounds of communication, with each round having 5 local training

© 2026 IJSRET

epochs, with an expected 80% of clients participating each round. The use of the Gaussian mechanism provides differential privacy; we will use a noise multiplier of σ = 1.1 and a gradient clipping norm of C = 1.0 to achieve (ε = 3.2, δ = 10−5) differential privacy guarantees for each communication rate of training.

E. Assessment Procedures

We evaluate the model with stratified 5-fold cross-validation to protect from data leakage, where all folders have their independence to perform tasks according to their needs. Evaluation metrics comprise accuracy, precision, recall,

F1score, Matthews Correlation Coefficient (MCC), and Area Under the ROC Curve (AUC-ROC). Since MCC is used for balance measures, it has different sizes it can balance. Performance differences are statistically significant when tested by the Wilcoxon signed-rank test at significance level α = 0.05.

V. RESULTS AND DISCUSSION

A. Overall Performance of Methods

Table I compares the results of the proposed method to eight baseline and state-of-the-art methods when applied to the PaySim dataset. The proposed hybrid method outperformed all metrics when compared with each of these eight methods, indicating that the four base models were able to enhance each other's performance as well as the stacking model's (i.e., metamodel) ability to combine their predictions. All performance differences between the proposed method and the bestperforming single model baseline were statistically significant ($p < 0.01$, Wilcoxon signed-rank test).

TABLE I
PERFORMANCE COMPARISON ON PAYSIM DATASET (%)

Method	Acc.	Prec.	Rec.	F1	MCC	AUC
Rule-Based Baseline	84.2	76.3	71.8	74.0	0.701	0.812
Logistic Regression	88.5	82.1	79.4	80.7	0.774	0.876
SVM (RBF)	90.1	85.3	83.7	84.5	0.812	0.901
Random Forest	94.7	92.3	90.8	91.5	0.891	0.961
XGBoost	95.2	93.1	91.6	92.3	0.901	0.968
LSTM	96.1	94.5	93.2	93.8	0.921	0.974
GraphSAGE	95.8	93.9	92.5	93.2	0.913	0.971
Autoencoder	91.3	88.7	87.1	87.9	0.851	0.942
Proposed	98.3	97.1	96.5	96.8	0.961	0.991

The proposed framework has an accuracy of 98.3% (PaySim Data), up to 2.2%, 3.0%, and 14.1% improvement in F1-Score compared with the first-class pre-groups (XGBoost) and rulebased baseline, respectively. The 97.1% high accuracy of this approach can be low enough in false positives (transactions falsely flagged as fraudulent) so that a good person enjoys and the operational cost associated with manual review of fraud is low.

B. IEEE-CIS Dataset Results

Table II shows the performance of the proposed method on the IEEE-CIS dataset, which is a more complex dataset (i.e., higher fraud rate, greater number of features, more variation among classes). The proposed method performed much better than the best performing previous model (i.e., XGBoost), achieving an F1-Score of 94.3% vs 90.1% (4.2% improvement). The more complex feature space of the IEEECIS dataset will assist the GNN portion to accurately model the relationship between common device identifiers, email domains, and billing addresses associated with synthetic identity fraud

TABLE II
PERFORMANCE COMPARISON ON IEEE-CIS DATASET (%)

Method	Acc.	Prec.	Rec.	F1	AUC
Random Forest	91.2	88.4	85.3	86.8	0.941
XGBoost	92.7	90.3	89.9	90.1	0.956
LSTM	93.1	91.2	90.4	90.8	0.961
GraphSAGE	92.9	90.8	90.1	90.4	0.958
Proposed	96.5	95.2	93.4	94.3	0.982

C. Ablation Study

To investigate how much each component of the fraud detection tool affects the overall performance, a decomposition is performed. Table III shows the F1 ranking of the PaySim dataset with the anomalous components of the model removed from the overall model. The results of the ablation test show that each one of the included components definitely contributes to the performance of the overall fraud detection tool, with LSTM presenting the best contribution rate (2.1 pp F1-score

© 2026 IJSRET

gain) and then GNN (1.4 pp) through Random Forest (1.2 pp). Federated learning has the best small effect compared to centralized training, showing a 0.9 pp F1-Score decrease, which shows that there is an acceptable trade-off between privacy and overall performance.

