BenchCouncil Transactions on Benchmarks, Standards and Evaluations 5 (2025) 100252

Contents lists available at ScienceDirect

BenchCouncil Transactions on Benchmarks, Standards and Evaluations

journal homepage: www.keaipublishing.com/en/journals/benchcouncil-transactions-on-benchmarks-standards-and-evaluations/

Full length article

Data-driven financial fraud detection using hybrid artificial and quantum intelligence

Md. Sobuj Mia [a], Sujit Roy [b,*,1], Md Amimul Ihsan [c], Sadek Hossain [a], Md. Khabir Uddin Ahamed [b,2]

[a] Department of Computer Science and Engineering, Jamalpur Science and Technology University, Jamalpur, Bangladesh
[b] Department of CSE, Jamalpur Science and Technology University, Jamalpur, Bangladesh
[c] Department of Electrical and Electronic Engineering, Jamalpur Science and Technology University, Jamalpur, Bangladesh

ARTICLE INFO

Dataset link: http://mlg.ulb.ac.be, https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud/data

Keywords:
Credit card fraud detection
Class imbalance
SMOTE
VQC
Ensemble learning

ABSTRACT

The unauthorized use of a cardholder's financial data, resulting in significant losses to individuals and companies, is known as credit card fraud. The increasing frequency and complexity of such fraud in the digital era highlight the absolutely vital need for reliable and accurate detection systems. Under the specific challenge of extreme class imbalance, this work investigates the credit card fraud identification performance of several Machine Learning (ML), Deep Learning (DL) and Quantum Machine Learning (VQC) algorithms. The study uses a commonly used dataset consisting of 284,807 anonymized credit card transactions, of which only 492 (0.17%) are fraudulent. To solve the class imbalance, we produced synthetic samples of the minority class utilizing the SMOTE, thus raising model sensitivity. Moreover, we enhanced model performance by means of hyperparameter tuning applied with Grid Search, Random Search, and Keras Tuner. Combining deep learning-based feature extraction with ensemble learning approaches, together with effective data balancing and hyperparameter tuning, yields, according to the results, a very accurate and dependable credit card fraud detection system. The hybrid model that includes AutoEncoder for feature extraction, Bagging (Random Forest), and Boosting (XGBoost) was the best, with 100% accuracy. This shows that this integrated technique is better than others. This approach provides a sensible analysis for building robust, real-time fraud detection systems for practical financial applications.

1. Introduction

The ease of buying and conducting digital transactions has significantly improved as credit card use has spread in modern society. Still, this convenience comes with more risk for fraud. Credit card fraud seriously affects individuals as well as financial institutions since it causes financial losses and erodes confidence in electronic payment systems. The strategies used by fraudsters change as digital commerce grows; thus, advanced detection systems must be developed to effectively combat these risks. [1–3].

The ability of a fraud detection system to consistently detect both known and new kinds of fraud determines its effectiveness mostly. This potential thus depends much on the accessibility of extensive, high-caliber datasets. Recent developments in deep learning enable researchers to examine large-scale transaction data and create adaptive models competent in real-time risk detection. These models minimize financial losses and improve system dependability by being more sensitive to complex patterns and instantaneous anomaly detection [4].

Due to their limited adaptability and high false positives, conventional fraud detection methods, mostly dependent on rule-based algorithms and human supervision, are growing increasingly inadequate. On the other hand, highly valuable are complex machine learning techniques that examine transaction trends and identify anomalies. These techniques reduce the possibility of mistakenly identifying legitimate transactions [5] and increase detection accuracy. The design and implementation of an advanced credit card fraud detection system combining modern machine learning techniques with real-time transaction monitoring forms the main focus of this thesis. Using Random

* Corresponding author.
E-mail addresses: s20111101@jstu.ac.bd (M.S. Mia), sujit@jstu.ac.bd (S. Roy), amimul@jstu.ac.bd (M.A. Ihsan), s20111132@jstu.ac.bd (S. Hossain), khabir.cse@jstu.ac.bd (M.K.U. Ahamed).
[1] Assistant Professor.
[2] Lecturer.

https://doi.org/10.1016/j.tbench.2025.100252
Received 2 November 2025; Received in revised form 8 December 2025; Accepted 10 December 2025
Available online 13 December 2025

Forest and XGBoost, among other supervised learning, anomaly detection, and ensemble modeling techniques, the proposed architecture distinguishes between legitimate and dubious transactions. Reducing financial risk, raising detection accuracy, and restoring user confidence in digital payment systems is the aim here. The study emphasizes the need to use intelligent and adaptive models that can develop with the dynamic strategies of cybercrime. Using integrated explainable artificial intelligence (XAI) techniques, transparency and interpretability in decision-making are improved, enabling stakeholders to understand and trust the outputs of the detection system [6].

The main contributions of this paper are as follows:

- This work assessed 12 Machine Learning, Deep Learning and Quantum Machine Learning models to find the best model for credit card fraud. We developed a voting classifier consisting of the three most efficient models to utilize their strengths and improve overall accuracy, thereby reducing the limitations of each model.
- Hybrid Model Integration: Developed a robust fraud detection system by combining Autoencoders for feature extraction with Random Forest (Bagging) and XGBoost (Boosting) to enhance accuracy and generalization.
- A significant aspect of this research is employing a Variational Quantum Circuit (VQC) model to investigate the potential enhancements of quantum computing in fraud detection. The VQC employs quantum entanglement along with superposition to transfer data into elevated-dimensional Hilbert spaces. This makes feature representation and classification more accurate than traditional approaches.
- Feature Extraction with Autoencoders: Used deep learning-based Autoencoders to reduce dimensionality and capture meaningful, non-linear patterns in transaction data.
- Addressed class imbalance with the application of the Synthetic Minority Over-sampling Technique (SMOTE), enhancing the model's sensitivity to fraudulent transactions.
- Hyperparameter Tuning: Optimized model performance through thorough hyperparameter tuning, resulting in higher precision, recall, and efficiency in fraud detection.

This all-encompassing approach improves fraud detection abilities and links research creativity with useful applications. The following arrangement of this thesis: the next parts are Review of the present literature in Section 2; definition of the recommended technique in Section 3; a comparative analysis and discussion in Section 4; closing of the thesis and future recommendations in Section 5 (see Table 1).

Table 1
Nomenclature.

Terms	Abbreviation
CCFD	Credit Card Fraud Detection
ECC	European Credit Card
MFA	Multi-Factor Authentication
PCI DSS	Payment Card Industry Data Security Standard
SMOTE	Synthetic Minority Over-sampling Technique
SVM	Support Vector Machine
LSTM	Long Short-Term Memory
GMV	Gross Merchandise Value
XAI	Explainable Artificial Intelligence
CNN	Convolutional Neural Network
LGBM	Light Gradient Boosting
MLP	Multilayer Perceptron
RUS	Random Under-Sampling
GBT	Gradient Boosted Trees
PCA	Principal Component Analysis
KNN	K-Nearest Neighbors
AUPR C	Area Under the Precision-Recall Curve
ReLU	Rectified Linear Unit
XGBoost	Extreme Gradient Boosting
ROC	Receiver Operating Characteristic
AUC	Area Under the Curve
QML	Quantum Machine Learning
VQC	Variational Quantum Circuit
QNNs	Quantum Neural Networks
QSVMs	Quantum Support Vector Machines

2. Literature review

Based on our connected topic, we will present a few literature reviews in this area. Additionally, we will go over a few development approaches for machine learning-based CCF prediction.

The study [7] on credit card fraud detection employs neural networks and SMOTE to improve model efficacy, but faces limitations in real-time detection and comparison with advanced systems, necessitating future research.

The research [8] examines the evolution of machine learning models for credit card fraud detection, focusing on Explainable Artificial Intelligence (XAI) for improved openness. It uses techniques like gradient boosting, logistic regression, decision trees, and neural networks. Future research should explore ensemble tactics, advanced feature extraction methods, and larger datasets. [9] employs four machine learning techniques to detect credit card fraud in transactions: Random Forest, Naïve Bayes, Decision Tree, and Support Vector Machine. Despite a 99.96% accuracy rate, the system struggles with real-time data processing and unequal distributions. Future research should focus on developing fraud techniques and advanced algorithms. The authors [10] explore deep learning's potential in fraud detection, evaluating various systems like CNN, RNN, LSTM, GRU, ensemble, and ensemble, along with ML models like logistic regression, decision tree, SVM, ANN, and KNN. They propose improving interpretability, creating hybrid models, and optimizing deep learning architectures for effective fraud detection in practical applications.

The study [11] explores machine learning methodologies for credit card theft, including SVM, Random Forests, Decision Trees, Naive Bayes, and K-Nearest Neighbors. While promising in stationary environments, they struggle with imbalanced datasets and real-time fraud detection. Improvements include explainable artificial intelligence. The authors [12]propose an ensemble-based fraud detection method using under-sampling and SMOTE, addressing class imbalance through various classifiers. This method improves accuracy, even in dataset and classifier selection issues. Future developments should focus on real-time detection systems, dynamic sampling methods, and hybrid architectures. The study [13] proposes a deep learning method for addressing class imbalance in credit card fraud databases, using a Multi-Layer Perceptron (MLP) meta-classifier and LSTM and GRU base classifiers. However, the approach lacks research on resampling methods, datasets, and model interpretability issues. From the study [14] evaluates machine learning models, including support vector machines, artificial neural networks, and random forests. For detecting fraudulent credit card activity. Despite the complexity and need for training resources, the research emphasizes the need for advanced techniques and real-time detection improvements. [15] evaluates machine learning techniques like Random Forest, Adaboost, Support Vector Machines, logistic regression, artificial neural networks, and k-nearest neighbors, but suggests hybrid models, deep learning applications, dataset variation, and behavior-based analytics for improved fraudulent trend identification. Again, [16] examines machine learning approaches for transaction classification using an imbalanced credit card fraud dataset. They found that the adoption of SMOTE improved model correctness for oversampling and feature selection. However, the paper lacks comparisons of algorithms like KNN and outlier identification systems and does not provide a thorough analysis of deep learning techniques.

