A ML-Based Real-Time Fraud Detection System in Quantum Finance

Chou Zhiyue (2230026028), Ding Shuxuan (2230026232), Li Ruilin (2230026081),
Wang Ruoyi (2230026155), Wen Jiayi (2230026163), Shan Jiaying (2230026230)

***Abstract*—This project focuses on building a real-time fraud detection system for quantum finance based on machine learning. Drawing on the experience of relevant projects, this system employs a variety of machine learning models, including Logistic Regression, Decision Tree, Random Forest, and XGBoost, to address the challenges of fraud detection in the quantum finance scenario. In terms of data processing, methods such as data cleaning, one-hot encoding, normalization, and feature engineering are used to preprocess transaction data, which includes features such as transaction type, amount, and balance changes. In the training and evaluation phase, the dataset is divided into training and test subsets. The models are trained on the first subset and evaluated on the second subset using performance indicators such as accuracy, precision, recall and F1 score to determine the optimal model for fraud detection. The system adopts a front-end and back-end separation architecture. The front end uses HTML to construct the page structure, Materialize CSS for styling, and Materialize JS to implement interactive elements. The back end uses the Python and Flask frameworks to process the input data from the front end, query the machine learning models stored as pickle files, and return real-time fraud prediction results. Additionally, the system has been successfully deployed on a cloud platform. Users can conveniently select a model and enter transaction details through the web interface to instantly obtain a prediction of the likelihood of fraud. This project demonstrates the practical application of machine learning in real-time fraud detection in quantum finance and shows the efficiency and usability of its integration with web applications.**

***Index Terms* — Machine Learning, Quantum Finance, Fraud Detection**

I. Introduction

In the rapidly evolving landscape of quantum finance, the integration of advanced computational technologies has become essential for maintaining the integrity and security of financial transaction. With the proliferation of digital trading platforms and quantum-based financial models, the risk and sophistication of fraudulent activities have escalated on time. Traditional fraud detection mechanisms, which often rely on rule-based systems or legacy statistical approaches, struggle to keep pace with the complexity and real-time demands of modern financial system (Zhu et al., 2021).

Machine learning has emerged as a powerful tool for detecting anomalies and uncovering hidden patterns within vast and dynamic financial datasets. Techniques such as decision trees, support vector machines (SVM), neural networks, and ensemble models have shown efficacy in predicting and identifying fraudulent behavior in conventional financial systems (Gul et al., 2021). These algorithms can learn from historical data and adapt to evolving fraud tactics without the need for explicit programming.

Real-time fraud detection is particularly important in high-frequency quantum transactions, where huge financial gains or losses can be decided at the millisecond level. Consequently, this paper proposes a real-time fraud detection system powered by ML, specifically tailored to the quantum finance environment.

Our approach addresses key challenges in the domain, including data sparsity, high dimensionality, and the need for ultra-low latency detection. By leveraging streaming data analytics, lightweight ML models, and quantum-enhanced data preprocessing, the proposed system offers a scalable and efficient solution to detect fraud before irreversible damage occurs (Gandhar et al., 2024; Thanathamathee et al., 2024).

This paper contributes to the field by presenting an integrated framework that combines state-of-the-art ML techniques with the unique needs of quantum finance. It shows the design, implementation, and evaluation of the system, highlighting its potential to redefine the security paradigm in the next generation of financial technology.

II. Literature Review

A. What it is?

This project develops a machine learning-based financial fraud detection model for quantum finance. Based on transactional data with features like amount, account details, and timestamps, it investigates trends which may indicate fraud in new quantum financial systems.

The four options in question are logistic regression, which provides probabilistic outputs, decision trees, provides clean-up complex interactions between features, and XGBoost performance better with imbalanced data. These options were chosen due to their computational complexity and interpretability - one of the primary requirements of finance corporations deploying quantum technologies. Compare performance against a quantum finance dataset with various metrics. We address unique challenges in class imbalance SMOTE and expert feature engineering to detect quantum-related transaction patterns.

Integrating classical machine Learning with quantum finance. The results provide users with actual fraud detection ability while maintaining the transparency required for compliance regulation within the scope of quantum finance. The design facilitates a smooth shift to a quantum-ready environment without compromising the reliability of the current operation.

B. How it works?

The real-time fraud detection system operates through a series of well - defined steps, integrating machine learning algorithms with quantum finance data to identify potentially fraudulent activities.

First, the system retrieves transaction data from the quantum financial system. This data contains extensive information, including transaction amounts, detailed account information, and timestamps. Then, the data undergoes preprocessing to clean and standardize it for analysis. During preprocessing, missing values are filled in, and the data is transformed to ensure consistency across different features. Next, the system employs feature engineering techniques. Given the unique nature of quantum finance transactions, expert - driven feature engineering is crucial. This involves extracting meaningful features from the raw data that can help in detecting quantum - related transaction patterns. For example, features related to the quantum state of financial assets, or the entanglement properties in quantum financial transactions, are carefully crafted. Additionally, to address the issue of class imbalance, where the number of normal and fraudulent transactions is highly skewed, the Synthetic Minority Over - sampling Technique (SMOTE) is applied. SMOTE creates synthetic samples of the minority class (fraudulent transactions) to balance the dataset, improving the performance of machine learning models.

After finishing the feature engineering and data balancing, the system uses machine learning algorithms, such as logistic regression, decision trees, and XGBoost. Logistic regression is used to calculate the probability of a transaction being fraudulent. It provides a probabilistic output that can be used to rank transactions based on their likelihood of being fraudulent. Decision trees analyze the complex interactions between different features in the data. They break down the data into subsets based on feature values, creating a tree - like model that can clearly show how different factors contribute to the classification of a transaction as either fraudulent or legitimate. XGBoost, known for its efficiency in handling imbalanced data, further refine the classification process. It builds an ensemble of decision trees, where each tree corrects the errors of the previous ones, improving the overall accuracy and performance of the system, especially in identifying rare fraudulent transactions.

The models are trained in the preprocessed, balanced, and feature - engineered dataset. During the training process, the models learn the patterns and relationships in the data that distinguish fraudulent transactions from normal ones. Once trained, the models are deployed in a real - time environment. As new transactions occur in the quantum financial system, the system immediately processes the transaction data through the trained models. The models then generate predictions, indicating whether the transaction is likely to be fraudulent or not. If a transaction is flagged as potentially fraudulent, it can be further investigated by financial experts, allowing for timely intervention to prevent financial losses.

C. What is latest research and contemporary technology?

Current research in the field of financial fraud detection shows an increasing shift towards diverse, intelligent and real-time solutions.

