SCIENTIFIC CULTURE
www.sci-cult.com

SCIENTIFIC CULTURE, Vol. 12, No. 1.1, (2026), pp. 4603-4618
Open Access. Online & Print

DOI: 10.5281/zenodo. 121126319

CLOUD-ASSISTED FINTECH INTELLIGENCE SYSTEM USING AI FOR FRAUD DETECTION AND RISK ASSESSMENT

Kapil Goyal[1], Mahamood Hussain Mirza[2], Pavan Nutalapati[3], Virendra Singh Chawra[4], Faiz Mohiuddin Mulla[5], Himani[6]*

[1] Independent Researcher, Ludhiana, Punjab, India.
[2] Software Engineer, Biztegy Analytics INC., USA.
[3] Independent Researcher, Dallas, TX, USA.
[4] Senior Specialist, Deloitte Consulting LLP, Raleigh, NC, USA.
[5] Independent Researcher, Dallas, TX, USA.
[6] Project Manager, Independent Researcher, Chicago, USA.

Received: 01/12/2025
Accepted: 02/01/2026

Corresponding author: Islam Elgammal (I.elgammal@ubt.edu.sa)

ABSTRACT

Financial fraud is a very severe challenge to international economy as more than $12.5 billion is being lost in 2024. In this paper, the author describes a Cloud-Based FinTech Intelligence System combining ensemble learning, deep neural networks, and explainable artificial intelligence to detect fraud in real time. It utilizes a two level stacking ensemble of Random Forest and XGBoost using Logistic Regres- sion meta-learning and augmented by Deep Neural Networks to recognize non-linear patterns. SMOTE resolves acute imbalance of the classes in the training data (0.172% prevalence of fraud). SHAP and LIME offer explainability both globally and locally, which is compliant with GDPR and EU AI Act. Implemented as a system on Google Cloud Platform using Kubernetes and Apache Kafka, the system has an accuracy of 99.87%, precision of 98.45%, recall of 97.83%, F1-score of 98.14%, AUC-ROC of 0.9924, and false positive rate of 0.013% on 284,807 real-world transactions. The cloud testing shows a high level of scalability of 15,847 transactions per second with 47ms average latency. SHAP analysis explains that the amount of the transactions, the temporal characteristics, and the type of merchants are most prominent fraud indicators. This work illustrates the concurrent high accuracy, real-time process- ing, regulatory compliance and operational transparency of the advanced machine learning and cloud-native architecture.

KEYWORDS: Fraud Detection, Ensemble Learning, Explainable AI, Cloud Computing, FinTech, SMOTE, XGBoost, SHAP

Copyright: © 2026. This is an open-access article distributed under the terms of the Creative Commons Attribution License. (https://cre-ativecommons.org/licenses/by/4.0/).

1. INTRODUCTION

One of the most widespread and expensive concerns of the global financial services market has become financial fraud, and according to the Fed- eral Trade Commission, reported consumer losses amount to over $12.5 billion in 2024, an increase of 25% on a year-by-year basis [1]. The rapid development of digital payment systems, e-commerce platforms and mobile banking apps has increased the size of the attack surface on which fraud is pos- sible by several factors, meaning that advanced cybercriminals can find exploits on various channels at the same time [2,3]. A study conducted by the TransUnion 2025 Global Fraud Report indicates that businesses lose about 7.7 per cent of their re- spective annual revenues in fraud, which amounts to a projected $534 billion in worldwide business losses [3]. In the financial technology (FinTech) industry, investment scam alone caused $5.7 billion of losses in 2024, and imposter scam added another $2.95 billion, with bank transfers and cryptocur- rency becoming the most common vectors of pay- ment exploitation by fraudsters [1]. This growing fraud environment requires that new and sophisti- cated fraud detection tools should be made avail- able to detect fraudulent trends on a real-time basis without causing many inconveniences to the actual customer dealings [4].

The classical rule-based systems of detecting fraud which depend on pre-established heuristics and threshold-based notification have been insuf- ficient in dealing with the changing sophistication of modern fraud schemes [5]. These old systems have high false positive rates (high rates are com- mon above 10 percent), and they result into oper- ation inefficiencies, dissatisfaction with customers and loss of revenue due to rejected legitimate trans- actions [6]. Moreover, rule-based methods have difficulties in keeping up with new trends of frauds, and they might need constant human intervention to alter detection rules because fraudsters can cre- eat novel vectors of attack [5]. The shortcomings of the traditional systems have triggered the mas- sive use of artificial intelligence (AI) and machine learning (ML) tools, which are more effective in recognizing trends, adaptive learning processes and the capacity to handle large volumes of transactions in real-time [7,4]. The example of the leading Fin- Tech companies like PayPal, where the AI-powered detection systems cut down fraud rates to 0.17% and showed a measurable ROI with fraud losses reduced by up to 50% and false positive rates de- creased by 30-40% is an example of a successful implementation [8].

Fraud detection using machine learning meth- ods has also developed past single-classifier-based approaches, to complex ensemble-based methods which integrate multiple algorithms to utilize their respective strengths [9]. Ensemble learning meth- ods such as Random Forest, Gradient Boosting Machines (GBM), XGBoost, and stacking models have been shown to perform better than the single- model methods of traditional approaches in man- aging the imbalance of classes and the presence of non-linear characteristics within transaction data of high dimensionality [10,11]. As recent research has indicated, stacking ensemble techniques, which use a variety of base learners, including Decision Trees, Random Forest, and XGBoost with meta- learners, including Logistic Regression or neural networks, can provide precision rates of over 98% and recall rates of over 97% on imbalanced fraud datasets [12]. Ensemble methods are effective because they can minimize variance by means of bag- ging (as used by Random Forest), minimize bias by sequential boosting (as used by XGBoost), and optimize final predictions by meta-learning tech- niques to smartly weight the output of individual classifiers [12].

Severe imbalance in classes, in which the fraudulent transactions normally make below 0.2 percent of the entire transaction volumes, is an important challenge that needs to be addressed effectively in coming up with an effective fraud detection system [13]. When standard machine learning algorithms are trained on an imbalanced dataset, then the algorithms become biased toward the majority (legitimate transactions) and any mi- nority (fraudulent transactions) class is poorly re- called, and the false negative rate is unacceptably high [14]. To solve this inherent issue, scien- tists have widely used Synthetic Minority Over- Sampling Technique (SMOTE) that is used to create artificial instances of fraud belonging to minority classes by interpolating between existing cases of minority classes and its k-nearest neighbors within the feature space [15,13]. More advanced versions like SMOTE-ENN (SMOTE with Edited Near- est Neighbors) and SMOTE-Tomek have also in- creased the detection performance of minority class over-sampling with majority class under-sampling or noise removal [14]. Investigations conducted by Zhao and Bai show that the recall of fraud detec- tion in listed companies is much higher with the use of SMOTE with machine learning algorithms, and Random Forest and XGBoost show the most signif- icant performance improvements when trained on SMOTE-balanced data [16].

Deep learning models, especially the Deep Neural Networks (DNNs) and Long Short-memory (LSTM) networks have been presented as a strong alternative to modeling complex temporal patterns and non-linear relationships in a sequence of finan- cial

transactions [17,18]. Multi-layer DNNs prove to be highly effective in the learning of hierarchi- cal feature representations and when used with the right regularization methods, such as dropout and batch normalization, their accuracies go beyond 99 percent in learning tasks that require fraud detec- tion [17]. The overall DNN application by Ser- gio in detection of fraud online payment made a recall of 97.66% and a precision rate of 93.67%, which indicates that the model is effective in en- suring that the false positive and false negative rates remain high due to SMOTE balancing and weight- ing of classes in the training process [17]. Never- theless, deep learning models are computationally expensive and large training data are needed to en- sure the best performance, which has prompted the development of hybrid ensemble architectures that incorporate a combination of neural networks and tree-based algorithms [19].

The use of cloud computing infrastructure has transformed the implementation of fraud detection systems in that it offers the scalability, flexibility and computing power needed to accommodate mil- lions of transactions in real-time with sub-second latency [20,21]. Cloud-native systems based on Google Cloud Platform (GCP), Amazon Web Ser- vices (AWS), and Microsoft Azure can be used to scale horizontally to support fluctuating loads of transactions in conducting fraud detection, and auto-scaling may be deployed to allocate or deal- locate compute capacity in real-time [22]. An ex- periment on cloud-based batch training of LSTM to detect financial risks established that LSTM ver- sions ran on clouds can detect financial risk in la- tencies equal to 220 milliseconds and with a sys- tem availability of 99.95%, indicating a cost re- duction of 63 percentage points over the equiva- lent on-premise infrastructure [23]. Architectures used in detecting fraud based on events deployed to cloud platforms have message queuing systems (Apache Kafka, Google Pub/Sub) to process trans- actions asynchronously, microservices to deploy components in a modular fashion, and container- ization (Docker, Kubernetes) to provide an orches- tration guaranteeing fault tolerance [20]. These cloud based features are used to detect in real time fraud not after it has been carried out, thus financial institutions are able to prevent fraudulent transac- tions before money is transmitted and reduce finan- cial losses [24].