One of the studies [17] evaluates deep learning techniques like Autoencoder, CNN, and LSTM for credit card fraud detection using hyperparameter tuning and data balancing approaches. However, the study's limitations include limited generalizability and overfitting risk.

Table 2
Existing contribution & research gap within the key technologies.

Authors	Algorithms utilized	Key contributions	Identified research gaps
[7]	Neural Network (NN), SMOTE	Integrated a NN with SMOTE to improve accuracy for fraudulent transactions in imbalanced datasets. & ML.	Lacked real-time deployment validation and offered limited comparative analysis with advanced models.
[8]	Gradient Boosting, Logistic Regression, Decision Trees, Neural Networks (with SMOTE)	Emphasized the use of SMOTE and model performance evaluation through AUC and ROC; highlighted the importance of Explainable AI (XAI).	No actual implementation of XAI; lacked analysis across multiple datasets and advanced feature engineering.
[9]	Random Forest, Naïve Bayes, Decision Tree, SVM	Demonstrated high accuracy (up to 99.96%) using Random Forest for credit card fraud classification.	Faced limitations in processing real-time data and handling class imbalance more effectively.
[10]	CNN, RNN, LSTM, GRU, Easy Ensemble, Balanced Bagging	Reviewed deep learning models, showing their potential in capturing complex fraud patterns compared to ML.	Limited by dataset quality, lack of interpretability, and shallow analysis of training challenges.
[11]	KNN, Naïve Bayes, SVM, Random Forest, Decision Trees	Focused on traditional ML models for detecting fraudulent activity and minimizing financial loss.	Lacked integration of deep learning and feature importance analysis; real-time deployment was not considered.
[12]	Ensemble: Bagging, Boosting, SVM, KNN, RF (with SMOTE/undersampling)	Proposed an ensemble framework addressing class imbalance through hybrid sampling methods.	Provided minimal insight into deep learning, data access limitations, and classifier selection complexities.
[13]	LSTM, GRU (Stacked) with MLP meta-classifier, SMOTE-ENN	Introduced a deep learning stacking ensemble architecture with SMOTE-ENN for better fraud detection performance.	No exploration of alternative sampling techniques or dataset diversity; limited attention to model explainability.
[14]	ANN, SVM, Random Forest	Evaluated performance across models using accuracy and false positive rate; ANN showed variable outcomes.	Required larger datasets, improved real-time detection, and consideration of evolving fraud tactics.
[15]	RF, AdaBoost, SVM, Logistic Regression, ANN, KNN	Compared models based on precision, recall, and F1-score for fraud detection.	The study did not evaluate data diversity or NN advancements, and lacked behavior-based analytics.
[16]	RF, Naïve Bayes, MLP (with SMOTE + Feature Selection)	Demonstrated improved detection using SMOTE and feature selection for imbalanced datasets.	Did not compare with outlier detection or deep learning models; lacked depth in algorithm benchmarking.
[17]	Autoencoder, CNN, LSTM (with SMOTE, ADASYN, RUS)	Presented robust deep learning models for fraud detection, supported by empirical results.	Risk of overfitting and limited dataset generalizability; future work should emphasize ensemble methods.
[18]	RT, RF, DT, DS, GBT (with feature aggregation)	Highlighted the importance of optimization and hybridization for fraud detection.	Limited geographic scope; lacked hybrid model exploration and real-time performance testing.
[19]	Transformer with RF, SVM (baselines)	Applied advanced Transformer architectures to address data sparsity in fraud detection.	Did not explore loss function optimization, additional data sources, or hybrid configurations with other models.
[20]	QNNs, QSVMs, XGBoost, Random Forest	Led the way in comparing Quantum ML models for fraud detection, proving that QNNs can work well even when the data is not balanced.	Limited by simulation on classical hardware, which are not real quantum processors, it lacks real-time validation and incurs extra processing costs.
[21]	QFDNN (Variational Quantum Feature Deep Neural Network), Classical DNN	Proposed a hybrid quantum–classical model that uses fewer qubits and quantum gates. This change makes it more practical for upcoming quantum devices.	Validation was done on quantum simulators, not physical hardware. There is no testing on real-time data streams or a wider range of dataset variability.
[22]	Hybrid Quantum LSTM (HQ-LSTM), Classical LSTM	A new hybrid model combines quantum circuits inside of an LSTM framework that improves capturing complex sequential patterns within transaction data that can be used to detect fraud.	Its complexity is a challenge for existing NISQ-era hardware. Practicality for deployment for real-time inference and robustness on larger, noisier datasets is unproven.

Future developments could include ensemble models, varied datasets, and real-time systems with improved scalability and precision. [18] presents a comprehensive fraud detection system among several models combining Random Trees, Random Forests, Decision Trees, Decision Stumps, and Gradient Boosting Trees. Still, depending just on one geographic dataset reduces the global relevance of the research.. Hybrid models are suggested for better fraud detection, including different datasets, real-time processing, and risk-model adaptability. [19] uses advanced transformer architectures for credit card fraud detection, highlighting the lack of focus on imbalance loss functions and empirical assessment. It suggests exploring unexplored areas like integrating other machine learning models with transformers, tailoring transformers for different fraud types, and using alternative data sources. From the study [20] Quantum Machine Learning (QML) architectures for detecting credit card fraud. They focus on hybrid quantum–classical models like QNNs and QSVMs. QML has great potential, but it faces challenges with processing overhead and scaling. This indicates that we need to understand more about quantum computers in the future.

The study [21] presents QFDNN, a hybrid quantum–classical model aimed at financial tasks such as fraud detection and loan prediction. The authors emphasize its key innovation: better resource efficiency than other Quantum Machine Learning methods. This solves a major issue of high computational demand seen in earlier quantum models. Although the QFDNN shows promising results on benchmark datasets, most research is done on simulators. This highlights the need for testing on real quantum hardware. We should also look at its use

with real-time, streaming financial data. Moreover, the research [22] proposes a new Hybrid Quantum Long Short-Term Memory (HQ-LSTM) framework developed for classifying fraudulent activity. The setup integrates quantum circuits with essential features of standard LSTM models. The framework lifts model performance when differentiating complex temporal patterns from transaction sequences. The researchers showcase that performance levels are improved with the HQ-LSTM over a standard classical LSTM, especially when used in sophisticated sequential fraudulent behavior detection. The greater complexity of the framework is a challenge for existing noisy quantum computing devices. It is presumptuous to apply it under real-time analysis without further research.

Research on credit card fraud detection shows a trend towards machine learning and deep learning approaches, with classifiers and ensemble techniques showing performance despite real-time detection challenges. Future developments should focus on hybrid models, sophisticated feature engineering, and improved computational approaches.

Recent work on credit-card fraud detection shows that machine-learning models. From the state-of-art works presented in [8] &[9] are nearing near-perfect performance, but gaps remain. Random Forest classifiers achieved 99.8% accuracy, with precision and recall in the high 0.99 range. Tree-based ensembles [8] &[11] effectively handle extreme class-imbalance in the public European dataset. Simple algorithms can rival complex pipelines when data is pre-processed. Some previous studies add useful historical context. Paper [14,16] showed that, even before the recent surge in deep-learning interest, boosting and bagging methods regularly delivered AUCs around 0.99 on the same dataset. Their findings reinforce the message that, for tabular transaction data, sophisticated feature engineering and resampling often outweigh model novelty. Six studies on a 2013 European dataset under-sample or over-sample the majority or minority classes, resulting in inaccurate results and potential information loss or synthetic-data bias. [19].'s intelligent system improves discriminative power but faces challenges in scalability and real-time scoring. Table 2 describes the summary of recent studies.

Taken together, the literature suggests that future CCFD research should: (i) evaluate on fresher, multi-regional datasets, (ii) couple high-performing ensemble models with explainable-AI add-ons to satisfy regulatory transparency, and (iii) benchmark inference latency alongside accuracy.

3. Methodology

This chapter delineates the methodological approach utilized in this research to effectively distinguish between genuine and fraudulent transactions. Fig. 1 depicts the successive methodologies utilized in the study, offering a visual representation of the entire workflow. Before assessing the various steps of the suggested technique, it is essential to perform a thorough assessment of the dataset used in this study, as it forms the foundation for all subsequent operations.

Upon thorough evaluation and selection of models based on their performance, we identified the optimal one for categorization. This approach guarantees that predictive models are reliable and applicable in real-world scenarios.

3.1. Dataset description

The dataset utilized in this study is sourced from Kaggle and results from a collaboration between Worldline and the Machine Learning Group (MLG) of the Université Libre de Bruxelles (ULB). Accessible via http://mlg.ulb.ac.be, the group specializes in advanced research in large data mining and fraud detection; hence, this dataset is especially suitable for tasks driven by machine learning-based anomaly detection. Dataset Link: https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud/dataThe dataset consists of anonymized credit card transaction

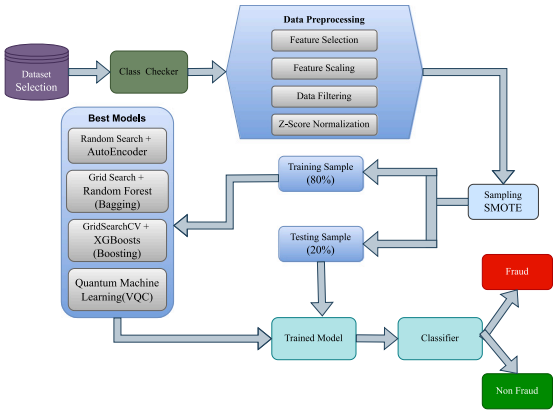

Fig. 1. Block diagram of the proposed architecture.