TABLE III
ABLATION STUDY – F1-SCORE ON PAYSIM (%)

Configuration	F1-Score	Δ vs. Full System
Full Proposed Framework	96.8	—
Without LSTM	94.7	−2.1
Without GNN	95.4	−1.4
Without Random Forest	95.6	−1.2
Without Autoencoder	96.0	−0.8
With Federated Learning	95.9	−0.9
Without Smart Contract Layer	96.8	0.0 (detection only)

B. Blockchain Performance Analysis

Blockchain integration introduces a measurable latency overhead due to the consensus mechanism and block verification process. On a Hyperledger Fabric test network of 10 peers with a four-node sorting cluster, the system achieves a continuous throughput of 1,200 TPS with an average stoptostop transaction confirmation latency of 0.8 seconds. For reference, UPI convergence in India has almost three peak payments of TPS 10,000, which indicates that the blockchain level is suitable for modern payment volumes, while optimization is needed for future growth.

The block size is configured to 500 consecutive transactions with a block timeout of 200 ms, balancing throughput and delay. Using Hyperledger Fabric's pluggable consensus mechanism, order providers are upgraded to more powerful security requirement environments for a fee of additional latency. Offchain computation is used for computational AI inference, with the best results of fraud sequencing and transaction registration in reserve.

VI. SECURITY AND PRIVACY ANALYSIS

A. Threat Model

We take a comprehensive combined contingency of an external adversary and an internal malicious insider adversary. The external threats include hostile actors who manipulate transaction capabilities, account takeover via compromised user credentials, and data-poisoning attacks against the fraud detection model in the federated training. The internal threats are malicious employees committing fraud and personal transaction opportunities through their personalized transaction options.

B. Blockchain Security Features

The Hyperledger Fabric blockchain provides a very strong deterrent and a means of protecting against the identified threats. By creating a cryptographic chain of the blocks, the immutability of the transaction record is ensured by requiring the recalculation of the hash value of each of the subsequent blocks based on altering the value of the hash from an earlier transaction, which will result in an inability for any of the nodes in the community to not identify that an alteration has taken place at the same time as also being computationally impossible to alter all of the subsequent blocks from the moment of alteration. The decentralized nature of the ledger provides multiple points of failure to eliminate the possibility that any one of the nodes in the community would be capable of causing an alteration in the data as well as hardware system downtime. The Raft system carrier provides that the (n−1)/2 node failure is tolerated due to the fact that all node failures can be covered by just surviving an n-node failure.

C. Adversarial Robustness

AI detection rates are evaluated against common negative evasion strategies, including feature imitation (trying to make fraudulent behavior look legitimate) and gradient-based adversarial perturbations. Class structure provides inherent robustness through model variation: an adversarial perturbation is adapted to stay away from the type of perturbation a model characteristics represent simultaneously. Empirical evaluation using the Carlini-Wagner (C&W) attack framework shows that the cluster requires 3.2× higher perturbations to achieve evasion compared to any one model, confirming the inverse robustness advantage of the ensemble.

D. Federated Learning Privacy

The differential privacy mechanism used in the updates of the federated version provides a formal privacy guarantee: ($\varepsilon = 3.2$, $\delta = 10{-}5$)-DP guarantee means that participation of transactional data by any individual in the learning technique changes the probability of outputting anything by a factor of at most e3.2. Secure aggregation ensures that the target server

© 2026 IJSRET

only observes the sum of encrypted model updates, deferring rebuilding of individual client contributions. Membership estimation attack experiments show that the connectivity model with DP reduces membership estimation accuracy from 71.3% (without DP) to 52.1% (with DP), approaching the 50% random baseline.