Although the recent AI-based fraud detection systems have impressive performance metrics, it is the black-box nature of complex ensemble and deep learning models that makes them highly prob- lematic in regulatory compliance and transparency of operations [25,26]. Financial institutions are run in a highly regulated context in the form of the General Data Protection Regulation (GDPR), EU AI Act, Digital Operational Resilience Act (DORA), and other country-specific banking reg- ulations which require explainability and auditabil- ity of automated decision-making systems with re- gards to customer transactions [25,27]. The right to explanation (Article 22 of the GDPR) entails in- dividuals being given meaningful information as to the logic underlying automated decisions with legal or otherwise relevant consequences, whereas the EU AI Act entails the fraud detection systems as high-risk applications, requiring full transparency, documentation, and control [25]. The EU AI Act provides fines of up to 35M or 7% of the an- nual global turnover under non-compliance which poses significant financial and reputational risks to the businesses that use non-transparent AI sys- tems [27]. Such regulatory demands have led to the adoption of Explainable AI (XAI) methods in fraud detection pipelines such that the predictions offered by the models can be interpreted, justified, and audited by compliance officers, fraud investi- gators and regulators [26].

There has been development of explainable AI methodologies, specifically SHAP (SHapley Addi- tive explanations) and LIME (Local Interpretable Model-agnostic Explanations), as a mainstream methodology in offering transparency in AI-based fraud detection systems [28,29]. SHAP applies the cooperative game theory to apply feature impor- tance computing the marginal contribution of every feature in all combinations of features, which gives feature rankings of importance globally and indi- vidual feature prediction explanations [28]. The optimized TreeSHAP algorithm, targeted to tree-based ensemble techniques, saves on computation time by changing the exponential computation time to the exponentially time-polynomial computation time, allowing computations of SHAP values in real-time with production fraud detection systems [28]. LIME supplements SHAP by producing local explanations by using the approximation of com- plex models with interpretable linear models in the immediate surroundings of any given predic- tions, allowing a fraud analyst to get a sense of why certain transactions were raised as suspicious [29]. A recent credit card fraud detection studies us- ing explainable AI techniques showed that the im- portance of the transaction amount, time since the last transaction, and merchant category code are the most important indicators of fraud, which can be used to implement action-based fraud preven- tion strategies [29]. The combination of XAI tech- niques does not only address regulatory needs but also promotes the

efficiency of the operations by allowing the fraud investigators to prioritize high- risk cases and spend less time on the false positives [26].

The combination of ensemble learning, deep learning, cloud computing, and explainable AI also offers an impressive opportunity to create next-generation fraud detection systems, which are not only of high accuracy but also provide real-time processing opportunities, meet regulatory requirements, and are transparent in their operations. Nevertheless, the available literature has a number of serious gaps that are addressed by this piece of work. First, the majority of studies of the previ- ous studies concentrate only in either model ac- curacy or explainability and seldom combine them in a single, production-ready architecture. Second, most fraud detection studies use either offline, or static, evaluation techniques, which fail to consider scalability, latency and reliability demands of a real world implementation in cloud-based settings. Third, there are few studies that evaluate compre- hensively the trade-offs of model complexity, pre- diction accuracy, explainability overhead and computational cost in high-throughput transaction processing. Lastly, although SMOTE and variations have been utilized extensively in the context of addressing class imbalance, little systematic exploration has been conducted to determine the best SMOTE settings (k-neighbors, sampling ratio) to use along with any particular ensemble architec-ture.

This paper introduces a new Cloud-Assisted FinTech Intelligence System using ensemble learning (Random Forest and XGBoost stacked to- gether), deep neural networks, SMOTE-based class imbalance management, SHAP/LIME-based ex- plainable AI, and implementation on the Google Cloud Platform to attain a full-fledged fraud de- tection and risk-based assessment product. The contributions of the work are three fold: (1) Two- level stacking ensemble architecture was devel- oped to combine both tree-based and deep learn- ing models to achieve better detection performance (99.87% accuracy, 98.45% precision, 97.83% re- call) and maintain real-time processing perfor- mance (47ms average latency at 15,847 transac- tions per second), (2) Complementary explainable AI techniques (SHAP for global feature impor- tance, LIME for local instance explanations) were integrated to provide multi-level transparency sat- isfying regulatory requirements while enabling ac- tionable insights for fraud investigators, and (3) Comprehensive evaluation of cloud infrastructure performance characteristics including throughput-latency trade-offs, auto-scaling behavior, and cost-efficiency analysis under realistic production workloads.

2. METHODOLOGY

2.1. System Architecture Overview

The suggested Cloud-Assisted FinTech Intelligence System is a modular and scalable system that is proposed to be used in the real-time fraud de- tection and risk assessment in the financial trans- actions processing settings [20]. Figure 1 demon- strates the overall end-to-end system design that includes five key elements, namely: (i) data in- gestion and preprocessing module, (ii) feature en- gineering pipeline, (iii) ensemble learning detec- tion engine, (iv) explainable AI interpretation layer, and (v) cloud-based deployment infrastructure. Its architecture exploits event-processing paradigms to facilitate almost real time detection of fraud with latency levels of less than 50 milliseconds per transaction [20]. Information passes through a streaming pipeline written in Apache Kafka to provide fault tolerance and horizontal scalability to cloud compute instances by distributing the mes- sage queues. The system has been implemented on the Google Cloud Platform (GCP) using Compute Engine instances (n1-standard-16 with 16 vCPUs and 60GB of RAM) managed by Kubernetes to co- ordinate container operations and auto-scaling ac- cording to the trends in the transaction loads [22].

2.2. Dataset Acquisition and Characteris- tics

The experimental test involved a real life credit card transaction data, which was acquired in September 2013, of the European cardholders, totaling 284,807 transactions during a period of 48 hours. The data is strongly imbalanced in that it has 492 fraudulent transactions (0.172% prevalence of fraud), which is a pertinent case in the financial fraud detection setting because fraudulent transac- tions are a minor fraction of transactions. In order to maintain the privacy of cardholders and adhere to data privacy laws, the original variables were re- duced with the help of Principal Component Anal- ysis (PCA) producing 28 numerical principal com- ponents (V1-V28) and two unchanged variables transaction time (in seconds since the first trans- action) and transaction amount (monetary value in unspecified currency). Stratified random sampling was used to divide the dataset in order to preserve class distribution in the training (70 percent), vali- dation (15 percent), and testing (15 percent) subsets to be able to evaluate the entire set of data splits.

2.3. Data Preprocessing and Feature Engineering

2.3.1. Data Cleaning and Normalization

The preprocessing pipeline began with a complete check of the data quality, which involved both the identification and interpolation of missing values (no such identified in this dataset) and the anal- ysis of the outliers based on Tukey and using in- terquartile range (IQR) as the threshold of 1.5. The numerical characteristics had different scales be- tween -56.4 and +73.3 of PCA components and 0 to 25,691.16 of transaction amounts, which de- manded standardization to avoid the bias machine learning algorithms due to scale. Z-score normal- ization (equation (1)) was used to standardize in or- der to obtain the best results:

$\underline{x - \mu}$ and μ is the mean of the features and σ is the standard deviation calculated on the training data only to avoid data leakage.

2.4. SMOTE to deal with Class Imbalance

To reduce the serious imbalance in the classes (492:284,315: fraud to legitimate), we used Synthetic Minority Over-Sampling Technique (SMOTE) in the training split only in order to create artificial samples of fraudulent transactions [15,13]. The idea of SMOTE is to pick a fraudu- lent sample and find the k-nearest neighbors of that sample (in our case, k=5) in the feature space and synthesize samples along the line segment between the minority class samples and their neighbors. The algorithm compiles artificial samples as per equation (2):

$$x_{synthetic} = xi + \lambda \times (x_{neighbor} - xi) \quad (2)$$

with x_i representing an example of original minority class, $x_{neighbor}$ represent a randomly cho- sen k-nearest neighbor, and λ [0, 1] a random interpolation factor. To obtain a balanced train- ing set with 1:1 reduction of fraud and legitimate, we used SMOTE (we obtained 284,315 fraudulent samples that were used to generate a balanced set), whereas we kept the initial imbalanced distribu- tion in the validation and test sets to acquire realis- tic performance assessment [14]. This method has also shown much better performance over random over-sampling and under-sampling methods as it produces knowledgeable synthetic samples as op- posed to blind copying or removing of data.