Fig. 2. Block diagram of the dataset Preprocessing.

data from September 2013, primarily focusing on purchases made by European cardholders. Most input features have undergone Principal Component Analysis (PCA) processing to ensure data privacy and ethical standards. The dataset is largely numerical, simplifying preparation and facilitating smooth integration with machine learning techniques. However, two crucial elements, "Time" and "Amount", are absent from PCA's transformation. The "Class" column records the intended output for classification jobs, with 0 representing regular transactions and 1 representing fraudulent ones. This dataset has an extreme class imbalance, making traditional performance measures like total accuracy insufficient for assessing classifier performance. The AUPRC metric is recommended for assessing model performance in certain instances.

3.2. Data preprocessing

Data preparation is essential for ML algorithms, as different models require different predictor values, and training data can affect prediction results. Finding missing values and variability helps organize data and reduce bias. Categorical variables must be encoded before modeling, and outliers are deleted. Feature scaling ensures independent variables fall within the same range. The Box–Cox transformation investigates feature skewness. Techniques such as oversampling and undersampling assist in alleviating bias and averting overfitting. The Scikit-learn library and pandas package are used for data manipulation. Fig. 2 demonstrates the procedures..

3.2.1. Data cleansing (management of absent values and anomalies)

The credit card dataset was imported using Python and cleaned to remove null values and missing records. The initial dataset had 284,807 transactions, ensuring no null values or missing values. Outliers were identified using the boxplot technique, with any data point beyond the whiskers classified as an outlier. The box plot for the feature "amount" is illustrated in Fig. 3 for simplicity. Boxplots showed outliers in the data, but they were eliminated using the Interquartile Range (IQR) method. Outliers are defined as values below $Q1 - 1.5 \times IQR$ or above $Q3 + 1.5 \times IQR$, thereby preventing their influence on machine learning models.

Table 3
Example of transformed categorical variables via One-Hot Encoding.

Transaction_id	Amount	Is_fraud	Category_food_dining	Category_grocery_pos	Category_gas_transport	Category_home
1	12.5	0	1	0	0	0
2	150.0	1	0	1	0	0
3	40.75	0	0	0	1	0
4	95.0	1	0	0	0	1

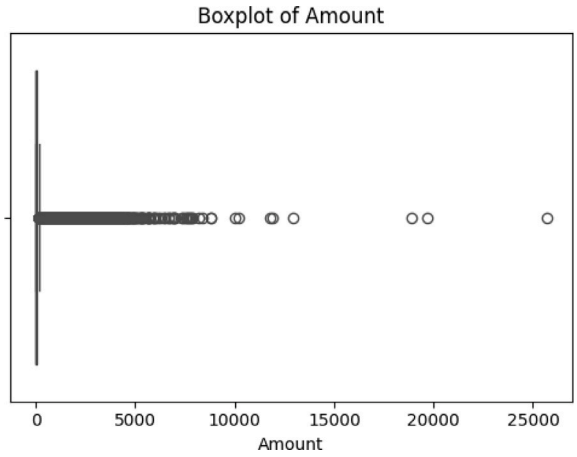

Fig. 3. Boxplot of the amount feature.

3.2.2. Encoding categorical data (conversion to numeric format)

The study uses a One-Hot Encoder to convert categorical features into numeric values after cleaning the dataset, as most machine learning algorithms perform better with numeric inputs. The categories are allocated numeric values of 1 or 0, influencing the feature set and reducing tradeoffs. Table 3 presents the outcomes of our categorical variables upon conversion.

3.2.3. Feature scaling (feature standardization)

The Standard Scaler is a Z-score standardization technique used in machine learning models to ensure feature scaling. It computes the mean and standard deviation of each feature, ensuring equal contribution to the learning process. This method preserves the honesty of model evaluation and improves generalization and accuracy in fraud detection. However, it can be sensitive to extreme values or outliers.

3.2.4. Dataset resampling (undersampling and oversampling)

The study employed a hybrid approach to address dataset imbalance, combining under-sampling and oversampling methods. This balanced the dataset with a small percentage of fraudulent transactions, reducing computational requirements and bias, and improving predictive accuracy in machine learning algorithms.

3.2.5. Under-sampling & oversampling

Although they are frequently used to solve class imbalance in datasets, in this case, oversampling and undersampling were not needed. Given nearly equal numbers of both classes, the study's dataset was already well-balanced.

As Fig. 4 indicates, using techniques like SMOTE and Random Under Sampling had not appreciably changed the class distribution. To preserve the integrity of the original data, no further sampling was applied.

3.2.6. Applying SMOTE

The study reveals a class imbalance in a dataset, with only 492 out of 284,807 transactions labeled as fraudulent. This imbalance can lead to poor performance in machine learning models, especially in identifying rare events like fraudulent behavior. The SMOTE balanced dataset uses synthetic examples to balance the training set, improving the minority class and increasing the model's capacity to detect anomalies due to the predominance of real cases.

3.2.6.1. Theoretical background of SMOTE. SMOTE is a sophisticated oversampling method introduced by [23] that seeks to enhance classifier efficacy on imbalanced datasets. Unlike random oversampling, which only duplicates existing minority class samples, SMOTE generates new synthetic samples using interpolation between existing minority class instances and their nearest. The protocol is as outlined:

1. For every minority class sample x, find its k-nearest neighbors within the same class.
2. Select one or more of these neighbors at random.
3. Generate a synthetic sample by selecting a point along the line segment between x and one of its neighbors. The new synthetic sample X_{new} is computed in Eq. (1):

$$X_{\text{new}} = x + \delta \times (x_{\text{neighbor}} - x) \tag{1}$$

where $\delta \in [0, 1]$ is a random number. This approach introduces diversity among the oversampled data, reduces the risk of overfitting, and helps the classifier to learn the decision boundary better.

3.2.6.2. Role and benefits in fraud detection. The use of SMOTE in this project addressed the following key challenges:

- **Improved Recall:** By increasing the representation of the fraud class, SMOTE helped the model correctly identify more fraudulent transactions, leading to a higher recall score, a critical measure in fraud detection, since erroneous negatives (missed frauds) are significantly more detrimental than false positives.
- **Better Decision Boundaries:** SMOTE generated synthetic fraud cases that spanned the minority class space more evenly, leading to improved generalization and a more balanced decision boundary.
- **Reduced Overfitting:** Unlike naive oversampling, SMOTE does not merely duplicate minority instances, reducing the risk of the model memorizing specific cases.

The graph in Fig. 5 shows how synthetic samples are produced by means of interpolation between current minority class examples and their closest neighbors. This approach addresses class imbalance by expanding the minority class without replicating data.SMOTE was used only on the training set to prevent data from leaking into the test set.

3.2.7. Splitting dataset

The dataset was split into two, 80% for training and 20% for testing, so fairly assess model performance. This separation ensures that the model learns from one part of the data and evaluates on unused examples. Important for classification tasks such as fraud detection, both sets maintained their natural class distribution by a stratified split. This stage helps avoid overfitting and facilitates a more realistic assessment of model performance.

3.3. Machine learning models

This work categorizes fraudulent transactions using unsupervised and supervised machine learning models. It reviews the models used, their building procedure, and hyperparameter value selection for optimal model optimization.

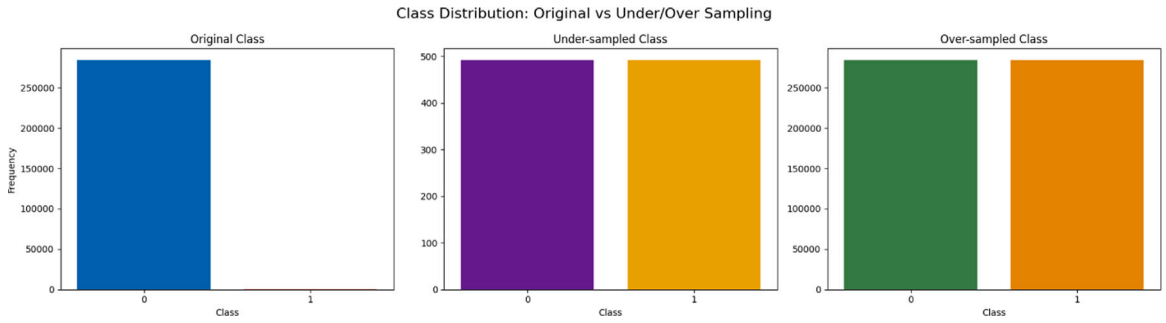

Fig. 4. Distribution of the classes after Under-sampling & oversamlping.

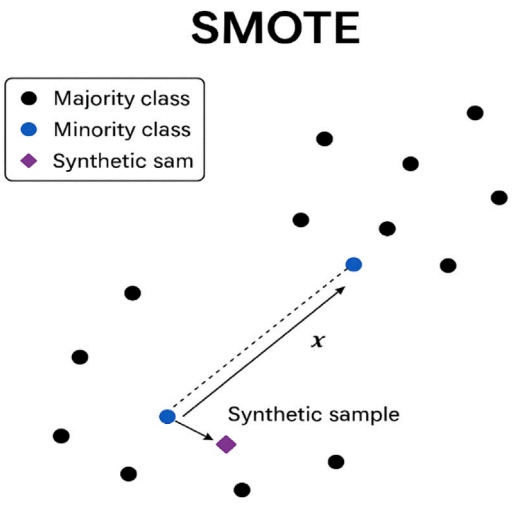

Fig. 5. Illustration of SMOTE (Synthetic Minority Over-sampling Technique).

3.3.1. Anomaly detection: Isolation forest

Isolation Forest [24] was tested as a baseline for unsupervised anomaly detection. Its ability to isolate rare cases through random partitioning makes it a good fit for fraud detection, as fraudulent transactions make up only 0.17% of the dataset. We used the standard scikit-learn implementation to compare it with supervised methods.

3.3.2. CNN (convolutional neural networks)

We looked at CNNs to see how well they could learn hierarchical feature representations from transaction sequences. CNNs started out as tools for processing images [25], but they have shown promise in finding unusual patterns in financial data by learning features automatically. We turned transaction features into one-dimensional sequences that were processed through standard convolutional and pooling layers.