VII. IMPLEMENTATION CONSIDERATIONS

A. Deployment Architecture

Manufacturing planning of the proposed framework requires careful consideration of infrastructure, regulatory compliance, and operational requirements. The AI detection layer is containerized using Docker and orchestrated through Kubernetes, which allows scaling of horizontal inference capacity in response to transaction volume peaks. Model serving uses TensorFlow Serving for LSTM and Autoencoder components and a custom FastAPI web server, with Triton ensemble NVIDIA inference server for acceleration to batch assessment during recovery windows.

The Hyperledger Fabric community is deployed in partners' geographically distributed data centers, with each cluster running 2–3 peers for redundancy, including network management, subscription management, chain upgrade policies, and channel configuration adjustments for preventing any unauthorized use of the group.

B. Regulatory Compliance

The framework is designed with regulatory compliance as the best requirement. The immutable blockchain audit trail meets document retention requirements under PCI-DSS (Payment Card Industry Data Security Standard), which mandates the retention of audit logs for at least 12 months with three months available immediately. Differential privacy ensures compliance with the European General Data Protection Regulation (GDPR) and the Personal Data Protection Bill (PDPB) in India, which imposes restrictions on the use of private data for automated decision-making. The smart contract automated response protocol includes a mandatory human review pathway for account suspension and requirements for human oversight of the resulting automated decisions.

C. Integration with Existing Payment Infrastructure

Integration with the current payment infrastructure is done via REST API, which includes fraud detection pipeline, so that plug-ins can be deployed with modern payment authorization systems. Synchronous (inline) fraud scoring for real-time authorization options is supported, which reduces latency to acceptable levels for current card network requirements.

VIII. LIMITATIONS AND FUTURE WORK

Despite its promising effects, several limitations warrant consideration and inspire future research directions. First, the 1,200 TPS throughput at the blockchain level, while good enough for a state-of-the-art payment volume, may also become a bottleneck as digital payment adoption continues to grow. Future work will discover the advanced consensus mechanisms of Hyperledger Fabric v3.0 and layer-2 state channels approach that ensures external immutability is sacrificed to achieve throughput in excess of 10,000 TPS.

Second, the current framework has been evaluated on static benchmark datasets collected at specific points in time. Actual international fraud styles exhibit conceptual drift — the statistical properties of fraudulent behavior change over the years as fraudsters adapt their techniques. Addressing concept drift requires online-learning mechanisms that continuously replace model parameters as new classified fraud examples emerge.

Third, graph neural networks now only use stationary behavioral graphs at a given time factor. Therefore, the addition of dynamic graph evolution to track account transactions over time will help identify fraudulent schemes over a longer period in a much better way. Temporal Graph Networks (TGNs) are therefore a good candidate for similar research in this area.
Fourth, the combination detection method guarantees different privacy, but the privacy requirement is $\varepsilon = 3.2$. The desire for extra strict privacy guarantees (meaning a lower level of ε) requires adding more noise to the model, which makes extension difficult. Future work will have a way to embellish privacy-promoting policies by using Renyi Differential Privacy´ (RDP) accountants. Furthermore, the secure multiparty computing (SMPC) protocol can be explored as a potential privacy mechanism for aggregating models.

Fifth, a description of the opportunity of fraud detection will facilitate the regulator and analyst to gain further insight and

© 2026 IJSRET

attention. Interpretation within an in-depth description, composed by SHAP (SHapley Additive exPlanations) and GNN fraud-embedded causes directing regulatory compliance and operational fraud investigations, can be attempted as a blockchain phase predictor concerning the emergence of the risks posed by new quantum computers.

IX. CONCLUSION

This paper proposed a novel and holistic architecture that uses artificial intelligence and blockchain technology for realtime fraud detection in virtual payment infrastructure. By overcoming the major drawbacks of current structures, combining Long Short-Term Memory networks (LSTM), Random Forests (RF), Graph Neural Networks (GNN), and autoencoders into a 4-model-set AI architecture, the system detects the maximum types of fraud correctly. The Hyperledger Fabric blockchain framework is adopted to maintain immutability and auditability of transaction ledgers and smart contracts are used to allow automatic incremental fraud detection with full framework tracking capabilities. With federated learning and differential privacy, distributed financial institutions can learn collaboratively without exposing customers' sensitive information.