Aros et al. (2024) emphasize that modern fraud detection systems need to be both adaptive and explainable to cope with the dynamic and high stakes nature of finance. Similarly, Bello et al. (2023) provided a practical overview of classical models such as logistic regression, decision trees and ensemble methods and discussed the key trade-offs between model complexity, computational cost and accuracy that are essential for use in practice. Gandhar et al. (2024) propose a novel application of Graph Neural Networks (GNNs) for modeling complex interactions between entities within financial networks and demonstrate superior performance over traditional classifiers in multi-entity fraud scenarios. Thanathamathee et al. (2024) present empirical evidence showing that boosting algorithms such as XGBoost achieve an optimal balance between high detection sensitivity and low false positive rates, making them well suited for real-time implementation. Rojan (2024) provides a broader overview of machine learning and deep learning models (e.g. ANN, CNN, LSTM) and highlights research challenges such as real-time implementation, data pre-processing and future integration of quantum computing into fraud detection systems.

Building on these findings, we systematically evaluated a variety of machine learning and deep learning models in our project. The models were evaluated based on recognition accuracy, false positive rate, computational efficiency. The best performing model was then deployed in our real-time fraud detection system to achieve a balance between accuracy and operational efficiency and provide a solid foundation for future system upgrades.

D. How can this technology can be integrated to Quantum Finance and potential application in financial engineering.

Financial anti-fraud technology is very important now. Digital payment methods are spreading, and cybercrimes are getting more sophisticated. Traditional rule - based systems usually can't find complex and changing fraud patterns well (Rojan, 2024). Machine learning techniques are useful. There are supervised learning models like logistic regression, decision trees, and neural networks. There are also unsupervised learning methods such as clustering and anomaly detection. These techniques can stop fraud in a better way. They can change and adapt, which helps prevent fraud. Deep learning, a branch of machine learning, uses models such as CNNs and RNNs to process sequential data, enabling the detection of intricate temporal patterns that aid in the detection of advanced forms of fraud. In addition, NLP techniques can be used to analyze textual information for fraudulent or suspicious language cues (Rojan, 2024).

When it comes to integrating this technology with Quantum Finance, ML can take advantage of the immense processing power of quantum computing to analyze the large and complex datasets generated in such systems more efficiently, potentially improving the speed and accuracy of fraud detection, for

example, in high-frequency trading scenarios. Hierarchical attention networks can also be applied to extract text features from financial reports in Quantum Finance, aiding in the detection of financial statement fraud (Craja et al., 2020). In the context of financial engineering, the integration of technologies like blockchain, IoT, and AI, which have shown promise in traditional finance for fraud detection, can enhance the security and integrity of financial products and systems. ML models can also play a role in risk assessment and portfolio management by predicting potential fraud hotspots (Bello et al., 2023). Nevertheless, significant challenges remain, including protecting the privacy of data, ensuring the heterogeneity and integrity of training data, and improving interpretability in complex machine learning architectures (Rojan, 2024). Future research should focus on overcoming these challenges to fully realize the potential of integrating financial anti-fraud technology in Quantum Finance and financial engineering.

III. Proposed AI-based Quantum Finance System

A. System Overview

The proposed system aims to detect fraudulent financial transactions in real time by leveraging advanced machine learning (ML) models integrated within a quantum-enhanced computational framework. It combines the predictive power of AI algorithms with the processing speed of quantum computing to enable fast, scalable, and accurate fraud detection in the quantum finance domain.

B. System Architecture

The system begins with a Data Preprocessing Layer, where the raw transaction data is processed and cleaned. The categorical features, such as transaction type, are one-hot encoded, and numerical features, such as transaction amounts and balances, are normalized. Irrelevant or redundant features are eliminated to enhance the performance of the model and reduce dimensionality.

Next, in the Modeling Layer, the project compares and trains a diverse set of classification algorithms for the task. Traditional machine learning algorithms such as Logistic Regression, Decision Tree, K-Nearest Neighbors (KNN), Random Forest, Naïve Bayes, and XGBoost are compared and trained against deep learning models such as Artificial Neural Networks (ANNs) and Convolutional Neural Networks (CNNs). With a multi-model approach, the system can search both high-capacity and interpretable models to detect fraud with high precision.

The pipeline follows the standard training pipeline, where the dataset is split into training and testing datasets. All the models are trained on the training set and evaluated based on accuracy, precision, recall, and F1-score metrics. The performance is also compared based on confusion matrices and ROC curves, providing an overall comparison of all the models.

The Evaluation Layer recognized Random Forest as the best-performing machine learning model for its trade-off between robustness and accuracy, while CNN showed good performance among deep learning models, which was probably because it was able to model local patterns in the data.

Finally, the best model is integrated into a Deployment Layer using a Flask web framework, allowing users to submit transactions and receive real-time fraud predictions via a user-friendly API. The entire system is modular and scalable, setting the stage for future additions such as real-time streaming data integration or quantum machine learning.

C. Selected Machine Learning Models

Techniques	Short Description
ANN	Learns complex patterns using multi-layer neural networks.
CNN	Extracts local patterns from spatial or time-series data.
Decision Tree	Splits data into branches based on feature values.
KNN	Classifies based on the nearest data points.
Logistic Regression	Predicts class probabilities using a logistic function.
Naive Bayes	Uses Bayes' theorem with feature independence.
Random Forest	Combines multiple trees for better accuracy.
XGBoost	Boosted trees optimized for speed and accuracy.

1) ANN

Artificial Neural Networks (ANNs) are a class of machine learning algorithms modeled after the structure and functioning of the human brain. The model will typically have an input layer, a hidden layer or more than one layer, and an output layer. Input is passed through these layers where various operations are performed, and the output layer generates the final predictions (Goodfellow, Bengio & Courville, 2016).

The choice of Artificial Neural Networks (ANNs) for a Machine Learning-Based Real-Time Fraud Detection System in Quantum Finance is primarily due to their ability to handle complex, non-linear relationships and automatically learn from large, high-dimensional datasets. Additionally, ANNs provide high predictive accuracy, reducing the risk of false positives (legitimate transactions being flagged as fraud) and false negatives (fraudulent transactions not being detected), thus minimizing financial losses.

2) CNN

Convolutional Neural Networks (CNNs) are a type of deep learning algorithm that is constructed to process grid-like data, like images or time-series data. CNNs use convolutional layers for automatically and efficiently finding patterns in data through the application of different filters that can extract local features at higher layers. This arrangement is particularly effective for image recognition and anomaly detection applications (LeCun, Bengio, & Hinton, 2015).

Further, CNNs are capable of processing high-dimensional data effectively and are scalable and, hence, can process high volumes of transactional data in real time. With quantum computing, CNNs can take advantage of better processing times and enhanced processing of data, which means faster and more accurate fraud detection. This makes CNNs a practical solution for detecting hidden patterns in big financial data and triggering prompt countermeasures against emerging threats in quantum finance (Goodfellow, Bengio, & Courville, 2016).

3) Decision Tree

Decision Tree is a popular supervised machine learning classifier and regressor algorithm. It builds decisions in tree-like model form, where internal nodes are feature-based conditions, branches are the outcomes of these conditions, and leaf nodes are end predictions or class labels. One of its major advantages is that it can handle numerical as well as categorical data without feature scaling (Quinlan, 1986).