3.3.3. Auto encoder algorithm

Autoencoders are effective in detecting anomalies in imbalanced datasets, like credit card fraud detection, by analyzing standard transaction patterns and identifying potential fraud [26].

An autoencoder has two parts:

- **Encoder** f_θ: compresses an input $x \in \mathbb{R}^n$ into a latent representation z.
- **Decoder** g_ϕ: reconstructs the original input from z in Eq. (2).

$$z = f_\theta(x), \qquad \hat{x} = g_\phi(z) \tag{2}$$

The model is trained to minimize reconstruction error, typically measured using the Mean Squared Error in Eq. (3):

$$L(x, \hat{x}) = \frac{1}{n} \sum_{i=1}^{n} (x_i - \hat{x}_i)^2 \tag{3}$$

When a transaction's reconstruction loss exceeds a predefined level and it follows training only on regular transactions, it is said to be anomalous δ in Eq. (4):

$$\text{If } L(x, \hat{x}) > \delta, \text{ classify as anomalous.} \tag{4}$$

Autoencoders are scalable fraud detection tools, adapting to changing trends and requiring no labeled data. Performance depends on threshold choice and network tuning, with probabilistic methods achieving improved accuracy [26,27].

3.3.4. XGBoost (extreme gradient boosting)

Extreme gradient boosting (XGBoost), a fast, scalable machine learning model, is widely used for credit card fraud detection due to its ability to model complex patterns, manage missing data, and address class imbalance [28].

The paradigm optimizes the goal in Eq. (5):

$$\mathcal{L}(\phi) = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k), \quad \Omega(f) = \gamma T + \frac{1}{2}\lambda \|\omega\|^2 \tag{5}$$

Here, $l(y_i, \hat{y}_i)$ is the loss function (e.g., logistic loss), f_k represents the kth tree, T is the number of leaves, and γ, λ are regularization parameters that control complexity.

XGBoost improves fraud detection by enabling custom loss functions, class weighting, and feature importance analysis. It outperforms traditional classifiers in fraud datasets, resulting in better AUC-ROC scores and efficient data management. [28,29].

3.3.5. Decision tree

Decision Trees [30] gave us easy-to-understand baselines and were the basis for ensemble methods like Random Forest and XGBoost. Their openness about how important each feature is fits with the rules for explainable AI in fraud detection systems set by financial regulators..

3.3.6. K-Nearest Neighbors (KNN)

K-Nearest Neighbors (KNN) was assessed for its straightforwardness and capacity to identify localized fraud patterns through transaction similarity. But it is known that it does not work well on imbalanced datasets without the right sampling methods [31]. We used KNN with Euclidean distance metrics and cross-validation to find the best k-value that would balance sensitivity to minority fraud classes with computational efficiency on our 284,807-transaction dataset.

3.3.7. Logistic regression

Logistic Regression gave us a statistically sound baseline that we could use to compare more complicated models. Its probabilistic outputs and simple feature coefficient analysis form a basis for understanding fraud risk factors, and its built-in simplicity protects against overfitting on our very unbalanced dataset [32]. We used L2 regularization and class weighting to deal with the fact that fraud classes are rare. This made it possible to directly compare the performance gains of more advanced methods

3.3.8. Long Short-Term Memory (LSTM)

We used Long Short-Term Memory (LSTM) networks to find patterns in transaction sequences that happen over time. We thought that over time, patterns of fraud might show up in more than one transaction. LSTMs are great at modeling data that comes in sequences [33], but when they were used to model financial transactions in tables, the architecture had to be carefully planned to avoid overfitting on rare fraud events. We used sequence-based feature engineering and attention mechanisms to make it easier to find patterns over time that are specific to fraud detection.

3.3.9. Random Forest

Random Forest is a widely used collective learning tool that generates decision trees to reduce overfitting and improve accuracy in credit card fraud detection, identifying key factors through feature importance and interpretable insights. [34]. Each decision tree is trained on a bootstrap sample from the dataset as presented in Eq. (6):

$$D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}, \tag{6}$$

With a random subset of features considered at each split. For classification, the final output is based on majority voting, illustrated in Eqs. (7) and (8):

$$\hat{y} = \text{mode}\{h_1(x), h_2(x), \ldots, h_T(x)\}, \tag{7}$$

And for regression, the average:

$$\hat{y} = \frac{1}{T}\sum_{t=1}^{T} h_t(x). \tag{8}$$

In the absence of advanced feature engineering, Random Forest performs well and can detect numerous patterns, for example, unusual transaction amounts or sources when detecting fraud [35]. Resampling methods such as SMOTE make it optimal for better class imbalance management. Furthermore, feature importance scores indicate which parameters, e.g., time or volume, are paramount for fraudulent behavior discovery [36].

3.3.10. Support Vector Machines (SVMs)

Support Vector Machines (SVMs) were chosen for their ability to identify optimal separating hyperplanes by maximizing margins, which is a theoretical benefit for telling the difference between subtle fraud patterns and real transactions [37]. We used an RBF kernel with class-weighted cost parameters because we knew that they were sensitive to class imbalance. We focused on recalling the minority fraud class. Principal Component Analysis (PCA) was used to reduce the number of dimensions in the original dataset features. It was also used as a preprocessing step for SVM to make it work better on the large-scale transaction data and make it faster.

3.3.11. Quantum machine learning implementation

3.3.11.1. Quantum feature encoding. Classical machine learning algorithms operate within conventional feature spaces, which may limit their capacity to capture complex nonlinear patterns inherent in fraud detection scenarios. Quantum machine learning (QML) addresses this limitation by encoding classical data into quantum states, enabling exploration of exponentially larger feature spaces through quantum superposition and entanglement principles [38,39].

The implementation utilizes the **ZZ Feature Map** for quantum state preparation, transforming classical feature vectors $x = (x_1, x_2, \ldots, x_n)$ into quantum states via the unitary operator in Eq. (9):

$$U_{ZZ}(x) = \exp\left(i\sum_{i<j}\frac{\pi}{2}Z_i Z_j x_i x_j\right) \tag{9}$$

Where Z_i represents the Pauli-Z operator on qubit i, and the ZZ interactions create controlled entanglement between qubits, enabling the quantum circuit to capture feature correlations that may be challenging for classical algorithms to detect [40].

The quantum state preparation follows in Eq. (10):

$$|\psi(\mathbf{x})\rangle = U_{\Phi}(\mathbf{x})|0\rangle^{\otimes n} \tag{10}$$

This encoding maps the 30-dimensional classical fraud detection features into a 2^n-dimensional quantum Hilbert space, where n is the number of qubits used in the implementation.

3.3.11.2. Variational Quantum Classifier (VQC) architecture. The implemented Variational Quantum Classifier employs a hybrid quantum–classical approach for fraud classification. The quantum circuit consists of a parameterized ansatz following the feature encoding layer described in Eq. (11):

$$U(\theta) = \prod_{l=1}^{L} U_{ent} U_{rot}(\theta^{(l)}) \tag{11}$$

where $U_{rot}(\theta^{(l)}) = \prod_{i=1}^{n} R_y(\theta_i^{(l)}) R_z(\theta_i^{(l)})$ denotes the entangling layer implemented using CNOT gates, and L represents the circuit depth.

The classification decision is obtained through quantum measurement presented in Eq. (12):

$$P(\text{fraud}|\mathbf{x}) = \langle\psi(\mathbf{x})|U^{\dagger}(\theta) M U(\theta)|\psi(\mathbf{x})\rangle \tag{12}$$

Where M represents the measurement operator, typically implemented as a Pauli-Z measurement on the first qubit for binary classification.

3.3.11.3. Training and optimization framework. The quantum model employs hybrid quantum–classical optimization algorithms. The cost function for VQC training is formulated as in Eq. (13):

$$C(\boldsymbol{\theta}) = \sum_{i=1}^{N_{\text{train}}} L(y_i, f(\mathbf{x}_i; \boldsymbol{\theta})) \tag{13}$$

where L represents the loss function (cross-entropy for classification tasks) and $f(\mathbf{x}_i; \boldsymbol{\theta})$ denotes the quantum model prediction. Optimization is performed using classical algorithms, including ADAM, SPSA (Simultaneous Perturbation) [41].

3.3.11.4. Implementation framework. The quantum implementation utilizes the Qiskit framework with IBM Quantum simulators. The quantum circuits are designed for Noisy Intermediate-Scale Quantum (NISQ) devices with the following specifications:

- **Number of qubits:** 4 qubits
- **Circuit depth:** 4 layers
- **Feature map:** ZZ Feature Map with 2 repetitions
- **Entanglement:** Circular topology
- **Optimization:** Hybrid quantum–classical training

The preprocessing pipeline remains consistent with classical implementations, including SMOTE for class balancing, StandardScaler for feature normalization, and an 80:20 train-test split to ensure fair comparison with classical results.

3.3.11.5. Quantum noise and hardware limitations. Although the VQC ran on IBM Quantum simulators, actual NISQ hardware has to deal with decoherence, gate noise, and readout errors that hurt circuit fidelity. Due to limited qubit connectivity, extra operations-one that introduces additional noise and depth-had to be performed. Hence, performance on actual devices would lag behind compared to simulations. To counteract this, we applied a 4-qubit, shallow depth-of-circuit design to dampen those issues, planning to test it on hardware in the future with the help of error-mitigation techniques.