Rigorous experimental validation on PaySim (6.35M transactions) and IEEE-CIS (590K transactions) benchmark datasets proved that our framework achieves 98.3% accuracy, 97.1% precision on the PaySim dataset with AUC-ROC of 0.991, and accuracy of 96.5% with AUC-ROC of 0.982 for the IEEE-CIS dataset, respectively. An ablation study verifies the best-sized contribution of each subsystem, and adversarial robustness analysis shows that ensemble models cause 3.2× larger perturbations than the single model baseline. The blockchain layer achieves 1,200 TPS on Hyperledger Fabric test network with a confirmation latency of 0.8 seconds.

The proposed system is the first form of detecting fraud through the use of AI and blockchain combined together; it produced a more complete, clearer, flexible, and efficient fraud detection tool. Due to the proliferation and extent of virtual values, new fraud technologies are becoming more diverse, and the need for personalized security to enhance components through design configuration, our proposed model offers a foundation for the future of financial security throughout all of today's evolving digital economies.

REFERENCES

1. A. Dal Pozzolo, O. Kellen, R. A. Johnson, and G. Bontempi, "Measuring likelihood by undersampling for imbalanced classes," in Proc. IEEE Conference on Computational Intelligence for Data Mining (CIDM), 2015, pp. 159–166.
2. S. Bhattacharya, S. Jha, K. Fix, and J. C. Westland, "Data mining and analysis of credit card fraud: A comparative case study," Decision Support Systems, vol. 3, pp. 602–613, Feb. 2011.
3. S. Jha, M. G. Guillen, and J. Westland, "Detecting credit card fraud using a transaction aggregation method," Expert Systems with Applications, vol. 39, no. 16, pp. 12650–12657, Nov. 2012.
4. Z. Liu, Y. Dou, P. S. Yu, and H. Deng, "Graph neural networks for fraud detection by matching perturbation," in Proc. 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2020, pp. 1569–1572.
5. Y. Guo and J. S. Liang, "Application and implementation of blockchain technology in banking," Financial Innovation, vol. 2, no. 1, pp. 1–12, Dec. 2016.
6. K. Christidis and M. Devetsikiotis, "Blockchains and smart contracts for the Internet of Things," IEEE Access, vol. 4, pp. 2292–2303, 2016.
7. E. Androulaki et al., "The Hyperledger Fabric: A distributed framework for developing permissioned blockchains," in Proc. EuroSys Conference, Porto, Portugal, Apr. 2018, pp. 1–15.
8. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in Proc. 20th International Conference on Artificial Intelligence and Statistics (AISTATS), vol. 54, 2017, pp. 1273–1282.
9. X. Zheng, Z. Sha, and Q. Liu, "Cross-transaction network fraud detection utilizing federated learning preserving privacy," IEEE Transactions on Information Forensics and Security, vol. 18, pp. 1253– 1264, 2023.
10. E. A. Lopez-Rojas, A. Elmir, and S. Axelsson, "PaySim: A simulator for fraud detection using financial mobile money," in Proc. 28th European Modeling and Simulation Symposium (EMSS), Larnaca, Cyprus, 2016, pp. 249–255.
11. IEEE Computational Intelligence Society, "IEEE-CIS Fraud Detection Dataset on Kaggle Competition," 2019. [Online]. Available: https://www. kaggle.com/c/ieee-fraud-detection

© 2026 IJSRET

12. Y. Dou, Z. Liu, L. Sun, Y. Deng, H. Peng, and P. S. Yu, "Combining balanced and proximity-aware augmentations to improve fraud detection using graph neural networks," in Proc. 30th ACM International Conference on Information and Knowledge Management (CIKM), 2021, pp. 3535–3539.
13. S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf
14. R. Patel and M. Sharma, "Fraud reporting framework for financial institutions based on the blockchain," in Proc. IEEE International Conference on Blockchain and Cryptocurrency (ICBC), 2022, pp. 1– 6.
15. R. C. Geyer, T. Klein, and M. Nabi, "Differentially private federated learning: A client level perspective," arXiv preprint arXiv:1712.07557, 2017.

© 2026 IJSRET