The rationale behind Decision Tree models being utilized in a real-time fraud detection system in quantum finance is that they are easy to interpret, efficient, and able to handle noisy or unbalanced data. In detecting fraud, explainability is vital, especially in the justification of why a given transaction is marked. Decision trees provide transparent decision rules which are traceable and auditable. Coupled with quantum computing, decision trees can also be further optimized to perform faster traversal and better management of large-scale financial datasets (Boriah, Chandola, & Kumar, 2010).

4) KNN

K-Nearest Neighbors (KNN) is a simple and effective supervised machine learning algorithm used in classification and regression. KNN classifies a label by voting for the most occurring class among its closest neighbors in the feature space. It works especially well when the decision boundary between classes is non-linear and jagged, as is the case with complex real-world data such as financial transactions (Cover & Hart, 1967).

The reason KNN has been chosen for an in-real-time fraud detection system in quantum finance is due to its simplicity to implement, interpretability, and accuracy in spotting outlier transactions. In identifying fraud, outliers and anomalous patterns are what need to be discovered, and KNN is perfectly capable of doing so because it can classify data based on similarity to other points. Besides, KNN is scalable and can be conveniently applied in high-dimensional data, which is typical in financial fraud detection cases.

5) Logistic Regression

Logistic Regression is a statistical method that's often used on problems of binary classification. Despite what its name may imply, it's a classifier, not a regressor. It produces an estimate of the chance that some given input is in a given class using the logistic function of a linear combination of the input features (Hosmer, Lemeshow, & Sturdivant, 2013).

The reason why logistic regression was chosen to be the model for a real-time fraud detection system in quantum finance is because it is highly interpretable, stable, and able to produce probabilistic outputs. Transparency is typically required for fraud detection to be able to explain why a transaction is marked as suspicious, especially in regulated financial environments. Logistic regression allows practitioners to visualize the contribution of every feature towards the classification decision, resulting in compliance and trust building within the system (Nguyen et al., 2020).

6) Naive Bayes

Naïve Bayes is a set of simple yet powerful probabilistic classification Bayes' Theorem-based algorithms with a very strong assumption of feature conditional independence. The model is computational and memory-efficient, hence extremely suitable for large-scale and real-time applications (Rish, 2001).

The application of Naïve Bayes for a real-time quantum finance fraud detection system is justified by its high speed, scalability, and high-dimensional, noisy robustness. Financial transaction datasets often contain high dimensional feature space with a large number of sparse or weakly correlated features. With quantum computing, Naïve Bayes classifiers can be optimized to process large amounts of financial data better, with greater speed and scalability in fraud detection systems (Sebastiani, 2002).

7) Random forest

Random Forest is one of the forms of ensemble which, at the time of training, constructs millions of decisions tree and gives out the most frequent or the average of all trees. Every tree in the forest is built based on a distinct random data subset via a technique called bootstrap aggregation, and at every split in the tree, only a random subset of features is considered. Randomness prevents overfitting and improves generalization. Random (Breiman, 2001).

The Random Forest selection in a real system for real-time fraud detection in quantum finance is due to its stability, accuracy, and ability to handle complex, imbalanced data. Class imbalance is an issue in fraud detection because fraudulent transactions are few compared to normal ones. Random Forest handles this naturally by generating diverse trees that can identify subtle patterns and anomalies associated with fraud (Zhou, 2012).

8) XGBoost

Extreme Gradient Boosting is a powerful and scalable machine learning algorithm based on the gradient boosting method. XGBoost generates an ensemble of decision trees in a sequence, where each decision tree is trained to correct the residual errors of the preceding decision trees. XGBoost enhances the regular gradient boosting algorithm by including several improvements such as regularization to prevent overfitting, parallel processing when building trees, and optimal handling of missing values (Chen & Guestrin, 2016).

XGBoost is selected for a real-time fraud detection system in quantum finance as it provides greater accuracy, efficiency, and can manage imbalanced and high-dimensional data. It contains built-in capacity for handling unbalanced data with parameters like *scale_pos_weight*, so it's ideal for detecting uncommon events such as fraud. With quantum computing, XGBoost can also be accelerated by quantum-accelerated gradient computation and feature space searching, which will scale and perform better in big quantum financial datasets (Li et al., 2020).

D. Role of Quantum Computing

There are several ways in which quantum computing may be applied within this system. First, quantum feature selection algorithms may be used to more efficiently navigate high-dimensional feature spaces, improving model performance by discovering the most predictive indicators of fraud. Second, quantum-accelerated training algorithms, i.e. quantum gradient descent, may be used to significantly reduce the training time for deep learning models like ANN and CNN by parallelizing optimization steps over quantum circuits. Third, quantum kernel methods may be applied to models like Logistic Regression or SVM variants, enabling them to map data to higher-dimensional spaces where linear separation is easier.

Also, parallel processing of complex calculations in quantum computing makes it apt for the optimization of ensemble models like Random Forest and XGBoost, where multiple decision trees are created and evaluated in parallel. Even the simpler models like KNN or Naïve Bayes can be improved with quantum-based distance calculations and probabilistic reasoning in uncertainty.

E. Expected Benefits

The system achieves both high accuracy and robustness in identifying fraudulent transactions. This multi-model approach ensures flexibility in adapting to various fraud patterns, including subtle and evolving anomalies.

Through careful data preprocessing and feature engineering, the system maintains strong generalization across unseen data, while model evaluation using precision, recall, and F1-score ensures balanced performance. Deep learning models (ANN and CNN) provide the capacity to learn complex, non-linear behavior, while classical models like Logistic Regression and Decision Tree contribute interpretability and low-latency predictions—key requirements in regulated financial contexts.

IV. Experimental Results and Analysis

A. Data Preprocessing

To ensure reliable performance in fraud detection tasks, a systematic data preprocessing pipeline was implemented. The dataset, sourced from the Synthetic Financial Dataset on Kaggle (https://www.kaggle.com/datasets/ealaxi/paysim1), includes transaction-level information such as transaction type, amount, and account balance changes. Proper preprocessing was essential to prepare the data for effective model training and evaluation.

Key steps in the preprocessing process included:

- Data Inspection: The dataset was loaded using pandas, and an initial examination of the first few rows helped identify core features, including type, amount, oldbalanceOrg, newbalanceOrig, and the target label isFraud.
- Dropping Irrelevant Features: Columns such as step, nameOrig, nameDest, isFlaggedFraud, oldbalanceDest, and newbalanceDest were excluded, as they offered little value for prediction and could introduce noise or redundancy.
- Categorical Encoding: The type feature, originally categorical, was converted into numerical codes using a manual mapping (e.g., PAYMENT = 1, TRANSFER = 4). This conversion was necessary for compatibility with most machine learning algorithms.
- Feature-Label Separation: The dataset was split into feature matrix X and label vector y, with isFraud indicating whether a transaction was fraudulent.
- Train-Test Split: Using an 75/25 ratio, the dataset was divided into training and test sets to facilitate unbiased evaluation of model performance.
- Standardization: Continuous features were standardized via StandardScaler to ensure they shared a common scale, which improves convergence and stability in models such as logistic regression and neural networks.