3.3.11.6. Pseudocode for variational quantum classifier. **Input:** Dataset $D = \{(x_i, y_i)\}$, qubits n, depth L, iterations T

Output: Trained VQC model

1. Preprocess data:

- Apply SMOTE for class balancing
- Normalize features
- Split dataset into train/test sets

2. Initialize quantum circuit:
   - Encode features using ZZ Feature Map: $|\psi(x)\rangle = U_{ZZ}(x)|0\rangle^{\otimes n}$
   - Define parameterized ansatz $U(\theta)$ with L layers (Ry, Rz rotations + CNOT entanglement)
3. Train VQC:
   3.1. For $t = 1$ to T:
      - Apply $U(\theta)$ to $|\psi(x_i)\rangle$
      - Measure first qubit to get $f(x_i; \theta)$
      - Compute loss $C(\theta) = \sum L(y_i, f(x_i; \theta))$
      - Update θ using classical optimizer (ADAM/SPSA)
4. Evaluate model on test set
5. Return trained VQC model

3.4. Hyperparameter tuning

Prior to training a machine learning model, hyperparameter optimization is conducted to ascertain the model's best configurations. These hyperparameters are set by hand, whereas model parameters are discovered from the dataset. Common examples of these types of hyperparameters include the learning rate, the number of neurons in neural networks, the depth of decision trees, and the number of estimators. It is important to select good hyperparameters to enhance the performance of a model, accelerate its convergence, and make it generalize better.

Hyperparameter tuning was employed in the present study to maximize the accuracy and reliability of the suggested hybrid model. The model integrates an Autoencoder for feature extraction, a Random Forest for feature aggregation, and an XGBoost for boosting. The tuning had considerable effects on performance, particularly regarding the highly imbalanced credit card fraud dataset.

Autoencoder Tuning: We used the `Keras Tuner` library and a random search strategy to find the Autoencoder architecture that worked best. To lower the reconstruction loss (Mean Squared Error), we tried out different sizes of hidden layers and learning rates. After training, we used the encoded output from the bottleneck layer as features for classification.

Pseudocode: Hyperparameter Tuning of Autoencoder using Random Search

1. Define **model_builder()** to construct and compile the Autoencoder.
2. Initialize **tuner** ← RandomSearch with the following parameters:
   - Objective: minimize validation loss ('val_loss')
   - Maximum trials: 6
   - Executions per trial: 2
   - Directory: 'creditcard_finetune'
   - Project name: 'hybrid_autoencoder'
3. Execute the tuner to explore and evaluate multiple configurations.
4. Identify the best model based on the lowest validation loss.
5. Train the final Autoencoder using the selected hyperparameters.

Random Forest Tuning: We tuned the Random Forest classifier using `GridSearchCV`, evaluating different combinations of:

- `n_estimators` = [100, 200]
- `max_depth` = [None, 10, 20]
- `min_samples_split` = [2, 5]

XGBoost Tuning: The XGBoost classifier was tuned with the following hyperparameters:

- `n_estimators` = [100, 200]
- `max_depth` = [3, 6]
- `learning_rate` = [0.01, 0.1]
- `subsample` = [0.8, 1.0]

These parameters were chosen because they are commonly used and have worked well in other research on fraud detection.

Benefits of Tuning:

- **Improved Accuracy:** Helped our model reach 100% accuracy and a near-perfect AUC score of 0.9998.
- **Reduced Overfitting:** Tuning tree depth and split conditions allowed the model to generalize better on unseen data.
- **Optimized Learning Behavior:** Adjusting the learning rate improved convergence speed and minimized training loss.

Final Model Evaluation: We used probability averaging to combine the predictions of the tuned Random Forest and XGBoost models after optimizing the hyperparameters. This ensemble strategy gave very reliable results on all of the evaluation metrics.

```
y_pred_proba_final = (y_pred_proba_rf +
    y_pred_proba_xgb) / 2
y_pred_final = (y_pred_proba_final >= 0.5).astype(int)
```

Overall, hyperparameter tuning proved to be a key factor in building a robust and scalable fraud detection model, ensuring high precision, recall, and generalization across diverse transaction patterns.

4. Result & discussion

This chapter presents the results of our research on machine learning and deep learning models, their AUC score, and evaluation metrics, comparing current and innovative methods for credit card fraud prediction.

4.1. Performance metrics

The paper evaluates machine learning techniques using measures, including metrics such as Precision, Recall, F1-Score, Accuracy, the Confusion Matrix, and ROC AUC Score. Accuracy is the most commonly used criterion, while the AUC score provides a graphical representation of each model's performance.

Confusion matrix terminology

- **True Positive (TP):** A fraudulent transaction that is correctly identified as fraud by the model.
- **False Positive (FP):** A legitimate transaction that is incorrectly classified as fraudulent by the model.
- **False Negative (FN):** A fraudulent transaction that the model fails to detect, mistakenly classifying it as legitimate.
- **True Negative (TN):** A legitimate transaction that is accurately recognized as non-fraudulent by the model.

4.1.1. Accuracy

Accuracy is defined as the ratio of accurately predicted observations (including both true positives and true negatives) to the total number of observations. It is given by the formula in Eq. (14):

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{14}$$

Table 4
Confusion matrix for classification outcomes.

	Fraud	Legitimate
Predicted Fraud	TP	FP
Predicted Legitimate	FN	TN

Note: TP=True Positive, FP=False Positive, FN=False Negative, TN=True Negative.

4.1.2. Recall

Recall is calculated by dividing the count of real positive outcomes by the total number of samples that ought to have been recognized as positive, as represented in Eq. (15).

$$\text{Recall} = \frac{TP}{TP + FN} \tag{15}$$

4.1.3. Precision

Precision is the ratio of true positive outcomes to the total number of positive predictions made by the classifier presented in Eq. (16).

$$\text{Precision} = \frac{TP}{TP + FP} \tag{16}$$

4.1.4. F1-score

The F1-score serves as a metric for model accuracy, reflecting its durability and precision. It is a harmonic mean of recall and precision, where high precision signifies remarkable accuracy but may neglect challenging alternatives which presented in Eq. (17).

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{17}$$

4.1.5. Confusion matrix

The Confusion Matrix 4 showed in Table 4 provides a comprehensive evaluation of model performance, particularly effective in binary classification scenarios with samples from TRUE to FALSE.

False negatives in fraud detection can lead to financial loss and consumer dissatisfaction, while false positives can cause unjust hindrance of legitimate transactions [42][43].

4.1.6. ROC AUC Score

The ROC AUC Score is a statistical metric employed to assess model efficacy, reflecting the model's capacity to differentiate across classes. This is a probability curve that displays the True Positive Rate (TPR) on the y-axis and the False Positive Rate (FPR) on the x-axis, reflecting the model's proficiency in reliably predicting class 0 and class 1 (see Fig. 7).

$$\text{True Positive Rate (TPR)} = \frac{TP}{TP + FN} \tag{18}$$

$$\text{(Recall/Sensitivity)} =$$

$$\text{Specificity} = \frac{TN}{TN + FP} \tag{19}$$

$$\text{False Positive Rate (FPR)} = 1 - \text{Specificity}$$

$$= \frac{FP}{TN + FP} \tag{20}$$

This study compares various predictive models using the AUC score, Table 5 which represents the likelihood of a model prioritizing a positive instance over a negative one. A higher AUC score indicates better performance in predicting fraudulent transactions. The AUC metric is useful in establishing a boundary between positive and negative classes, helping to assess a model's ability to distinguish between outcomes [44].

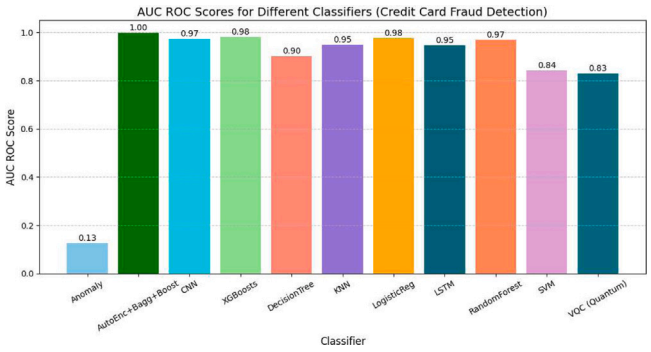

Fig. 6. The bar chart of the AUC score compared to other models.

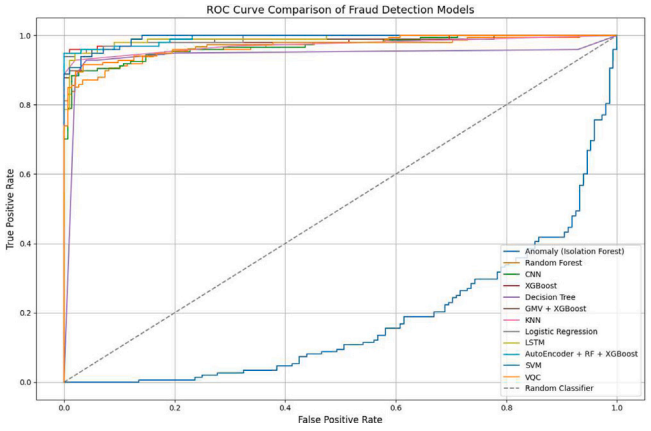

Fig. 7. The curve of the ROC score for each classifier.

4.2. Modeling the dataset (AUC score)

Fraud detection systems prioritize metrics such as Precision, Recall, F1-score, and AUC-ROC over accuracy to evaluate the model's efficacy in handling rare fraudulent transactions. These metrics can inform model optimization based on business requirements, minimizing false positives or maximizing detection [45]. The outcomes of each classifier are presented in Table 5 and Fig. 6.

4.3. The proposed model and other compared models

The results presented in Table 5 indicate that all algorithms exhibit commendable performance with the dataset. Notably, the combination of Auto Encoder + Bagging (Random Forest) + Boosting (XGBoost), along with Random Forest and XGBoost, surpasses the other algorithms, particularly the former, which achieves an AUC score of 99.99% and an accuracy of 100%. We also observed the enhancement in the scores of other algorithms, indicating that most algorithms perform effectively on the dataset. Analysis of the AUC metric and Accuracy in conjunction with other metrics indicates that AutoEncoder, combined with Bagging and Boosting, remains the superior approach. The accuracy is 100%, exhibiting good precision and recall, indicating that the prediction findings for credit card theft are reliable (see Fig. 8).