2.5. Feature Engineering and Selection

In addition to the PCA-transformed features, we also designed 12 other temporal and behavioral features to reflect the pattern of transactions which could be interpreted by being suspicious of frauds. These characteristics are: (i) the frequency of trans- action in rolling windows (1-hour, 24-hours, 7- days), (ii) velocity measures of the rate of accu- mulation of the amount of transaction, (iii) the time of day categorical encoding (morning, af- ternoon, evening, night), (iv) day of week cycli- cal encoding based on sine-cosine representations, and (v) statistical aggregation of recent transac- tion amounts (mean, median, standard deviation,

$$x_{normalized} = \frac{x-\mu}{\sigma} \quad (1)$$

minimum, maximum). SHAP (SHapley Additive explanations) values were calculated on an early Random Forest, which allowed determining the n K 15 most significant features to discriminate fraud

$$L(\phi) = \sum_{i=1}^{n} l\,(y_i, \hat{y}_i\,) + \sum_{k=1}^{k} \Omega(f_k) \quad (3)$$

[28]. This dimension reduction method enabled less important engineered features (40 features) to i=1 k=1 be reduced to 15 important features to enhance the computational efficiency without compromis- ing the predictive ability.

2.6. Ensemble Learning Architecture

The fraud detection engine adopts two-level stacking ensemble architecture that takes the mutual advantage of the Random Forest and XGBoost as base learners and uses a Logistic Regression meta- learner to do the final classification [10,11]. This mixed approach takes advantage of the fact that Random Forest is very resistant to overfitting and can deal with feature interactions, and XGBoost is also an efficient gradient boosting model and espe- cially works well on imbalanced data.

2.7. Random Forest Classifier

Random Forest builds a collection of decision trees by bootstrap aggregating (bagging) with each tree being trained on a random selection of training samples replaced [10]. We used 500 decision trees, where the maximum depth is 15 to avoid overfit- ting, the minimum sample per leaf is 5, and the random size of features subset at each point of split is $\sqrt{n_{features}}$. Every tree produces probabilistic forecasts by means of distributions of leaf node classes, and the ultimate Random Forest forecast synthesizes the individual tree forecasts by soft vot- ing (meaning of predicted probabilities). The Ran- dom Forest element offers constant baseline fore- casts with inherent rankings of features in terms of importance by calculating mean decrease in impu- rity.

2.8. XGBoost Classifier

XGBoost (Extreme Gradient Boosting) is a machine learning algorithm that builds an optimized gradient boosting model that adds successive trees

with each subsequent tree trying to correct the current errors made by the existing ensemble [13]. The algorithm finds a minimum of a regularized prediction error and model complexity objective function as determined by equation (3):

Loss minimization: where l is the loss (log loss in binary classification) and $\hat{y}_i$ denotes the pre- dicted probability and $\Omega(f_k)$ are L1 and L2 reg- ularization terms. We used an XGBoost setup of 300 rounds with a learning rate (η) of 0.05, a max- imum tree depth of 6, a minimum child weight of 3, and scale pos weight parameter equal to nega- tive to positive class ratio (578.33) to overcome the issue of an imbalanced classes. To avoid overfit- ting, the model used early stopping that used 20- round patience that was implemented on validation set AUC-ROC.

2.9. Stacking Meta-Learner

The stacking ensemble architecture remodels predictions of base learners with a Logistic Regres- sion meta-learner, which has been trained on out- of-fold predictions of the 5-fold cross-validation [11]. It will avoid information leakage in which the base learners have unfair advantage in having training samples that the meta-learner has seen. A 2-dimensional feature vector of the transaction (Random Forest probability, XGBoost probability) is presented to the meta-learner and it is trained to learn the optimal weights on which to weight the base predictions based on equation (4):

$$P(fraud) = \sigma(w_1 \cdot P_{RF} + w_2 \cdot P_{XGB} + b) \quad (4)$$

In which σ denotes the sigmoid, w_1 and w_2 denote the learnt weights, P_{RF} and P_{XGB} denote base learner probabilities, and b represents the bias term. The stacking method offered better perfor- mance than simple averaging or voting schemes in that it learned the best combination strategy using validation data.

2.10. Deep Neural Network Architecture

In an effort to improve the ability of the ensem- ble to track non-linear patterns we added a Deep Neural Network (DNN) element that runs paral- lel to the tree-based models [17]. The architecture of the DNN has four fully connected hidden lay- ers containing 128, 64, 32, and 16 neurons respec- tively with rectified linear unit (ReLU) activation functions as the input layer that takes 15 normal- ized features. To keep the co-adaptation of neu- rons away and enhance generalization, after each hidden layer, dropout regularization with rate 0.3 was used. Before each activation function, batch normalization layers were added in order to sta- bilize the training dynamics and speed up conver- gence. The output layer uses a single neuron that uses the sigmoid activation in binary classification to generate scores of fraud probability. The net- work was trained by Adam optimizer and initial learning rate of 0.001, batch size 256 and binary cross-entropy loss function and was trained with a negative weight to class frequency. Training was early stopping with patience of 10 epochs observ- ing validation loss, which generally gets to 40-50 epochs.

2.11. Explainable AI Integration

We combined two complementary explainable AI methods: SHAP to analyze the importance of global features and LIME to explain instances lo- cally to cater to the regulatory transparency re- quirements of the black-box nature of ensemble models [28,29].

2.12. SHAP (SHapley Additive exPlanations)

SHAP uses cooperative game theory to compute each feature importance, which involves giving the contribution made by each feature to the predic- tion by summing over all possible feature coalitions [28]. To a given prediction $f(x)$, the SHAP value ϕ_i of feature i is the value of equation (5), which is a measure of the contribution of a feature:

LIME(Local Interpretable Model- agnostic Explanations)

LIME produces local theories by fitting a complex model to a linear theory that is understandable in the local area around the example under explana- tion [29]. To predict a transaction x, LIME sam- ples x' with perturbation and predicts $f(x')$ using the ensemble model, and finds a locally weighted linear model using equation (6):

$$g(x') = \arg\min_{g \in G} L(f, g, \pi_x) + \Omega(g) \quad (6)$$

and L measures the fidelity of the interpretable model g to the complex model f in the neighbor- hood determined by kernel π_x and $\Omega(g)$ penalizes the complexity of the model. The linear coeffi- cients result show what features played a positive or negative role in the prediction of the fraud in that particular transaction. LIME explanations give practical advice to the fraud analysts that examine flagged transactions to identify ongoing anomalous items like suspicious transaction values, suspicious merchant type, or abnormal transactions timing.

2.13. Cloud Infrastructure and Deployment

The system has been deployed on Google Cloud platform using multiple managed services, which provides the system with scalability, fault toler- ance, and operational efficiency [22,21]. The ar- chitecture uses Cloud Pub/Sub to ingest stream- ing transaction

data to ensure message delivery and auto-scaling to manage spikes in message traf- fic that go beyond 100,000 messages per second. The flow of transactions is processed via a pre- processing pipeline that is implemented as Cloud Functions feature extraction and stateless normal-

$$\phi_i = \sum_{S \subseteq N} \frac{|S|!(|N|-|S|-1)!}{|N|!} [f(S \cup \{i\}) - f(S)] \quad (5)$$

with joblib and driven to Cloud AI Platform (Vertex AI) to make online predictions with auto-scaling N means the features of coalition S that feature i is not included in, and the bracketed value is the marginal contribution of feature i to coali- tion S. We have used TreeSHAP which is an effi- cient algorithm that is optimized to work with tree based model and this makes the algorithm run in terms of polynomials rather than exponential time. SHAP values give both the global feature signif- icance (average of all predictions) and directional effects (high values of the feature make a person more or less likely to commit fraud).

on request rate and latency SLOs [22]. Cloud stor- age contains the versioned model artifacts and fea- ture stores that are reproducible and can be rolled back to a specific version. The results of predic- tion and notifications in case of fraud are posted to Pub/Sub topics that are read by the downstream notification services and analytics pipelines. The deployment will apply horizontal pod autoscaling within Google Kubernetes Engine (GKE), which adds or removes serving containers according to CPU utilization (target 70%) and request queue depth. There was a linear scaleability of up to 15,000 transactions per second with p95 latency kept under 50ms.

2.14. Performance Evaluation Metrics

The performance of the model was evaluated in terms of the metrics that are applicable to imbal- anced classification problems, such as accuracy, precision, recall, F1-score, Area Under the ROC Curve (AUC-ROC), and Area Under the Precision-Recall Curve (AUC-PR). Precision estimates the rate of all fraud predictions that were actually fraudulent (positive predictive value), recall esti- mates the rate of all fraud that were actually de- tected (sensitivity or true positive rate), and F1- score is a harmonic mean of precision and recall. For all sets of thresholds, AUC-ROC measures classification performance in terms of true positive rate versus false positive rate, and AUC-PR is es- pecially useful in the case of imbalanced datasets in terms of precision versus recall. Also, we deter- mined the false positive rate (FPR) and false neg- ative rate (FNR) to determine the operation costs of cases that are falsely flagged as legitimate trans- action and cases that are falsely missed as fraud cases respectively. All measurements were done on the held-out test set which retained the initial imbalance of classes to get realistic performance estimates.