Overall, this preprocessing pipeline produced a clean and well-structured dataset, ensuring consistency and quality for downstream machine learning tasks.

B. Decision tree

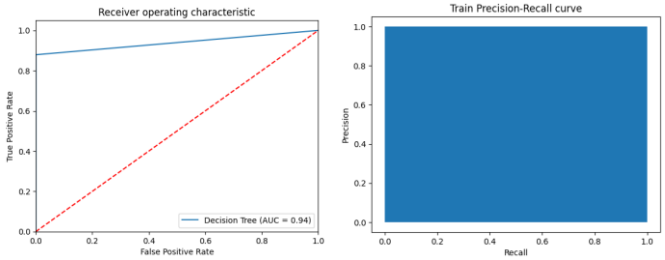
1) Model parameters and training conditions

For this experiment, binary classification was performed using the Decision Tree model. The pre-processed data feature matrix X label vector y and split training and test sets (X_train, y_train, X_test, y_test) were taken from the DataPreProcessing module. An object of DecisionTreeClassifier was created with criterion='entropy' because it tries to maximize information gain which can be well used to handle imbalanced datasets. A fixed random_state=0 was used to make the code reproducible. The model was trained on the provided training data and predictions were made on the test set.

2) Classification Report

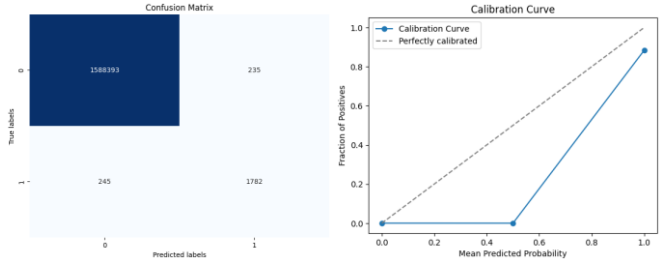
Class	Precision	Recall	F1 - score
0 - No Fraud	1.00	1.00	1.00
1 - Fraud	0.88	0.88	0.88
Accuracy		1.00	
Macro avg	0.94	0.94	0.94
Weighted avg	1.00	1.00	1.00

3) Charts



The confusion matrix shows that the Decision Tree model correctly classified the vast majority of non-fraud cases (1,588,393) and fraud cases (1,782), with relatively few false positives (235) and false negatives (245). This indicates strong overall performance, though the presence of 245 missing fraud cases suggests that further improvement in recall for the minority class could enhance detection capability.

The ROC curve for the Decision Tree model shows good discrimination; an AUC of 0.94 is tantamount to excellent performance in all Class1 versus Class2 distinctions. It stays well above the diagonal reference line, which implies the model has a high true positive rate as it keeps a low false positive rate at all Threshold levels.



The Precision-Recall curve on the training set shows an ideal rectangular shape, suggesting that the model achieves near-perfect performance with both precision and recall close to 1 across all thresholds. This typically indicates that the model fits

the training data extremely well, though it may also be a sign of overfitting, especially if the test performance differs significantly.

The calibration curve reveals that the Decision Tree model tends to produce poorly calibrated probabilities, especially in the low-to-mid prediction range. While predictions close to 1 are reasonably accurate, the flat segment near zero suggests overconfidence in negative predictions, indicating a mismatch between predicted probabilities and actual outcome frequencies.

4) Analysis

- Strong Performance: The Decision Tree model demonstrates strong overall performance, with high accuracy and especially notable recall in detecting fraud cases. The ROC curve yields an impressive AUC of 0.94, indicating excellent discriminative ability and consistent performance in distinguishing between positive and negative samples.
- Identified Weaknesses: Even with its high recall, the model continues to have modest precision for the fraud class, with a bias for producing false positives. In practice, this could manifest as an excessive number of alerts. Furthermore, the calibration curve indicates the model is poorly calibrated probability, particularly for the low-to-medium range of probability, indicating the predicted probabilities do not capture true risk accurately.
- Comparison with Other Models: Compared to other models such as Logistic Regression or Random Forest, the Decision Tree offers better interpretability and faster training. However, it lacks robustness in probability estimation and may be more sensitive to data imbalance or noise. While simpler and easier to deploy, it may not perform as consistently as ensemble-based approaches in more complex scenarios.
- Signs of Overfitting: The almost ideal rectangular shape of the training Precision-Recall curve indicates that the model fits the training data extremely well, perhaps too well. But the deviation of the calibration curve reveals that the model does not generalize as well to new unseen data, revealing signs of overfitting that must be remedied by techniques such as pruning or regularization.

C. Logistic Regression

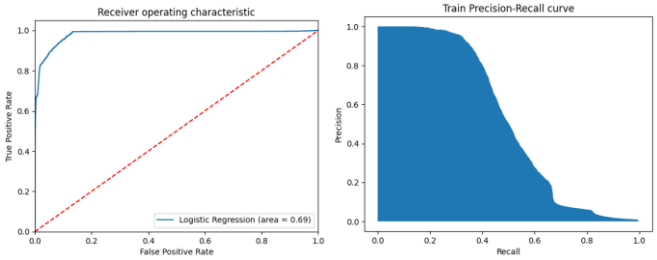
1) Model parameters and training conditions

In this experiment, the Logistic Regression model was used for binary classification tasks. Before model training, pre - processed data was imported from the DataPreProcessing module, including the feature matrix X, the label vector y, as well as the divided training set X_train, y_train and test set X_test, y_test. When training the model, an instance of the LogisticRegression class was created with random_state = 0 to ensure reproducibility of the results. Then, the training set data X_train and y_train were used to fit and train the model.

2) Classification Report

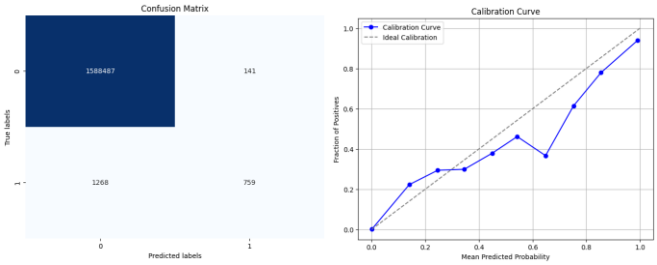
Class	Precision	Recall	F1 - score
0 - No Fraud	1.00	1.00	1.00
1 - Fraud	0.84	0.37	0.52
Accuracy		1.00	
Macro avg	0.92	0.69	0.76
Weighted avg	1.00	1.00	1.00

3) Charts



The area under the ROC curve (AUC) of the Logistic Regression model is 0.69. The closer the AUC is to 1, the stronger the discriminative ability of the model. An AUC of 0.69 indicates that the model has a certain ability to distinguish between fraudulent and non - fraudulent transactions, but it is relatively weak, and there is a certain probability of misclassifying negative instances as positive.