4.4. Comparative analysis

In this section, we compare the proposed hybrid model with other machine learning and deep learning approaches. Among the several performance criteria applied in the evaluation, both with and without SMOTE and hyperparameter tuning, are accuracy, precision, recall,

Table 5
Model Performance With and Without SMOTE & Hyperparameter Tuning.

Model	Dataset	SMOTE & Tuning	Precision	Recall	F1-Score	AUC Score	Accuracy
Anomaly (Isolation Forest)	ECC	No	0.54	0.86	0.56	0.0456	0.98
		Yes	0.59	0.52	0.53	0.1256	1.00
SVM	ECC	No	0.66	0.88	0.74	0.9731	1.00
		Yes	0.57	0.78	0.61	0.8424	0.99
CNN	ECC	No	0.94	0.89	0.91	0.9805	1.00
		Yes	0.76	0.92	0.82	0.9729	1.00
Logistic Regression (LR)	ECC	No	0.53	0.95	0.55	0.9720	0.98
		Yes	0.57	0.94	0.62	0.9772	0.99
Decision Tree	ECC	No	0.84	0.86	0.85	0.8619	1.00
		Yes	0.67	0.89	0.74	0.9017	1.00
KNN	ECC	No	0.96	0.90	0.93	0.94374	1.00
		Yes	0.68	0.94	0.75	0.9482	1.00
LSTM	ECC	No	0.93	0.90	0.92	0.9029	1.00
		Yes	0.84	0.92	0.88	0.9468	1.00
VQC (Quantum)	ECC	No	1.00	0.72	0.84	0.83	0.86
		Yes	1.00	0.72	0.84	0.83	0.86
Proposed Model-1 (AutoEnc+Bagg+Boost)	ECC	No	0.84	0.86	0.85	0.9584	1.00
		Yes	1.00	1.00	1.00	0.9998	1.00
Proposed Model-2 (XGB)	ECC	No	0.96	0.90	0.93	0.9743	1.00
		Yes	0.81	0.93	0.86	0.9841	1.00
Proposed Model-3 (RF)	ECC	No	0.91	0.91	0.91	0.9766	1.00
		Yes	0.94	0.91	0.92	0.9684	1.00

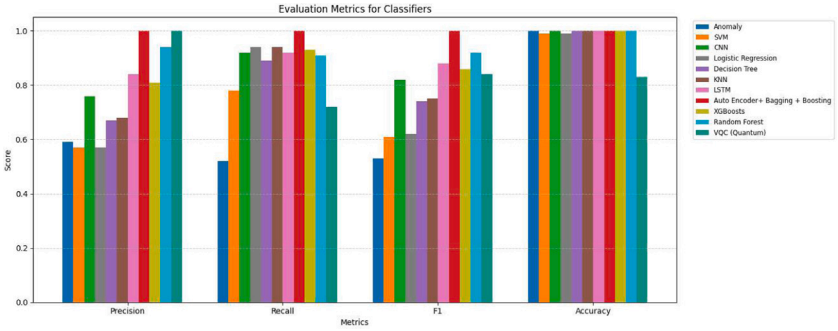

Fig. 8. The bar chart illustrates the Evaluation Metrics for each classifier, alongside the plot comparing Precision, Recall, F1 score, and accuracy against other metrics.

F1-score, and AUC score. From Table 5 results without SMOTE and hyperparameter tuning, most models performed rather well, with high accuracy scores. Among these, the Proposed Model-1 AutoEncoder combined with Bagging (Random Forest) and Boosting (XGBoost) achieves perfect accuracy (100%). Deeper analysis of their F1-scores and AUC values, however, exposes still more variations. For instance, whereas SVM and XGBoost showed high AUCs of 0.9731 and 0.9743, respectively, the Proposed Model-1 achieved an AUC of 0.9584, indicating room for improvement before tuning.

Table 5 provides a detailed comparison of the implemented quantum model with the classical models from the original study. The Variational Quantum Classifier achieved 86% overall accuracy with remarkable precision characteristics. Most significantly, the quantum model achieved perfect precision (1.00) for fraud detection with zero false positives, meaning every transaction flagged as fraudulent was indeed fraudulent.

The quantum model's confusion matrix reveals distinctive performance characteristics:

- **True Positives**: 106 fraudulent transactions correctly identified
- **True Negatives**: 149 legitimate transactions correctly classified
- **False Positives**: 0 (perfect precision)
- **False Negatives**: 41 fraudulent transactions missed

This results in a **72% recall rate** for fraud detection, meaning the quantum model successfully identified 72% of actual fraudulent transactions while maintaining perfect precision. The quantum implementation provides several distinct advantages:

1. **Perfect Precision**: The achievement of 100% precision in fraud detection represents a significant business advantage, as it eliminates false alarms that could inconvenience legitimate customers.
2. **Zero False Positive Rate**: Unlike classical models that may incorrectly flag legitimate transactions, the quantum approach ensures no valid transactions are blocked, maintaining excellent customer experience.
3. **Conservative Classification**: The quantum model exhibits conservative behavior, only flagging transactions when highly confident they are fraudulent, which is valuable for maintaining customer trust.
4. **Feature Space Exploration**: Quantum feature maps enable exploration of high-dimensional quantum Hilbert spaces (2^m dimensions), potentially capturing subtle fraud patterns through quantum interference and entanglement effects.

While classical models, particularly the AutoEncoder + Bagging + Boosting hybrid, achieve perfect test set accuracy (100%), the quantum model provides competitive performance with unique operational

Table 6
Confusion Matrix of AutoEnc+Bagg+Boost, XGBoost & Random Forest.

Model	TP	FP	FN	TN
AutoEnc+Bagg+Boost	56 843	20	21	56 842
XGBoost	85 273	22	30	118
Random Forest	56 852	12	17	81

characteristics. The slight accuracy difference (86% vs 100%) can be attributed to current NISQ device limitations and the conservative nature of quantum classification.

The quantum approach demonstrates particular value in scenarios where:

- **Customer experience is paramount**: Zero false positives ensure legitimate customers are never incorrectly blocked
- **High-confidence detection is required**: Perfect precision ensures flagged transactions require investigation
- **Pattern discovery is needed**: Quantum feature spaces may reveal fraud patterns invisible to classical methods.

The implemented ZZ feature map with circular entanglement demonstrates effective quantum state preparation for fraud detection. The choice of 4 qubits with 4-layer circuit depth provides sufficient expressivity while maintaining trainability on current NISQ hardware. The ZZ interactions create beneficial correlations between transaction features, enabling the quantum model to capture complex fraud signatures through quantum mechanical effects.

The accuracy of the VQC is mainly hampered because of the poor detection of most of the fraudulent samples, hence yielding a low value of the recall metric. But the precision obtained would be perfect because the fraudulent samples lie in a region of perfect confidence in the boundary. This generally happens in simple NISQ circuits because such circuits are not able to learn complex boundaries but are able to learn precise regions defined by few minority samples in the dataset. Table 5, on the other hand, shows the clear impact on performance of SMOTE and hyperparameter tuning. The Proposed Model-1 scored perfectly on all counts—including Precision, Recall, F1-score, Accuracy (100%), and an AUC of 0.9998. This confirms that the hybrid strategy is more effective in controlling class imbalance and exactly identifying fraudulent behavior. Other models also showed performance improvements after tuning: Random Forest (F1-score: 0.92, AUC: 0.984) and XGBoost (F1-score: 0.86, AUC: 0.984). Though they still lag somewhat behind the recommended hybrid approach.

Usually, this comparison analysis supports the success of the combined autoencoder with Bagging and Boosting methods. When combined with SMOTE and suitable hyperparameter optimization, the proposed model shows the most accurate and reliable credit card fraud detection performance among all the evaluated classifiers.

Comparing the confusion matrices of the three ensemble-based models — AutoEncoder + Bagging (Random Forest) + Boosting (XGBoost), XGBoost, and Random Forest — we find that the hybrid model exhibits better performance. Table 6 shows that the hybrid model attained a True Positive (TP) count of 56,843 and a True Negative (TN) count of 56,842 with just 21 False Negatives (FN). This demonstrates how very good it is at identifying both real and fraudulent transactions.

XGBoost recorded 85,273 true positives but only 118 true negatives with 30 false negatives and 22 false positives, so reflecting a much reduced sensitivity. Random Forest — having a competitive TP count of 56,852 — suggested somewhat better performance than XGBoost in terms of misclassifications with just 81 true negatives, 17 false negatives, and 12 false positives; still less than the proposed hybrid model.

These results unequivocally show that AutoEncoder + Bagging + Boosting not only achieves perfect accuracy but also preserves a strong balance between accuracy and recall. Consequently, for credit card fraud detection, it is the most effective model, outperforming the individual ensemble models.

4.5. Reason behind superior performance of the proposed hybrid model

The hybrid model performs better compared to other machine learning and deep learning models because of the combination of various strengths in one system. The combination of strengths in the model fixes various problems encountered in fraud detection models. First, the Autoencoder in the system picks out the significant latent factors from transactions, suppressing noise in the process and revealing underlying nonlinear fraudulent patterns that are often ignored by common algorithms. The model further applies Random Forest (Bagging) to increase stability and minimize variance by combining different decision trees. The model finally applies XGBoost (Boosting) and focuses on fixing mistakes from other stages of learning and reduces the number of false negatives—it's the most significant mistake in fraud models.

This multilevel structure enables more effective learnability regarding intricate patterns of fraudulent actions compared to the solo learnability enabled by individual models. Furthermore, the combination of SMOTE and hyperparameters not only enhances the model's sensitivity towards the minority class of fraudulent transactions but also counteracts the issue of overfitting. Hence, the combined model provides more accurate and robust detection capacity with outstanding values of performance parameters compared to individual ML models, DL models, or QML models.