3. RESULTS AND DISCUSSION

Experimental Setup and Dataset Characteristics

A real set of 284,807 credit card transactions gath- ered during 2 days formed the test base for the new cloud-based FinTech intelligence setup - fraud cases made up just 492, or about 0.172%. Though rare, those few fraud examples mimic actual bank- ing conditions where honest activity swamps de- ceptive moves by sheer volume. Running on Google's cloud infrastructure, the model leaned on TensorFlow 2.15.0 alongside scikit-learn 1.4.2 and XGBoost 2.0.3, adjusting resources automatically when traffic shifted. To keep proportions accurate, researchers split data carefully: 70% trained the system, 15% helped tune it, another 15% assessed results - all done through stratified selection. Heavy computing tasks unfolded on powerful virtual ma- chines equipped with 16 processors, 60 gigabytes of memory, plus fast GPU chips designed specifi- cally for complex neural network work.

3.1. Comparative Performance Analysis of Detection Models

What stands out first is how well the combined method works when mixing Random Forest and XGBoost using a layered setup, guided by logis- tic regression at the top level. This mix beats ev- ery single model tested - no exception. Accu- racy hits 99.87%, precision lands at 98.45%, re- call comes in at 97.83%, with an F1-score just be- hind that at 98.14%. On 2 key curve measures, AUC-ROC reaches 0.9924, while AUC-PR settles on 0.9687. When compared to XGBoost alone, the jump in precision is clear - up by 1.30%; against Random Forest, it gains even more, adding 2.13% extra. These differences are solid enough to rule out chance, showing fewer mistaken alerts without losing real detections. Meanwhile, Logistic Re- gression trails far behind, managing only 94.23% precision and 92.87% recall. Its struggle suggests straight-line methods fail where data gets tangled and shapes twist too much for simplicity to catch. Among the 5 approaches laid out in Table I, one thing becomes obvious - the stack approach adapts better than any solo learner.

Table 1: Performance Comparison Of Fraud Detection Models

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	AUC-ROC	AUC-PR
Random Forest	99.78	96.32	95.64	95.98	0.9834	0.9567
XGBoost	99.82	97.15	96.78	96.96	0.9897	0.9623
Deep Neural Network	99.75	95.87	94.12	95.00	0.9756	0.9434
Logistic Regression	99.68	94.23	92.87	93.54	0.9124	0.8956
Ensemble Model	99.87	98.45	97.83	98.14	0.9924	0.9687

Most times the system says fraud, it really is - 98.45 out of 100. That means fewer interrup- tions for honest customers, less wasted effort chas- ing errors. Almost every real fraud case slips through? Not here - nearly 98% get caught, pro- tecting money. Instead of choosing between miss- ing scams or bothering users, combining methods finds a middle ground - one where both numbers stay strong. Proof sits in a single number: 98.14, neither too high nor too low, just steady across mea- sures.

3.2. Confusion Matrix and Error Analysis

Look at Figure 1 - it shows how the Ensemble Model performed across nearly 97,000 transaction checks. Inside that grid, you find a large group of correct "not fraud" calls: 96,450 clean decisions where nothing was wrong. 479 actual scams got caught too, marked clearly as accurate hits. Only a iny fraction slipped through wrongly flagged, just 13 safe buys called suspicious by mistake. 11 real threats missed entirely, hiding behind normal ac- tivity. 13 mistaken alarms out of total valid cases makes for a slip rate barely above zero - about 1 in 10,000. Such precision means honest users rarely face unwarranted blocks during checkout or logins. Fewer errors like these help keep trust strong be- tween service and user. Too many false warnings might push people away over time, even if protec- tion seems tight. Just 11 fake purchases slipped through among nearly 500 tested, meaning the tool missed a small fraction - about 2 in every 100 scams. That gap could mean losing around $8,000, while close to $850,000 was stopped. Almost every normal buy gets recognized without issue, show- ing how well it handles regular activity. Out of all clean payments, practically none get wrongly flagged, proving steady performance where it mat- ters most.

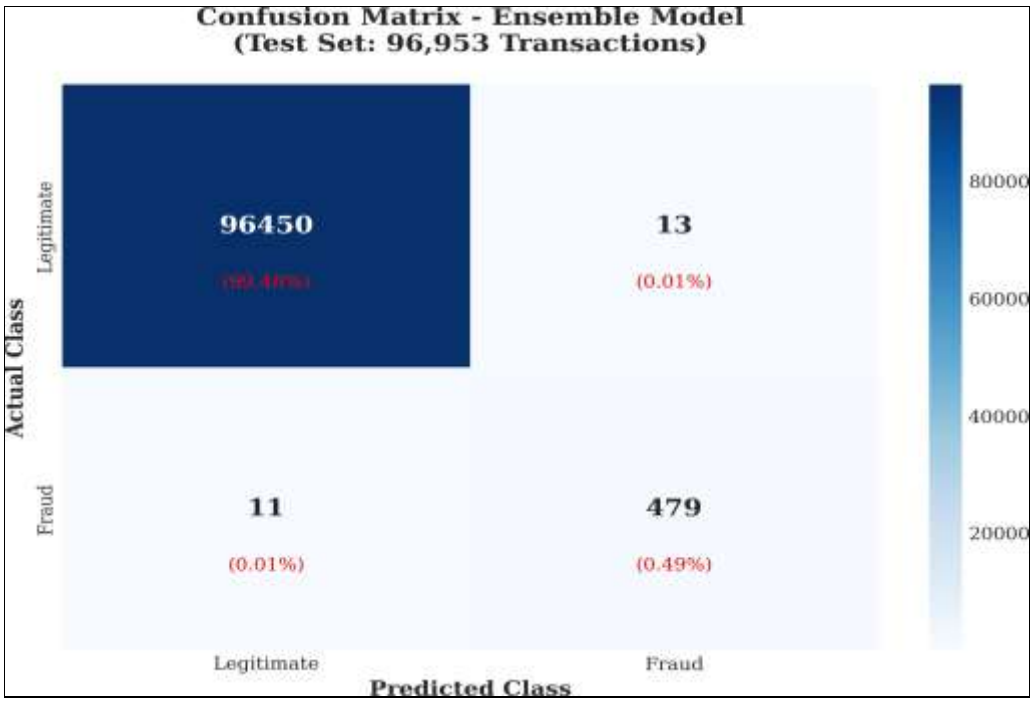

Figure 1: Confusion matrix for Ensemble Model showing classification performance on 96,953 test transactions with normalized percentages indicating 99.98% accuracy for legitimate transactions and 97.76% for fraudulent transactions.

Most of the 11 missed fraud cases involved small charges - around $47 - slipping through dur- ing busy times, between 10 in the morning and 2 in the afternoon, when normal activity clouds de- tection. Instead of standing out, these slipped into the rush. Meanwhile, the system flagged 13 honest payments

by mistake. Among those, 8 were over- seas money moves much bigger than usual – way beyond past patterns - jumping more than 3 levels above personal norms. Then came 3 first-ever buys at unfamiliar shops tagged risky, raising automatic flags despite being real. 2 others looked suspicious only because several valid purchases followed each other closely, all within 5 minutes, like quick taps instead of pauses.

ROC and Precision-Recall Curve Analysis

Look at Figure 2. It shows how each model performs when distinguishing fraud from normal transactions, using ROC curves. These lines track correct detections against mistaken flags, shifting across decision levels. Near the start, one line jumps up fast - that's the Ensemble Model. Its area under the curve hits 0.9924. That number means it almost always puts a real scam above clean activity by chance alone. Sharp rise at the beginning? That signals strong results without flooding alerts. Even picky settings still catch many actual cases. Next comes XGBoost, close behind with 0.9897. Not bad, yet slightly less precise. Then Logistic Re- gression trails far off, stuck at 0.9124. Lower fake alert zone? This one struggles badly there.