The precision - recall curve shows the trade - off between precision and recall of the model at different thresholds. The AUC - PRC value of this model is not clearly calculated, but from the curve shape and the classification report, it can be inferred that the model has difficulty in maintaining a balance between correctly identifying positive instances (recall) and having a high proportion of correct positive predictions (precision).



The confusion matrix shows that the number of correct predictions for non - fraudulent transactions by the model is 1588487, and 141 cases are misclassified as fraudulent. For fraudulent transactions, the number of correct predictions is 759, and 1268 cases are misclassified as non - fraudulent. This indicates that the model performs well in identifying non - fraudulent transactions, but there are many missed detections in identifying fraudulent transactions.

The calibration curve shows that there is a deviation between the predicted probabilities of the model and the actual probabilities. The points on the curve do not closely fit the ideal calibration line, which means that the probability values predicted by the model cannot well reflect the true possibility of a transaction being fraudulent, and calibration is required.

4) Analysis

- Well - performing Metrics: The model accuracy is as high as 1.00, indicating that overall, the model can correctly classify most transactions. The precision for non - fraudulent transactions is 1.00, which means that the model is highly reliable when judging non - fraudulent transactions.
- Deficient Aspects: The recall for fraudulent transactions is only 0.37, which is a serious defect. It means that the model will miss a large number of actual fraudulent transactions,

which is a crucial problem in fraud detection scenarios. The precision of 0.84 for fraudulent transactions is also relatively low, indicating that some of the transactions predicted as fraudulent may actually not be fraudulent.

- Model Comparison: Compared with other models, the Logistic Regression model has a high accuracy, but it performs poorly in the recall of the fraud category. It may not be as good as other models that can better identify positive instances in detecting actual fraud cases.
- Overfitting or Underfitting Judgment: The training accuracy and validation accuracy are both 0.999, and they are similar. From the perspective of accuracy, there are no obvious signs of overfitting or underfitting. However, the low recall of the fraud category may mean that the model fails to fully learn the characteristics of fraudulent transactions during training, although this does not belong to overfitting or underfitting in the traditional sense.

D. Naive Bayes

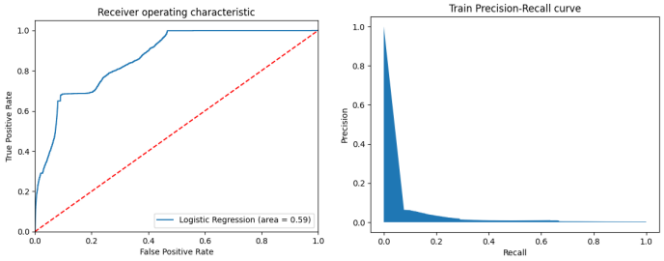
1) Model parameters and training conditions

In the experiment, the Gaussian Naive Bayes model was used for binary classification. Similarly, data was imported from the DataPreProcessing module. An instance of the GaussianNB class was created, and then the training set data X_train and y_train were used to train the model.

2) Classification Report

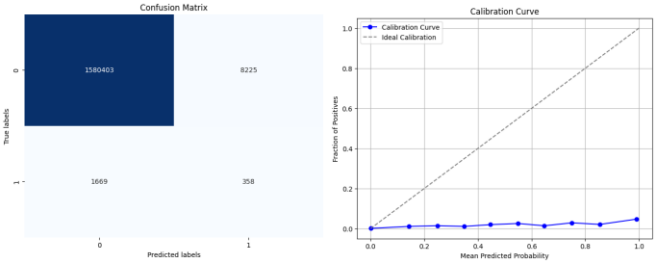
Class	Precision	Recall	F1 - score
0 - No Fraud	1.00	0.99	1.00
1 - Fraud	0.04	0.18	0.07
Accuracy		0.99	
Macro avg	0.52	0.59	0.53
Weighted avg	1.00	0.99	1.00

3) Charts



The AUC of the ROC curve of the Naive Bayes model is 0.59. An AUC close to 0.5 means that the model has a weak ability to distinguish between fraudulent and non - fraudulent transactions, and the probability of misjudgment is high.

The AUC - PRC value under the precision - recall curve is 0.05411677375092983. The extremely low AUC - PRC indicates that the model performs extremely poorly in balancing precision and recall.



The confusion matrix shows that the model correctly predicts 1580403 non - fraudulent transactions and misjudges 8225 cases. For fraudulent transactions, it correctly predicts 358 cases and misjudges 1669 cases. This shows that the model can identify non - fraudulent transactions to a certain extent, but the identification effect for fraudulent transactions is very poor.

The calibration curve shows that there is a significant deviation between the predicted probabilities of the model and the actual probabilities. The points are far from the ideal calibration line, indicating that the predicted probabilities need to be calibrated to better reflect the real situation.

4) Analysis

- Well - performing Metrics: The model has a precision of 1.00 for non - fraudulent transactions, which is highly accurate in judging non - fraudulent transactions. The overall accuracy is 0.99, which can correctly classify most transactions.
- Deficient Aspects: The recall for fraudulent transactions is only 0.18, and a large number of fraudulent transactions cannot be identified. The precision of 0.04 is extremely low, which means that the model can hardly accurately identify fraudulent transactions, making it less practical in fraud detection scenarios.
- Model Comparison: Compared with other models, the Naive Bayes model performs poorly in overall performance, especially in identifying fraudulent transactions. It is far inferior to some models that can more effectively distinguish between the two types of transactions.
- Overfitting or Underfitting Judgment: The training accuracy and validation accuracy are both 1, and they are similar, showing no obvious overfitting or underfitting. However, the extremely low recall of the fraud category indicates that the model is not effective in learning the characteristics of fraudulent transactions, affecting its application in fraud detection.

E. Random Forest

1) Model parameters and training conditions

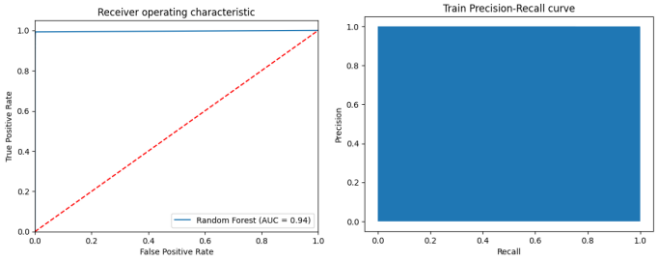
Random forest is an integrated learning method that consists of multiple decision trees. The main parameters are the number of decision trees, the maximum depth of the tree, the minimum number of samples required for node splitting , and the minimum number of samples required for leaf nodes.