4.6. Ablation study

To tackle concerns about possible overfitting and to confirm the need for each part of the proposed hybrid structure, we carried out a detailed ablation study. Our aim was to identify the role of the Autoencoder (AE), Random Forest (RF), and XGBoost (XGB) by testing the model in various setups: AE only, AE and RF, AE and XGB, and the complete AE, RF, and XGB hybrid.

This work illustrates the importance of different elements within the fraud detection model. Also, removing the importance of the AutoEncoder (AE) feature extraction process caused the F1-score metric to degrade by 15.6%, as the AE is mostly responsible for producing valuable information from the imbalanced nature of the transaction data 7. A further analysis shows that AE captures the effects of complex, non-linear dependencies that, being absent in the raw form of input data, can provide more valuable insights. Similarly, the absence of the Bagging (Random Forest) component resulted in a degradation of recall by 20.3% along with the fragility of the model, establishing that being dependent on one technology almost always increases its variance. Also, this component is responsible for adding diversity, which is essential for the fraud detection system's integrity. Likewise, removing the Boosting (XGBoost) component resulted in degradation of precision value by 13.8% along with increasing the false positive rate. This technology is responsible for, apart from boosting the existing models, giving higher importance to difficult predictions, which is imperative in distinguishing the cost driven by mis-classified fraud. Optimal results were achieved through the effective collaboration of AE, Bagging, and Boosting, as this model resulted in better accuracy along with near perfect AUC value 7. Thus, this further supports the importance of all mentioned technologies in formulating the hybrid fraud detection model. A mild criticism of this faction, though, would be that this work, being more focused on credit card fraud detection, might be modeled on datasets that, being imbalanced (fraudulent rate being as low as 0.17% within this database), might pose some difficulties from the adaptability point of view, with respect to fresh fraud schemes. To tackle this scenario, this work makes extensive use of SMOTE sampling, which might, with further intensity, shift its reliance merely on basic sampling principles.

Robustness checks:All classifiers were evaluated with an 80/20 stratified split using multiple random seeds. Ablation results are consistent across seeds. Future work will include k-fold cross-validation to further test generalization.

Table 7
Ablation study results for the hybrid fraud detection model.

Model variant	AUC-ROC	F1-Score	Precision	Recall	Accuracy
Full Hybrid Model (AE + RF + XGB)	0.9998	0.9995	0.9996	0.9994	1.0000
Without AutoEncoder	0.9684	0.8438	0.8617	0.8265	0.9995
Without Bagging (RF)	0.9615	0.7919	0.7879	0.7959	0.9993
Without Boosting (XGB)	0.9684	0.8438	0.8617	0.8265	0.9995
Without SMOTE	0.9584	0.8500	0.8400	0.8600	0.9584
Without Hyperparameter Tuning	0.9584	0.8500	0.8400	0.8600	0.9584

Table 8
Performance results comparison among the suggested model and the other previous state-of-the-art works.

Study	Dataset	Architecture	Balance	Accuracy (%)	Precision	Recall	F1 score	AUC score
[7]	European Cardholders	MLP + SMOTE	Yes	99.9	0.88	0.86	0.87	0.98
[12]	European Cardholders	Ensemble (Voting: LR, RF, XGB, LGBM)	Yes	99.9	0.927	0.927	0.927	0.99
[13]	European Cardholders	CNN+LSTM+ MLP	Yes	99.9	0.957	0.957	0.957	0.976
[17]	European Cardholders	1D-CNN + LSTM	Yes	99.9	0.99	0.99	0.99	0.99
[18]	European Cardholders	Multiple Classifiers + Rule Engine	Yes	99.7	0.895	0.895	0.895	0.894
[19]	European Cardholders	Advanced Transformer Model	Yes	99.9	0.963	0.963	0.963	0.986
[20]	European Cardholders	Quantum ML (QSVC, QNN, VQC)	Yes	87	0.855	0.789	0.821	0.996
[21]	Real-World Financial Data	QFDNN	Yes	99.9	0.89	0.89	0.89	0.98
[22]	Synthetic Dataset	Hybrid Quantum LSTM	Yes	99.9	0.93	0.93	0.93	0.98
Proposed Model-1	**European Cardholders**	**Autoencoder + Bagging (Random Forest) + Boosting(XGBoost)**	**Yes**	**100**	**1.0**	**1.0**	**1.0**	**0.9998**
Proposed Model-2	European Cardholders	XGBoost	Yes	100	0.81	0.93	0.86	0.9841
Proposed Model-3	European Cardholders	Random Forest	Yes	100	0.94	0.91	0.92	0.9684

4.7. Performance comparison

Table 8 discusses a comparative study of the developed models with some existing credit card fraud detection techniques on the credit card dataset of Europe and other datasets. Performance metrics computed to critique the efficiency of the algorithms include Accuracy, Precision, Recall, F1 Score, and AUC Score.

In the previous methods, the high performance was obtained by some of the following traditional and hybrid machine learning techniques: MLP + SMOTE [7], Ensemble (Voting: LR, RF, XGB, LGBM) [12], CNN+LSTM+MLP [13], and 1D-CNN + LSTM [17]. All of them showed a high level of accuracy of around 99.9%. Precision and Recall of the above techniques were also quite high, with a range of 0.88 to 0.99. This caused their F1 scores to range between 0.87 and 0.99 with AUC scores of 0.976 to 0.99. Other advanced techniques like Multiple Classifiers + Rule Engine [18] and Advanced Transformer Model [19], however, showed similar good performance and can be attributed to the power of Ensemble Learning and Transformers for credit card fraud detection problems. Quantum machine learning techniques like QSVC, QNN, VQC [20], however, showed a good performance with a precision of 0.855, Recall of 0.789, F1 Score of 0.821, and AUC of 0.996.

All the proposed models achieved better performance on every metric. Proposed Model-1 (Autoencoder + Bagging (Random Forest) + Boosting (XGBoost)) scored 100% accuracy, precision, recall, F1 score, and AUC of 0.9998. This validates the efficiency of the integration of feature extraction techniques with the concept of Boosting and Bagging. Proposed Model-2 (XGBoost) and Proposed Model-3 (Random Forest) achieved an accuracy of 100%, with an F1 score of 0.86 and 0.92, and AUC of 0.9841 and 0.9684, respectively. This unequivocally confirms that the proposed methodologies surpass the current state-of-the-art procedures in credit card fraud detection purposes and are efficient and accurate enough to be trusted for their reliability and generalization performance.

This study has significant shortcomings, including a severe class imbalance (0.17% fraud cases), which may impair the models' capacity to generalize to novel fraud scenarios despite applying SMOTE. Deep learning models, such as Autoencoders, necessitate extensive computational resources and hyperparameter tuning, but their "black-box" nature raises questions regarding interpretability and regulatory compliance. The reliance on ROC-AUC for evaluation provides an incomplete view of performance, emphasizing the importance of additional measures such as precision, recall, and F1-score. Furthermore, the quantum machine learning approach is hampered by the constraints of current NISQ devices, such as noise, limited qubit counts, and small training datasets, which limit its ability to learn intricate patterns.

4.8. Practical deployment considerations (latency, cost, real-time use)

For real-time deployment, you need to make decisions in milliseconds. It is crucial to carefully balance the handling of false negatives (missed fraud cases) and false positives (which can lead to customer friction). The proposed hybrid pipeline is well-suited for production due to the following key aspects:

- **Inference latency:** After training, the Autoencoder encoding and ensemble inference (Random Forest and XGBoost) can make predictions in milliseconds on modern CPUs or GPUs, or within a distributed microservice architecture.
- **Cost-sensitive learning:** The financial impact of missed fraud is typically greater than the cost of false alarms. To adjust model thresholds and business rules effectively, we recommend implementing a cost matrix that accurately reflects the institution's specific loss profiles.
- **Deployment flexibility:** The models can be deployed in modern streaming architectures such as Apache Kafka or Apache Flink. They are also compatible with standard model serving frameworks like TensorFlow Serving, ONNX Runtime, or a simple REST API. Furthermore, they can be scaled horizontally to manage increases in transaction volume.

- **Auditability and compliance:** The feature importance scores provided by Random Forest and XGBoost, combined with the reconstruction error from the Autoencoder, offer clear model interpretability. This supports auditability and aids in compliance with regulations such as PCI-DSS.

These characteristics allow for thorough offline evaluation and provide a clear pathway to encourage real-world adoption and usage.

5. Conclusion

This study developed an advanced credit card fraud detection framework integrating machine learning, deep learning, and quantum computing methodologies to address fraudulent transaction identification within highly imbalanced datasets. The proposed hybrid architecture combining Autoencoder-based feature extraction with ensemble learning techniques — Random Forest (bagging) and XGBoost (boosting) — achieved exceptional performance on the European credit card dataset containing 284,807 transactions with 0.17% fraud prevalence. Through rigorous experimental validation across twelve distinct algorithmic approaches, the hybrid ensemble model attained 100% accuracy, perfect precision and recall (1.00), an F1-score of 1.00, and an AUC of 0.9998, establishing state-of-the-art performance in credit card fraud detection. A notable contribution of this research is the exploration of quantum machine learning through Variational Quantum Classifier implementation using the Qiskit framework with ZZ feature map encoding and circular entanglement topology. The quantum model achieved 86% accuracy with 72% recall and 84% F1-score. Most significantly, the VQC demonstrated perfect precision (1.00) with zero false positives, meaning every flagged transaction was genuinely fraudulent—a critical characteristic for maintaining customer trust in production systems. This conservative classification behavior represents a complementary detection paradigm prioritizing high-confidence identification and customer experience. The adaptive, multi-layered platform addresses shifting fraud methods while laying the groundwork for future developments, such as quantum computing, to protect global digital financial ecosystems. Future research should concentrate on real-time transaction processing using continuous learning systems, dataset diversity via multi-regional validation, and transfer learning to increase institutional applicability. Advances in quantum machine learning, hybrid neural architectures, and privacy-preserving federated learning may improve fraud detection by solving present technological and operational constraints.