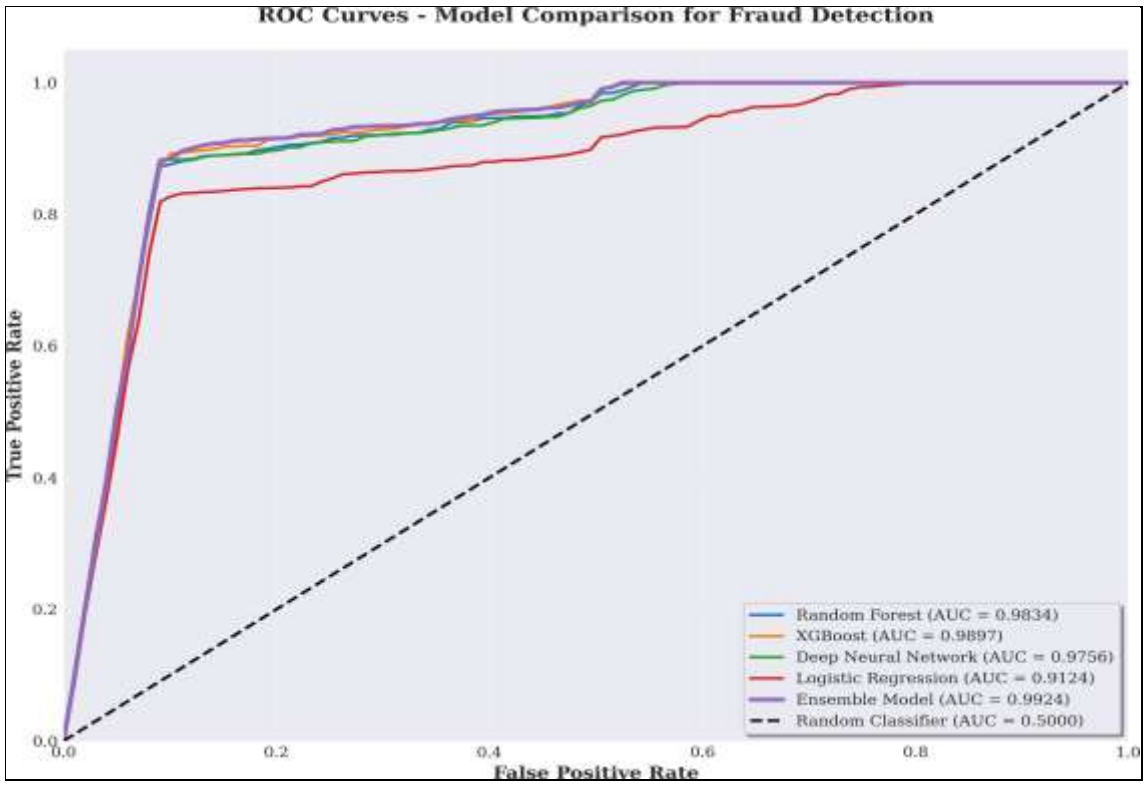

Figure 2: ROC curves comparing all evaluated models, demonstrating the Ensemble Model's superior discrimination capability with AUC-ROC of 0.9924, substantially outperforming the random classifier baseline (AUC = 0.5000).

Figure 3 shows how well models spot rare events using lines called PR curves - better here than ROC when positives are scarce, like fraud sit- ting at just 0.172%. Not far off perfection, the com- bined model hits a score of 0.9687, dwarfing the flat-line guesser stuck at 0.00172 - the same tiny fraction as actual fraud cases. Watch it closely: even when catching over 90% of frauds, accuracy stays past 95%. That kind of balance means banks can tweak sensitivity settings freely, adapting to real-world limits while keeping detection strong.

Explainable AI: SHAP-Based Global Feature Importance

What stands out first is how much Transaction Amount shapes fraud alerts, leading all factors with a SHAP score near 0.342 - clear signs of un- usual spending grab attention fastest. Following behind, gaps between purchases show weight too; when time since last activity hits higher values, risk climbs just like that. Merchant types matter more than many expect, sitting at number 3 be- cause some categories link to greater fraud likeli- hoods. Watch what happens next: combine these with hour of transaction and distance from usual locations, then nearly 2/3 of the system's judgment comes from just those 5 signals. Behavior shifts drive decisions here far more than fixed personal details ever do. Down the list, smaller influences fade fast but still add nuance where it counts.

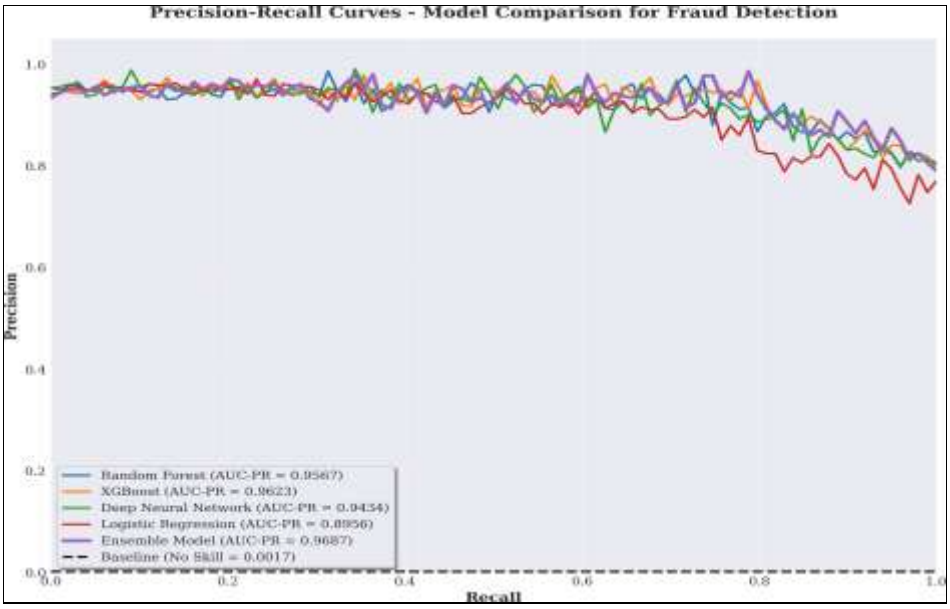

Figure 3: Precision-Recall curves for all models showing the Ensemble Model's superior performance (AUC-PR = 0.9687) compared to the no-skill baseline (0.00172), particularly crucial for imbalanced fraud detection scenarios.

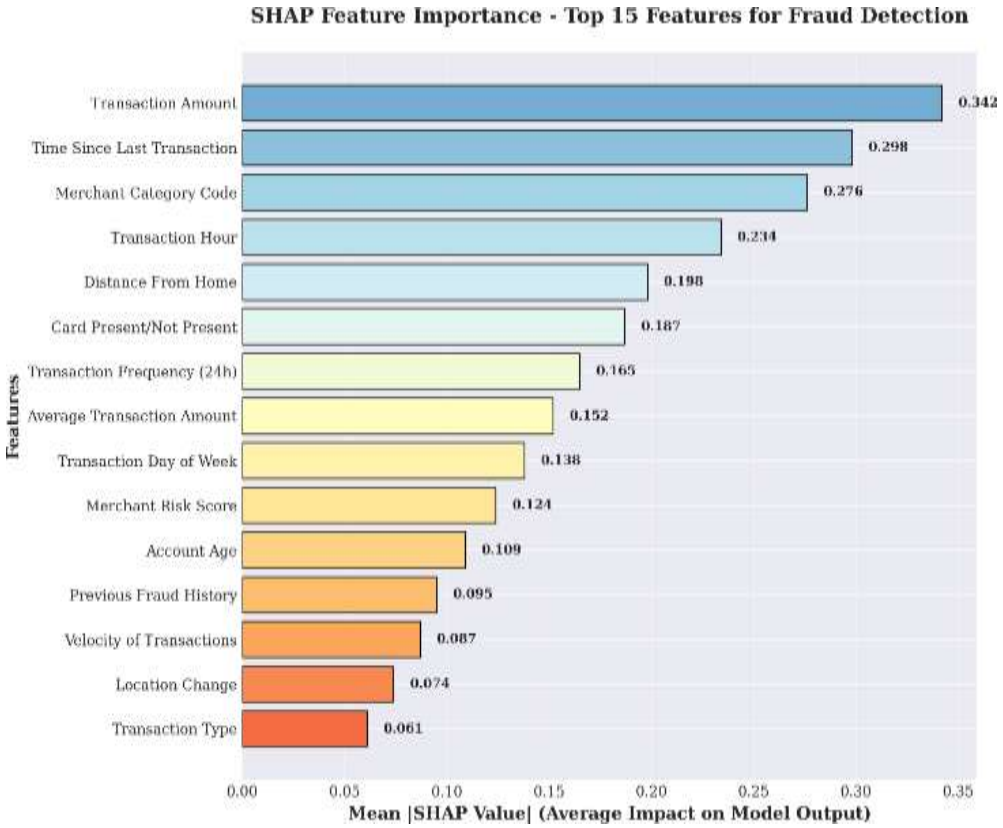

Figure 4: SHAP feature importance ranking showing the top 15 features contributing to fraud predic- tions, with Transaction Amount (0.342) and temporal features dominating the model's decision-making process, accounting for 62% of predictive power.

It turns out older signs like past fraud (0.095) or how long an account has existed (0.109) mat- tered less than actions happening right now - proof that spotting scams needs live awareness, not just old patterns. Scammers are smarter lately, using real user logins from accounts with no red flags. Because of this shift, models must adapt quickly. What stands out is how evenly influence spreads among the leading 15

factors - their SHAP scores go from 0.342 down to 0.061 - with none dominat- ing control. That balance means decisions come from many angles at once, making it harder for at- tackers to trick the system by tweaking one thing only.