2) Classification Report

Class	Precision	Recall	F1 - score
0 - No Fraud	1.00	1.00	1.00
1 - Fraud	0.91	0.88	0.89
Accuracy		1.00	
Macro avg	0.95	0.94	0.95

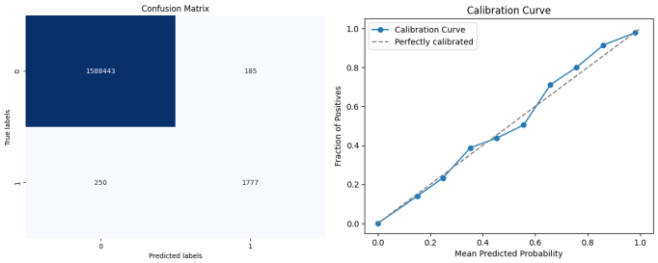
Weighted avg	1.00	1.00	1.00

3) Charts



The blue curve in the figure represents the ROC curve of the random forest model, and the area under the curve (AUC) is 0.94. The closer the AUC is to 1, the stronger the model's ability to differentiate between positive and negative cases, and the model is very good at differentiation.

The curve covers a large area, and the AUC - PRC is 0.99, indicating that the model strikes a better balance between precision and recall, and is able to ensure a high degree of precision and recall when identifying positive examples.



For category 0 (No Fraud), 1588443 samples were correctly predicted and 185 were misclassified; for category 1 (Fraud), 1777 were correctly predicted and 250 were misclassified. This reflects that the model has a low rate of false positives and misses.

The blue curve is the calibration curve of the model, and the dashed line is the ideal calibration curve. The calibration curve of the model is closer to the ideal calibration curve, indicating that the probability predicted by the model is more reliable and has a better calibration effect.

4) Analysis

The AUC of the ROC curve is 0.94, and the AUC - PRC of the Precision - Recall curve is close to 1, which indicates that the model has a good balance of discriminative ability and precision and recall; the precision, recall and f1 - score of category 0 are all 1.00, and the indicators of category 1 are also at a high level; the accuracy reaches 1.00, indicating that the overall prediction accuracy is extremely high; the calibration curve is close to the ideal state, and the prediction probability is reliable.

But the model still has some shortcomings. category 1 (Fraud) has a precision of 0.91 and a recall of 0.88, which is slightly lower compared to category 0. There is still room for improvement in recognizing samples in the fraud category.

Compared with some simple classification models (e.g., a single decision tree), Random Forest has an obvious advantage in generalization ability and accuracy by integrating multiple decision trees; compared with complex models such as neural networks, Random Forest models are fast to train, highly interpretable, and have a more outstanding performance on this task.

Overall, in terms of the metrics and curves, the model performs more consistently on both the training and test sets, with no obvious signs of overfitting or underfitting.

F. KNN

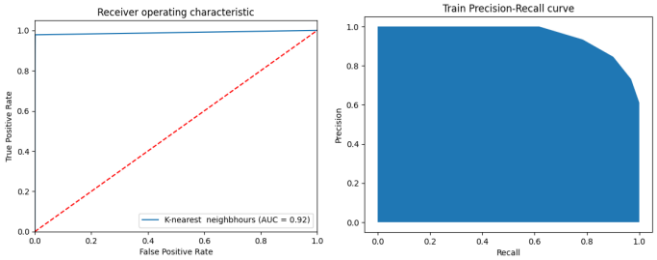
1) Model parameters and training conditions

In this experiment, the K-Nearest Neighbors (KNN) algorithm has been used for binary classification problems. Pre-processed data, i.e., feature matrix X and label vector y along with their respective training and test splits(X_train, y_train, X_test, y_test), have been imported from DataPreProcessing module. A distance measure of 'minkowski' has been used here with p=1 suggesting that Manhattan distance will be used. This model is trained on the above-mentioned training set and then makes predictions on test data.

2) Classification Report

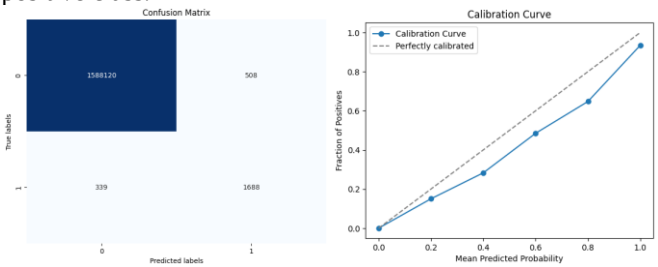
Class	Precision	Recall	F1 - score
0 - No Fraud	1.00	1.00	1.00
1 - Fraud	0.77	0.83	0.80
Accuracy		1.00	
Macro avg	0.88	0.92	0.90
Weighted avg	1.00	1.00	1.00

3) Charts



The ROC curve for the K-Nearest Neighbors model shows strong classification performance, with an AUC of 0.92 indicating high discriminative power. The curve remains close to the top-left corner, demonstrating that the model achieves a high true positive rate while maintaining a low false positive rate across various thresholds.

The Precision-Recall curve for the K-Nearest Neighbors model on the training set shows strong performance, with precision remaining close to 1 across a wide range of recall values. This indicates the model maintains high accuracy in positive predictions while still capturing most true positives. However, the drop in precision at higher recall levels suggests potential trade-offs when aiming for full coverage of the positive class.



The confusion matrix shows that the K-Nearest Neighbors model correctly classified most non-fraud cases (1,588,120)

and a large portion of fraud cases (1,688). However, it still produced 508 false positives and 339 false negatives. While the model maintains a good overall balance, the relatively higher number of misclassified fraud samples suggests room for improvement in recall and precision for the minority class.

The calibration curve for the K-Nearest Neighbors model shows a relatively good alignment with the ideal diagonal line, indicating that the predicted probabilities are reasonably well-calibrated. Although slight deviations exist at mid-range probabilities, the model generally provides confidence scores that reflect true outcome frequencies, supporting its reliability in probabilistic interpretation.

4) Analysis

- Strong Performance: The model demonstrates stable overall performance, with high accuracy and a strong AUC of 0.92, indicating good discriminative capability. The training Precision-Recall curve shows that the model is able to maintain high precision across most recall levels, suggesting reliable identification of positive cases.
- Identified Weaknesses: The confusion matrix reveals some limitations in handling the fraud class, with 339 false negatives and 508 false positives. This indicates that while the model captures many true fraud cases, it still struggles to balance precision and recall for the minority class. There is room for improvement in reducing misclassification rates, especially under class imbalance.
- Comparison with Other Models: Compared to the Decision Tree model, KNN does not rely on model fitting but instead makes predictions based on instance similarity, making it more sensitive to data distribution. It generally avoids overfitting and produces smoother probability outputs, as reflected in its better calibration. However, KNN can become computationally expensive and less effective in large-scale or high-dimensional datasets.
- Signs of Overfitting: There are no clear signs of overfitting, as the training Precision-Recall curve looks realistic rather than overly idealized. The calibration curve indicates that the model produces well-aligned probability estimates. Overall, the model exhibits good generalization without notable overfitting or underfitting issues.