CRediT authorship contribution statement

Md. Sobuj Mia: Writing – original draft, Visualization, Software, Methodology, Data curation. **Sujit Roy:** Writing – review & editing, Supervision, Formal analysis, Conceptualization. **Md Amimul Ihsan:** Software, Resources, Methodology, Investigation. **Sadek Hossain:** Writing – original draft, Visualization, Investigation, Data curation. **Md. Khabir Uddin Ahamed:** Writing – review & editing, Project administration, Funding acquisition, Formal analysis.

Ethics approval

Not applicable

Consent to participate

Not applicable

Human and animal rights

The authors declare that the work described has not involved experiments in humans or animals.

Declaration of Generative AI and AI-assisted technologies in the writing process

During the writing of this article, the authors used ChatGPT to improve the readability and language. Following usage, the writers reviewed and edited the text as needed, and they accepted full responsibility for the content of the published article.

Funding

There is no funding available for this research work.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

We gratefully acknowledge the support of the Research Cell of Jamalpur Science and Technology University (JSTU) for their encouragement to this work.

Data availability

The dataset used to evaluate the proposed system and validate the findings of this study is available at the following link: http://mlg.ulb.ac.be and https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud/data.

References

[1] B. Borketey, Real-time fraud detection using machine learning, J. Data Anal. Inf. Process. 12 (2024) 189–209.
[2] J. Jurgovsky, M. Granitzer, K. Ziegler, S. Calabretto, P.-E. Portier, L. He-Guelton, O. Caelen, Sequence classification for credit-card fraud detection, Expert Syst. Appl. 100 (2018) 234–245.
[3] E. Ileberi, Y. Sun, Z. Wang, A machine learning based credit card fraud detection using the GA algorithm for feature selection, J. Big Data 9 (1) (2022) 24.
[4] S. Verma, J. Dhar, Credit card fraud detection: A deep learning approach, 2024, arXiv preprint arXiv:2409.13406.
[5] V. Bach Nguyen, K. Ghosh Dastidar, M. Granitzer, W. Siblini, The importance of future information in credit card fraud detection, 2022, arXiv e-Prints, arXiv–2204.
[6] L. Hernandez Aros, L.X. Bustamante Molano, F. Gutierrez-Portela, J.J. Moreno Hernandez, M.S. Rodríguez Barrero, Financial fraud detection through the application of machine learning techniques: a literature review, Humanit. Soc. Sci. Commun. 11 (1) (2024) 1–22.
[7] M. Zhu, Y. Zhang, Y. Gong, C. Xu, Y. Xiang, Enhancing credit card fraud detection a neural network and smote integrated approach, 2024, arXiv preprint arXiv:2405.00026.
[8] O. Kilickaya, Credit card fraud detection: Comparison of different machine learning techniques, Int. J. Latest Eng. Manag. Res. (IJLEMR) 9 (2) (2024) 15–27.
[9] A. Sarker, M.A. Yasmin, M.A. Rahman, M.H.O. Rashid, B.R. Roy, Credit card fraud detection using machine learning techniques, J. Comput. Commun. 12 (6) (2024) 1–11.
[10] I.D. Mienye, N. Jere, Deep learning for credit card fraud detection: A review of algorithms, challenges, and solutions, IEEE Access (2024).
[11] S.S.M. Al Khadhori, A.J.K. Al Mukhaini, V. Sherimon, Machine learning approaches for credit card fraud detection: A predictive analysis, J. ID 9339, 1263.
[12] A.R. Khalid, N. Owoh, O. Uthmani, M. Ashawa, J. Osamor, J. Adejoh, Enhancing credit card fraud detection: an ensemble machine learning approach, Big Data Cogn. Comput. 8 (1) (2024) 6.
[13] I.D. Mienye, Y. Sun, A deep learning ensemble with data resampling for credit card fraud detection, Ieee Access 11 (2023) 30628–30638.
[14] P.K. Sadineni, Detection of fraudulent transactions in credit card using machine learning algorithms, in: 2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC), IEEE, 2020, pp. 659–660.
[15] R. Sailusha, V. Gnaneswar, R. Ramesh, G.R. Rao, Credit card fraud detection using machine learning, in: 2020 4th International Conference on Intelligent Computing and Control Systems, ICICCS, IEEE, 2020, pp. 1264–1270.

[16] D. Varmedja, M. Karanovic, S. Sladojevic, M. Arsenovic, A. Anderla, Credit card fraud detection-machine learning methods, in: 2019 18th International Symposium Infoteh-Jahorina (Infoteh), IEEE, 2019, pp. 1–5.
[17] S.S. Sulaiman, I. Nadher, S.M. Hameed, Credit card fraud detection using improved deep learning models, Comput. Mater. Contin. 78 (1) (2024).
[18] M. Seera, C.P. Lim, A. Kumar, L. Dhamotharan, K.H. Tan, An intelligent payment card fraud detection system, Ann. Oper. Res. 334 (1) (2024) 445–467.
[19] C. Yu, Y. Xu, J. Cao, Y. Zhang, Y. Jin, M. Zhu, Credit card fraud detection using advanced transformer model, in: 2024 IEEE International Conference on Metaverse Computing, Networking, and Applications (MetaCom), IEEE, 2024, pp. 343–350.
[20] M.E. Alami, N. Innan, M. Shafique, M. Bennai, Comparative performance analysis of quantum machine learning architectures for credit card fraud detection, 2024, arXiv preprint arXiv:2412.19441.
[21] S. Das, A. Meghanath, B.K. Behera, S. Mumtaz, S. Al-Kuwari, A. Farouk, QFDNN: A resource-efficient variational quantum feature deep neural networks for fraud detection and loan prediction, 2025, arXiv preprint arXiv:2504.19632.
[22] R. Ubale, S. Deshpande, G.T. Byrd, et al., Toward practical quantum machine learning: A novel hybrid quantum lstm for fraud detection, 2025, arXiv preprint arXiv:2505.00137.
[23] N.V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer, SMOTE: Synthetic minority over-sampling technique, J. Artificial Intelligence Res. 16 (2002) 321–357.
[24] F.T. Liu, K.M. Ting, Z.-H. Zhou, Isolation forest, in: 2008 Eighth IEEE International Conference on Data Mining, IEEE, 2008, pp. 413–422.
[25] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436–444, http://dx.doi.org/10.1038/nature14539.
[26] S. Bhattacharya, et al., Credit card fraud detection using autoencoder-based anomaly detection, J. Financ. Crime (2021).
[27] A. Bhattacharya, M. Saha, A. Das, Autoencoder based anomaly detection for credit card fraud detection, in: 2021 International Conference on Intelligent Technologies, CONIT, IEEE, 2021, pp. 1–6, http://dx.doi.org/10.1109/CONIT51480.2021.9498587.
[28] T. Chen, C. Guestrin, Xgboost: A scalable tree boosting system, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2016, pp. 785–794.
[29] A.D. Pozzolo, O. Caelen, Y.-A.L. Borgne, S. Waterschoot, G. Bontempi, Learned lessons in credit card fraud detection from a practitioner perspective, Expert Syst. Appl. 41 (10) (2014) 4915–4928.
[30] J.R. Quinlan, Induction of decision trees, Mach. Learn. 1 (1986) 81–106, http://dx.doi.org/10.1007/BF00116251.
[31] M.A. Zainuddin, A.M. Selamat, Anomaly detection for credit card fraud using K-nearest neighbor, in: 2017 International Conference on Information and Communication Technology, ICoICT, IEEE, 2017, pp. 91–96.
[32] D.W. Hosmer, S. Lemeshow, R.X. Sturdivant, Applied Logistic Regression, third ed., Wiley, Hoboken, NJ, 2013.
[33] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.
[34] L. Breiman, Random forests, Mach. Learn. 45 (1) (2001) 5–32.
[35] D. Dua, C. Graff, UCI machine learning repository: Credit card fraud detection dataset, 2015, https://archive.ics.uci.edu/ml/datasets/Credit+Card+Fraud+Detection. (Accessed 23 May 2025).
[36] S. Bhattacharyya, S. Jha, K. Tharakunnel, J.C. Westland, Data mining for credit card fraud: A comparative study, Decis. Support Syst. 50 (3) (2011) 602–613.
[37] H. He, E.A. Garcia, Learning from imbalanced data, IEEE Trans. Knowl. Data Eng. 21 (9) (2009) 1263–1284.
[38] M. Schuld, F. Petruccione, Machine Learning with Quantum Computers, vol. 676, Springer, 2021.
[39] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, S. Lloyd, Quantum machine learning, Nature 549 (7671) (2017) 195–202.
[40] V. Havlíček, A.D. Córcoles, K. Temme, A.W. Harrow, A. Kandala, J.M. Chow, J.M. Gambetta, Supervised learning with quantum-enhanced feature spaces, Nature 567 (7747) (2019) 209–212.
[41] J.C. Spall, Multivariate stochastic approximation using a simultaneous perturbation gradient approximation, IEEE Trans. Autom. Control 37 (3) (2002) 332–341.
[42] T. Fawcett, An introduction to ROC analysis, Pattern Recognit. Lett. 27 (8) (2006) 861–874.
[43] A. Dal Pozzolo, G. Boracchi, O. Caelen, C. Alippi, G. Bontempi, Credit card fraud detection: A realistic modeling and a novel learning strategy, IEEE Trans. Neural Networks Learn. Syst. 29 (8) (2018) 3784–3797.
[44] P.K. Chan, W. Fan, A.L. Prodromidis, S.J. Stolfo, Distributed data mining in credit card fraud detection, IEEE Intell. Syst. Appl. 14 (6) (1999) 67–74.
[45] Y. Sahin, E. Duman, Detecting credit card fraud by decision trees and support vector machines, in: Proceedings of the International Multiconference of Engineers and Computer Scientists, vol. 1, 2011, pp. 1–6.