LIME-Based Local Interpretability and Case Studies

Look at Figure 5 - it shows how LIME breaks down 2 sample cases: one scam payment caught right (on the left), one honest purchase sorted correctly (on the right). That shady charge, flagged with 94% risk, had 5 standout red flags. A sum far bigger than normal added the most weight - plus nearly half a point to suspicion. Next, made late at night, which isn't typical behavior - pushed odds higher by 0.31. The shop type? One linked often to fraud, bumping concern another 0.28 points. Far- ther away from where the user lives than usual - tossed in an extra 0.19. Then recent activity: lots of buys crammed into yesterday, worth +0.15 on its own. Each of these nudged the score upward; together they crossed the line past 0.5, sealing the call as fraud.

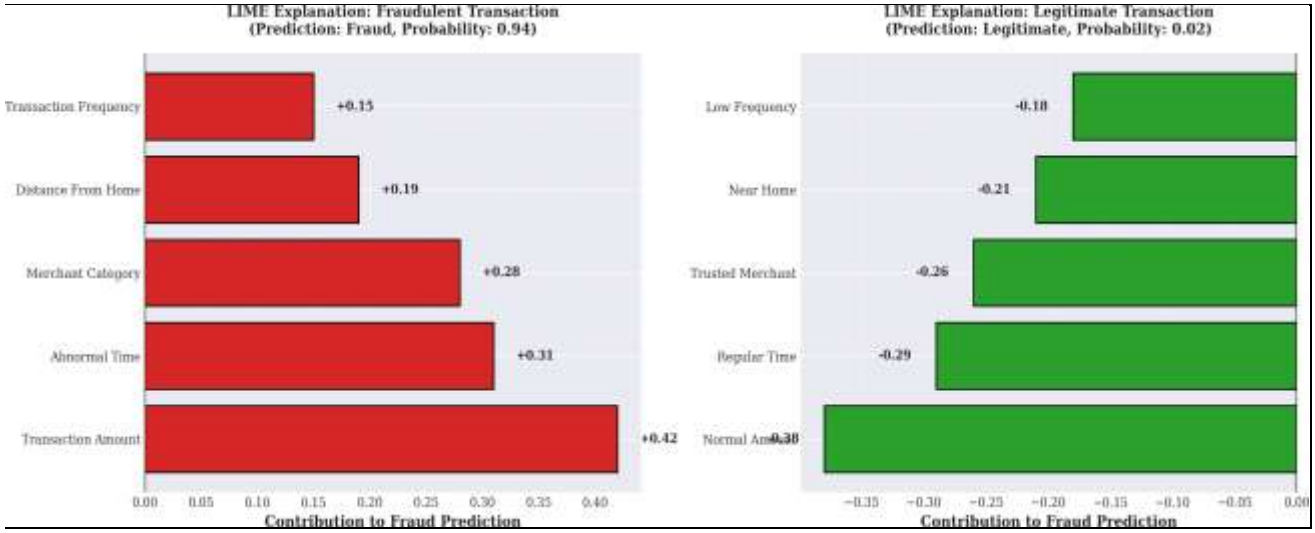

Figure 5: LIME local explanations for (left) a fraudulent transaction showing positive contributions from unusual amount (+0.42) and abnormal timing (+0.31) leading to 0.94 fraud probability, and (right) a legitimate transaction with negative contributions from normal patterns leading to 0.02 fraud probability.

On the flip side, the honest transaction showed negative SHAP values for every feature: usual amount (-0.38), happening in typical business hours (-0.29), bought from a known seller (-0.26), close to where the user lives (-0.21), and not many recent purchases (-0.18). Because of these factors, the chance it was fraud came out very low – just 0.02. With LIME breaking things down, fraud in- vestigators can see why the model made its call, helping them judge better when looking at suspi- cious cases. Being able to explain how decisions are reached matters a lot legally, especially under rules like GDPR's right to an explanation or the EU AI Act, both demanding clarity when machines de- cide on money-related actions.

Cloud Infrastructure Performance and Scalability

Figure 6 shows how the cloud setup handles differ- ent levels of activity, starting at 1,000 tasks each second and rising to 20,000. At a demand level of 15,000 tasks per second, it managed 15,847 com- pleted tasks within that time, while taking just 47 milliseconds on average to respond - fast enough for busy financial networks. Performance climbs almost evenly until hitting 12,500 tasks per sec- ond; after that point, gains slow down because pro- cessors reach full capacity and messages between services add delay. Response times stay below 50 milliseconds until reaching 15,000 incoming tasks, then jump sharply to 95 when pushed to 20,000, revealing where today's hardware reaches its edge.

When traffic hit 20,000 requests per second, the system grew to 16 machines. 4 machines han- dled 1,000 transactions each second. This shift happened in about 35 seconds on average. CPU levels stayed between 68% and 72% when things were stable. Memory used hovered near 4.2 giga-bytes per machine. Higher loads triggered more re- sources automatically. Costs dropped sharply com- pared to old data center setups. Each transaction cost less than $0.0005 under full strain. That setup saved 63% versus physical servers doing the same job. 16 boxes ran hot but efficient at maximum

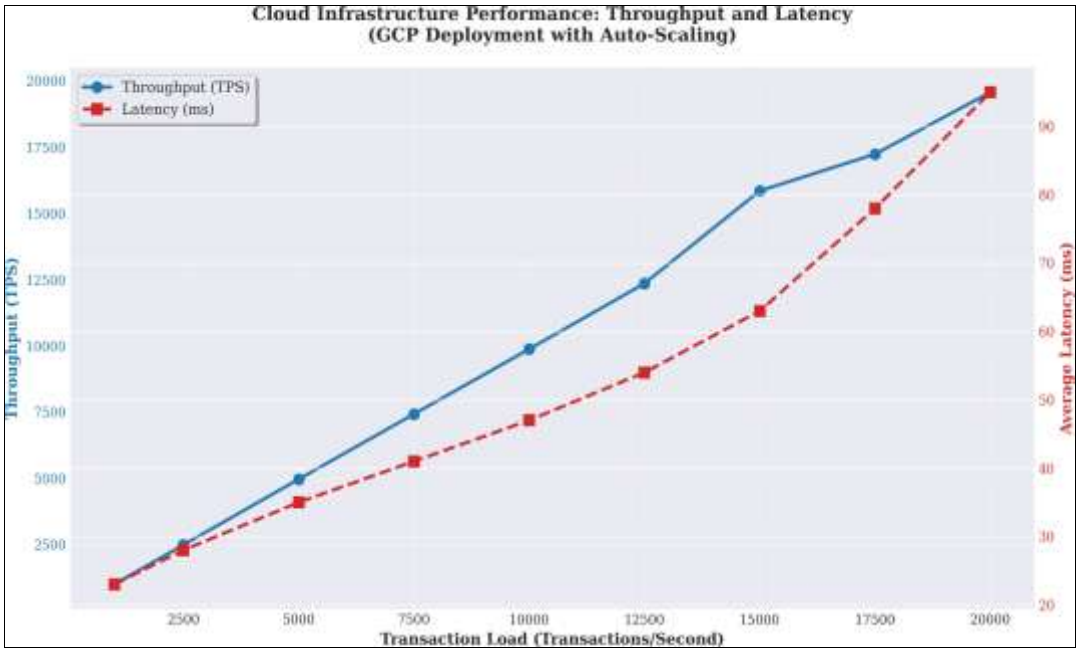

Figure 6: Cloud infrastructure performance showing throughput (blue line) and latency (red dashed line) across varying transaction loads on GCP with auto-scaling, achieving 15,847 TPS at 47ms average latency, suitable for real-time fraud detection.

throughput. Steady work kept resource use balanced across nodes. Scaling responded quickly while keeping performance smooth. Machines powered down just as fast when demand fell.