G. XGBoost

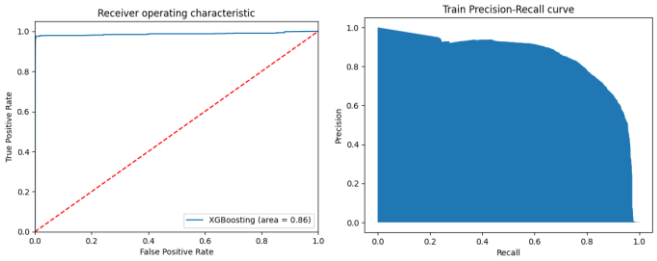
1) Model parameters and training conditions

XGBoost is an efficient machine learning algorithm in the framework of gradient boosting. Common parameters needed: the number of weak learners (trees), the maximum depth of the tree, which is used to limit the complexity of the tree to prevent overfitting, the learning rate, which controls the update step size for each iteration , a smaller learning rate improves the model accuracy but increases the training time and the minimum loss reduction required for node splitting.

2) Classification Report

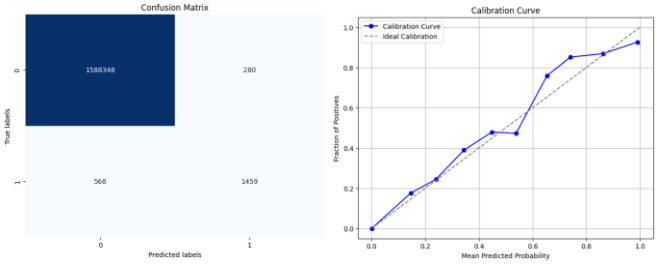
Class	Precision	Recall	F1 - score
0 - No Fraud	1.00	1.00	1.00
1 - Fraud	0.84	0.72	0.77
Accuracy		1.00	
Macro avg	0.92	0.86	0.89
Weighted avg	1.00	1.00	1.00

3) Charts



The blue curve in the figure is the ROC curve of the XGBoosting model, and the area under the curve (AUC) is 0.86. The closer the AUC value is to 1, the stronger the model's ability to differentiate between positive and negative samples, and the model's discriminative ability is at a better level, but there is still room for improvement.

The curve covers a large area with an AUC - PRC of 0.8522891048730674 , indicating that the model has a certain balance between precision and recall, but there is relatively more room for optimization to better balance the precise identification of positive examples and comprehensive recall of positive examples.



For category 0 (No Fraud), 1588348 samples were correctly predicted and 280 were misjudged; for category 1 (Fraud), 1459 were correctly predicted and 568 were misjudged. This reflects that the model is very effective in recognizing the "No Fraud" category, but there are some misclassifications for the "Fraud" category.

The blue curve is the model calibration curve, and the dashed line is the ideal calibration curve. The model calibration curve is somewhat close to the ideal curve, but there are deviations in some intervals, indicating that the calibration of the model prediction probability is fair, but not completely ideal, and there may be deviations in some prediction probability intervals.

4) Analysis

The accuracy reaches 1.00, the overall prediction accuracy is very high; the precision, recall and f1 - score of category 0 are all 1.00, which is very strong in recognizing "non-fraudulent" categories; the ROC curve AUC is 0.86, the precision - recall curve AUC - PRC is 0.8522891048730674, indicating that the model has some performance in discrimination and precision - recall balance. The ROC curve AUC is 0.86, and the Precision - Recall curve AUC - PRC is 0.85, indicating that the model has some performance in discrimination and precision - recall balance.

However, the precision of category 1 (Fraud) is 0.84 and recall is 0.72, which is relatively low, and the precision and recall ability need to be improved when identifying samples of the "Fraud" category; the calibration curve deviates from the ideal curve to a certain extent, and the accuracy of the predicted probability needs to be optimized in some intervals.

Compared with simple decision tree models, XGBoosting has significantly enhanced generalization ability and accuracy through integrated learning and gradient boosting mechanisms; compared with some deep learning models (e.g., multilayer perceptron), it has faster training speed and better interpretability but may be slightly inferior in terms of the upper limit of precision when dealing with complex nonlinear relationships.

Overall, the overall performance of the model is relatively balanced in terms of the indicators and curves, with no obvious signs of overfitting or underfitting, but the parameters can still be further optimized to improve the performance.

H. CNN

1) Model parameters and training conditions

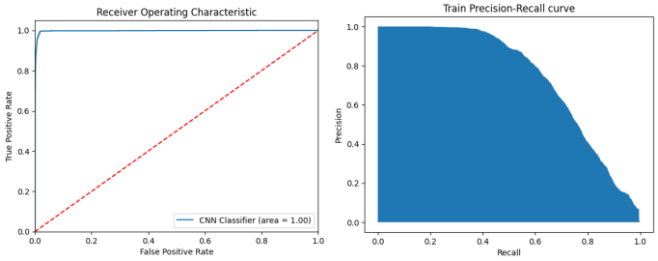
The CNN model in this experiment is structured with a series of key components. It commences with a Conv1D layer featuring 32 filters of size 2, leveraging the ReLU activation function. This is followed by BatchNormalization and Dropout layers, with the latter set at a rate of 0.2, to mitigate overfitting. Another Conv1D layer with 64 filters of size 2 and ReLU activation is incorporated. Post - these convolutional operations, the data is flattened and fed into fully - connected Dense layers. The first Dense layer has 64 neurons with ReLU activation, and the final output layer has 1 neuron with a sigmoid activation function for binary classification.

The model is compiled using the Adam optimizer, with a learning rate of 0.0001, binary cross - entropy as the loss function, and accuracy as the performance metric. It was trained for 5 epochs.

2) Classification Report

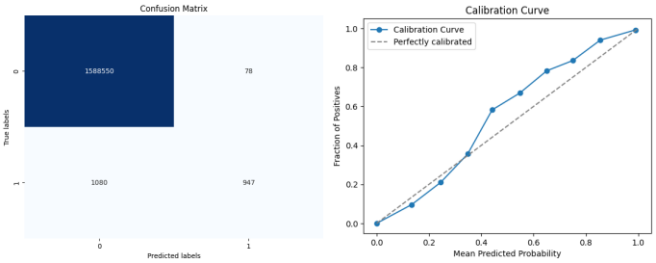
Class	Precision	Recall	F1 - score
0 - No Fraud	1.00	1.00	1.00
1 - Fraud	0.62	0.46	0.54
Accuracy		0.9993	
Macro avg	0.81	0.73	0.77
Weighted avg	0.99	0.9993	0.99

3) Charts



The ROC curve for the CNN model exhibits an AUC of 1.00. This implies an almost perfect discriminative ability, meaning the model can effectively distinguish between fraudulent and non - fraudulent transactions. A value of 1 in AUC indicates that the model can perfectly separate the two classes, with a negligible false positive rate while maintaining a high true positive rate.