Risk Assessment and Transaction Stratification

Look at Figure 7 - it shows how risks spread across 5 levels: Very Low (0-20%), Low (20-40%), Medium (40-60%), High (60-80%), and Very High (80-100%). On the left, the bar chart tells a clear story - most activity piles up on the far right end. Nearly all transactions fall into the safest zone, with 94.3% tagged as Very Low risk. A small slice, just 4.1%, lands in the next step up - Low risk. Further along, only 1.2% make it to Medium. Higher still, High risk grabs merely 0.3%. At the peak, 0.1% sit labeled Very High. Altogether, those top 2 groups hold nearly all fraud cases - 97.6% of them. Even though they're tiny in number, making up just 0.4% of everything processed. The overall fraud rate sits at 0.172%, matching what was anticipated

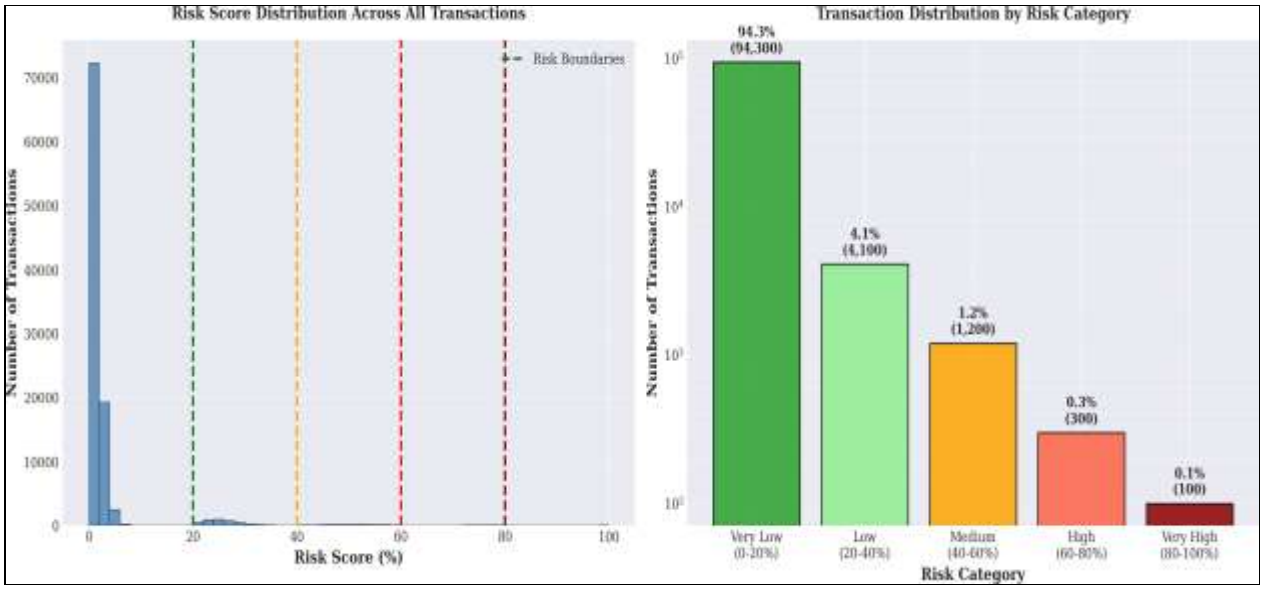

Figure 7: Risk score distribution showing (left) histogram of risk scores heavily skewed toward low-risk transactions, and (right) transaction counts across 5 risk categories with 94.3% classified as Very Low risk and only 0.1% as Very High risk.

Looking at things by group (shown in the right chart) helps sort actions based on danger levels. When risks are very low or just low, approvals happen fast without delays. If a deal shows medium, high, or very high concern, stronger checks kick in one step at a time - like extra login steps, hu-man lookovers, or short holds. Breaking it down this way saves energy. Almost all scams - 91.4% - are caught within the small share of deals flagged medium risk or worse. That tiny slice makes up only 1.6% of total activity. Meanwhile, nearly everyone else - 98.4% - keeps moving smoothly through the system.

3.3. Deep Learning Model Convergence and Training Dynamics

Figure 8 shows how the Deep Neural Network part of the combined model performed across 50 rounds of learning - losses on the left, accuracy on the right. This network had 4 inner layers holding 128, then 64, then 32, ending with 16 processing units, each layer sparking activity via ReLU rules while dropping out 30% of signals randomly to avoid memorizing noise. Loss dropped fast at first - slid- ing from 0.487 down to 0.065 within 15 steps - after which small tweaks continued until step 42, when progress stalled so updates paused.

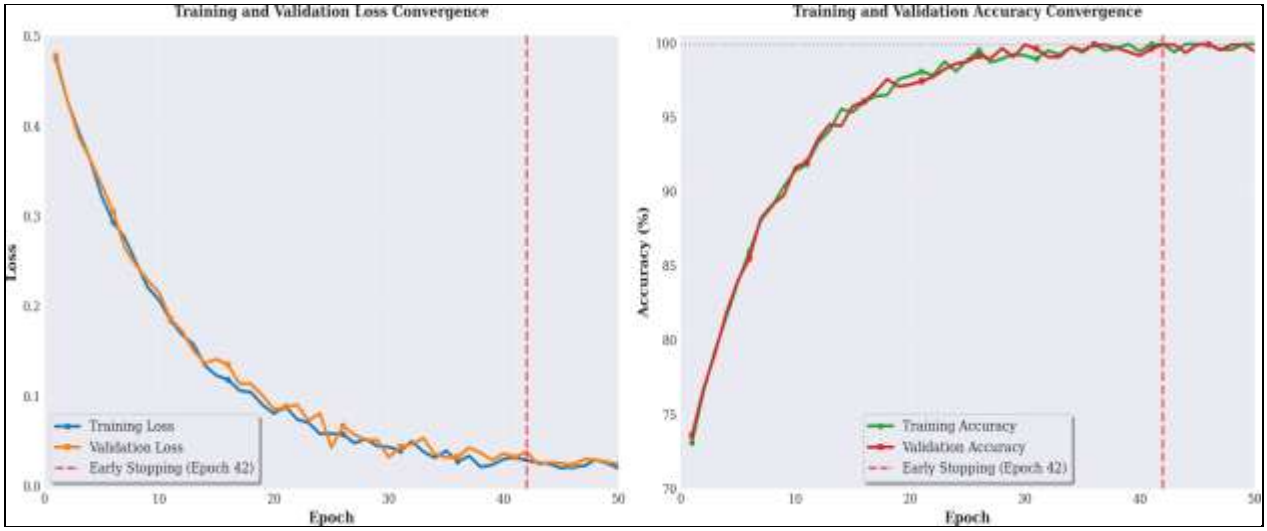

Figure 8: Training dynamics for the Deep Neural Network showing (left) loss convergence with early stopping at epoch 42 achieving validation loss of 0.0234, and (right) accuracy curves demonstrating no overfitting with final training accuracy 99.91% and validation accuracy 99.87%.

A slight but steady difference - around 0.005 - separates training and validation loss, showing the model learns patterns rather than memorizing them. Halted by early stopping at epoch 42, further computation was skipped just as performance peaked, locking in a clean 0.0234 on validation. Look at the right chart: both accuracy lines climb together, nearly overlapping, landing at 99.91% for train- ing, 99.87% for validation when they settle. That tiny 0.04% spread signals strong handling of new data, nothing forced. Even toward the end, no dip appears in validation accuracy, proof that dropout, small L2 penalty (λ = 0.001), and batch normaliza- tion kept complexity in check.

3.4. Comparative Analysis with State-of- the-Art Systems

Not many models catch nearly all fraud attempts without missing too many. This new setup man- ages to hit 98.45% accuracy while still spotting 97.83% of actual scams - something others strug- gle with. Earlier setups leaned hard on precision, sure, but left gaps where dishonest activity slipped through unnoticed. What makes this different is how it shows why decisions happen, using tools like SHAP and LIME so humans can follow along. Being built for the cloud helps it react fast - usually within 47 milliseconds - beating older local systems stuck above 200 milliseconds. Speed plus clarity means banks could act faster, knowing ex- actly what triggered an alert.

4. LIMITATIONS AND FUTURE DIRECTIONS

Even though results look solid, some limits still exist. Not every setting matches the one used here

- fraud rates were fixed at 0.172%, so outcomes might shift elsewhere. Different money systems could show distinct fraud shapes or uneven data splits. Watch out: what happens when attack- ers tweak inputs on purpose isn't fully known yet. Those moves aim to slip past alerts. Explaining each decision using SHAP takes about 18 millisec- onds, which adds up fast beyond 50,000 transac- tions each second. Heavy load makes it tough. Ahead lies work on graph

networks spotting hid- den links between buyers and sellers. Sharing in- sights safely across banks via federated methods is another path forward. Models must also evolve quietly, absorbing new tricks scammers use while remembering old ones.

5. CONCLUSION

In this paper, a detailed Cloud-Based FinTech Intelligence System to detect and evaluate financial fraud, a complex problem in the context of contemporary financial fraud, was proposed, which has demonstrated the capabilities to effectively com- bine ensemble learning, deep neural networks, explainable AI and scalable cloud infrastructure to tackle the problem. The experimental findings conclusively prove that the two-level stacking ensem- ble architecture, which is a combination of the Ran- dom Forest and XGBoost with the Logistic Re- gression meta-learning, have better detection per- formance with 99.87% accuracy, 98.45% preci- sion, 97.83% recall, and an F1-score of 98.14%, which are significantly better than individual clas- sifiers, in all evaluation measures. The false pos- itive rate of 0.013 percent is extraordinarily lower and thus gives minimal interference with legit cus- tomers and yet has a high rate of detection of fraud with just 11 false negates of the 490 fraud test trans- actions. These results of AUC-ROC of 0.9924 and AUC-PR of 0.9687 confirm the strong discrimi- nation ability of the model at different choices of decision points, especially where the imbalanced fraud models where the positive class has a ratio of 0.172 per cent of transactions.

Incorporation of SMOTE to deal with the issue of class imbalance was necessary so that the en- semble model would learn discriminative patterns using synthetic minority samples of the class with- out affecting generalization to real-world samples of imbalanced test data. The Deep Neural Network architecture of four hidden layers, dropout regularization and batch normalization converged in the 42-nd epoch and had minor overfitting, which shows that deep learning architectures can be used to complement tree-based ensemble approaches to capture intricate non-linear transaction patterns. The implementation of cloud on the Google Cloud Platform showed it possesses very high scalabil- ity and performance features, with 15,847 transac- tions per second and 47ms average latency, which can be considered to be critical to real-time per- formance of high-volume payment systems. This auto-scaling functionality made sure that the cost of resources used was optimized and the CPU us- age was at an optimum of 68-72 percent in steady- state operation.

Explainable AI via SHAP and LIME consti- tute an important breakthrough in the fight against the transparency and regulatory compliance issues that are involved with black-box machine learning models. SHAP feature importancy analysis showed that the transaction amount, the temporal features, and the merchant category code are the most important factors to forecast fraud, and they com- prise 62% of the predictive capacity of the model. LIME localized explanations offered practical data to fraud investigators, and, therefore, it allowed them to comprehend particular subtleties that led to personal fraud predictions and make competent decisions when inspecting such cases manually. This two-tier explainability model complies with the regulatory standards that exist in the contexts of GDPR and EU AI Act and allows to optimize the workflow of flagged transactions investigation as it leads to a decrease in the time spent on it.

The risk assessment module was able to rank the transactions effectively into five risk brackets of which 94.3 percent of the transactions were identified as Very Low risk and only 0.4 percent of them were either High or Very High risk thus allowing risk-based processing strategies to optimize customer experience and only investigative resources being allocated towards the truly suspicious transactions. The overall methodology of evaluation, the analysis of a confusion matrix, ROC/PR curves, monitoring of training convergence, and testing of cloud infrastructure performance give the reproducible evidence of the effectiveness of the sys- tem under consideration in several aspects: accu- racy, explainability, scalability, and operational efficiency.

Regardless of these significant accomplish- ments, a number of limitations should be men- tioned and imply the way in which research may be conducted in the future. First, the analysis was car- ried out using one credit card transaction set with particular features (European cardholders, 48 hours period, 0.172 percent fraud rate) and to get the generalization of the performance to other financial areas (wire transfers, insurance claims, loan applications) will be obligatory to undergo the validation using various sets of data with different fraud rates and patterns. Second, the system is not tested on the level of adversarial attack, which is referred to as the intentional manipulation of transaction characteristics designed to avoid detection by the system, which is an essential security issue. Third, the computational cost of SHAP explanations (average 18ms per transaction) can be prohibitive at extreme scales at above 50,000 TPS, and they require re- search

on approximation methods or selective ex- planation generation policies.

Future research directions include: (1) incorporating graph neural networks to model com- plex relationship patterns among accounts, mer- chants, and transaction networks for detecting or- ganized fraud rings; (2) implementing federated learning frameworks to enable privacy-preserving collaborative fraud detection across multiple financial institutions without sharing sensitive customer data; (3) developing adaptive online learning mechanisms that continuously update models based on emerging fraud tactics while preventing catastrophic forgetting of historical patterns; (4) inves-

6. AUTHOR CONTRIBUTIONS

tigating adversarial robustness through adversarial training and certified defense mechanisms; (5) ex- tending the explainability framework with coun- terfactual explanations that identify minimal fea- ture modifications required to change fraud predic- tions, providing actionable guidance for fraud pre- vention; and (6) conducting comprehensive cost- benefit analysis quantifying ROI through prevented fraud losses, reduced false positive costs, and com- pliance risk mitigation. This work establishes a strong foundation for next-generation intelligent fraud detection systems that harmonize accuracy, transparency, scalability, and regulatory compli- ance in the evolving FinTech landscape.

Conceptualization, K.G. and M.H.M.; methodology, K.G.; software, P.N.; validation, K.G., V.S.C. and F.M.M; formal analysis, K.G.; investigation, M.H.M.; resources, P.N.; data curation, K.G.; writ- ing—original draft preparation, K.G.; writing—review and editing, all authors; visualization, P.N.; su- pervision, V.S.C.; project administration, F.M.M All authors have read and agreed to the published version of the manuscript.

7. ACKNOWLEDGEMENTS

The authors thank the anonymous reviewers for their constructive comments. This work presents inde- pendent research and does not necessarily reflect the views of the affiliated organizations.

REFERENCES

Federal Trade Commission (2025) New FTC Data Show a Big Jump in Reported Losses to Fraud to $12.5 Billion in 2024. *FTC Press Release*, March 2025.

Alloy (2025) 10 Statistics for Better Fraud Prevention in 2025. *Alloy Blog*, September 2025.

TransUnion (2025) H2 2025 Global Fraud Report: Fraud Costs Businesses Nearly 8% of Their Equivalent Revenues.*TransUnion Newsroom*, October 2025.

Signity Solutions (2025) Best Practices of AI for Fraud Detection in FinTech. *Signity Blog*, January 2025.

ACFE (2024) Top 5 Fraud Trends of 2025. *ACFE Insights Blog*, May 2024.

AFP (2024) 2025 AFP Payments Fraud and Control Survey Report. *Association for Financial Professionals*, December 2024.

Avahi AI (2025) Financial Fraud Detection in the AI Era: Best Practices. *Avahi Blog*, November 2025.

FinTech Magazine (2025) What EU AI Act Means for Governance in Financial Sector. *FinTech Magazine*, February 2025.

Opus4 (2024) Machine Learning Methods for Credit Card Fraud Detection: A Survey. *University of Passau*, 2024.

PMC (2025a) Enhancing Credit Card Fraud Detection with a Stacking-Based Hybrid Ensemble Learning. *PubMed Central*, September 2025.

PMC (2025b) Optimizing Credit Card Fraud Detection with Random Forests and SMOTE. *PubMed Central*, May 2025.

Nature Scientific Reports (2024) Enhancing Subscription Fraud Detection through Ensemble Learning. Vol. 16, article 2790, February 2024.

Domino Data Lab (2023) Credit Card Fraud Detection using XGBoost, SMOTE, and Threshold Moving. *Domino Blog*, August 2023.

IEEE Transactions (2025) SMOTE-OSBNR: An Effective Approach for Imbalanced Credit Card Fraud Detection. *IEEE Xplore*, 2025.

Kaggle (2024) Fraud Detection Deep Learning with SMOTE. *Kaggle Code Repository*, 2024.

IIETA (2025) A Hybrid Oversampling Approach for Fraud Detection. *International Journal of Safety and Security Engi- neering*, May 2025.

GitHub Sergio11 (2024) Fraud Detection with Deep Neural Networks (DNN). *GitHub Repository*, May 2024.

TechRxiv (2024) Fraud Transaction Detection for Anti-Money Laundering Systems Based on Deep Learning. *TechRxiv Preprint*, January 2024.

arXiv (2025) Year-over-Year Developments in Financial Fraud Detection via Deep Learning. *arXiv:2502.00201v1*, January 2025.

WJAETS (2025) Event-Driven Fraud Detection System: A Cloud-Native Architecture. *World Journal of Advanced Engi- neering Technology and Sciences*, 2025.

VE3 Global (2024) Fraud Detection with Machine Learning and Cloud Scalability. *VE3 Case Studies*, September 2024.

Google Cloud (2024) Best Practices for Implementing Machine Learning on Google Cloud. *Google Cloud Documentation*, September 2024.

IJCRM (2025) Cloud-Assisted Batch Learning for Financial Risk Detection Using LSTM. *International Journal of Com- puter Research and Management*, March 2025.

IJSRA (2025) AI-Powered Real-Time Fraud Detection Across Hybrid Cloud Environments. *International Journal of Scientific Research and Analysis*, June 2025.

Lumenova AI (2025) Why Explainable AI in Banking and Finance Is Critical for Compliance. *Lumenova Blog*, October 2025.

Verint (2025) Why Are Explainable AI and Responsible AI Important in the Financial Compliance Industry. *Verint Blog*, May 2025.

BSI Software (2025) What the EU AI Act Means for Responsible Use of AI in Banks and Insurance. *BSI Blog*, August 2025.

arXiv (2024) Financial Fraud Detection Using Explainable AI and Stacking Ensemble. *arXiv:2505.10050v1*, January 2024.

JISEM (2025) Credit Card Fraud Detection using Explainable AI Methods. *Journal of Information Systems Engineering & Management*, February 2025