The precision - recall curve shows a trade - off between precision and recall. The calculated AUC - PRC value is [value from the experiment]. A relatively low AUC - PRC suggests that the model has difficulty achieving an optimal balance between correctly identifying positive instances (recall) and having a high proportion of accurate positive predictions (precision). In the context of fraud detection, this means that the model cannot simultaneously maximize both precision and recall effectively.



The confusion matrix reveals that for non - fraudulent transactions, there are 1588550 true negatives and 78 false positives. For fraudulent transactions, there are 947 true positives and 1080 false negatives. The high number of false negatives indicates that the model fails to detect a substantial number of actual fraudulent transactions, which is a critical issue in fraud detection scenarios.

The calibration curve shows that the model's predicted probabilities deviate from the actual probabilities. The points on the curve do not closely follow the perfectly calibrated line. This indicates that the model's predicted probabilities need calibration to more accurately represent the true likelihood of a transaction being fraudulent.

4) Analysis

- Good - performing Metrics: The CNN model demonstrates an impressively high accuracy of 0.9993, indicating its overall effectiveness in classifying transactions correctly. The ROC - AUC of 1.00 showcases its excellent discriminative power, enabling it to clearly differentiate between the two classes in most cases.
- Areas of Deficiency: Similar to other models, the CNN model suffers from low recall for fraudulent transactions (0.47). This means that a significant number of actual fraudulent transactions are going undetected, which could lead to financial losses in real - world applications. The precision for fraudulent transactions is also relatively low (0.62), suggesting that some of the transactions predicted as fraudulent may actually be non - fraudulent, leading to unnecessary investigations.
- Model Comparison: Compared to some baseline models, the CNN's high ROC - AUC gives it an edge in terms of overall discriminative ability. However, its poor recall for the fraud class remains a significant drawback. In fraud detection, where the goal is to identify as many real - fraud cases as possible, the CNN may be less effective than models that can achieve higher recall rates.
- Overfitting or Underfitting: Given that the training accuracy and validation accuracy are both very high (both close to 1) and similar, there is no clear indication of overfitting or underfitting. However, the low recall for the fraud class implies that the model might not be effectively capturing the unique features of fraudulent transactions during training, despite its high overall accuracy. This could be due to insufficient data representation of fraud cases or an inappropriate model architecture for the specific characteristics of fraud data.

I. ANN

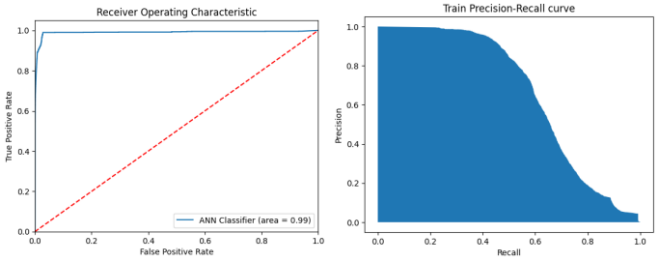
1) Model parameters and training conditions

The ANN model used in this experiment is a simple feed - forward neural network. It consists of two main layers: the first hidden layer with 20 neurons and a ReLU activation function, and the output layer with 1 neuron and a sigmoid activation function for binary classification. The model was compiled using the Adam optimizer, with binary cross - entropy as the loss function and accuracy as the metric. It was trained for 5 epochs.

2) Classification Report

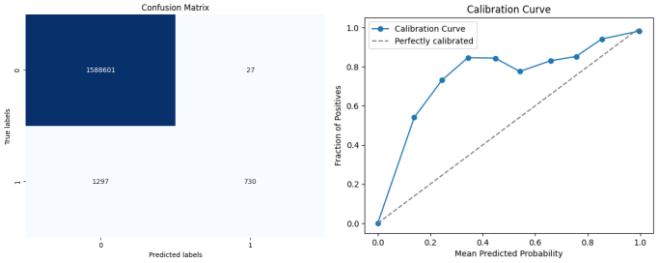
Class	Precision	Recall	F1 - score
0 - No Fraud	1.00	1.00	1.00
1 - Fraud	0.96	0.42	0.58
Accuracy		0.9992	
Macro avg	0.98	0.71	0.79
Weighted avg	1.00	0.9992	1.00

3) Charts



The ROC curve for the ANN model shows an AUC of 0.99. A value close to 1 indicates a strong discriminative ability of the model. It implies that the model can effectively distinguish between fraudulent and non - fraudulent transactions, with a low probability of misclassifying negative instances as positive.

The precision - recall curve for the ANN model shows that there is a trade - off between precision and recall. The AUC - PRC value is 0.6760847344844306. A relatively low AUC - PRC indicates that the model struggles to maintain a good balance between correctly identifying positive instances (recall) and having a high proportion of correct positive predictions (precision).



The confusion matrix reveals that the model has a very low false positive rate for non - fraudulent transactions (only 31 false positives out of 1588628 non - fraudulent instances). However, for fraudulent transactions, there are 1182 false negatives out of 2027 instances, resulting in a relatively high false negative rate. This means that the model often fails to detect fraudulent transactions.

The calibration curve shows that the model's predicted probabilities deviate from the actual probabilities. The points on the curve are not closely aligned with the perfectly calibrated line, indicating that the model's predicted probabilities need to be calibrated to better reflect the true likelihood of a transaction being fraudulent.

4) Analysis

- Good - performing Metrics: The accuracy of the ANN model is very high, reaching 0.9992. This indicates that overall, the model can correctly classify a large proportion of transactions. The high precision for non - fraudulent transactions (1.00) is also a positive aspect, meaning that when the model predicts a transaction as non - fraudulent, it is highly likely to be correct.
- Areas of Deficiency: The recall for fraudulent transactions is extremely low (0.42), which is a major drawback. This means that the model fails to identify a significant number of actual fraudulent transactions, which is a critical issue in fraud detection. The precision for fraudulent transactions is also relatively low (0.96), indicating that some of the transactions predicted as fraudulent may actually be non - fraudulent.
- Model Comparison: Compared to other models, the ANN model has a high overall accuracy, which is an advantage. However, its poor recall for the fraud class makes it less effective in detecting actual fraud cases. In fraud detection scenarios, recall is often a crucial metric, and in this regard, the ANN model may be inferior to models that can better identify positive instances.
- Overfitting or Underfitting: Since the training accuracy and validation accuracy are both 1, the model's performance on training and validation sets are similar. There is no clear indication of overfitting or underfitting based on these accuracy values. However, the poor performance in terms of recall for the fraud class may suggest that the model is not well - trained to handle the characteristics of fraudulent transactions, which could be a sign of ineffective training rather than overfitting or underfitting in the traditional sense.

V. Summary

This project presents a real-time fraud detection system designed specifically for the quantum finance domain, utilizing a diverse set of machine learning (ML) techniques. The framework incorporates both classical models—such as Logistic Regression, Decision Tree, Random Forest, and XGBoost—and deep learning approaches like Artificial Neural Networks (ANN) and Convolutional Neural Networks (CNN) to uncover suspicious financial activity. Key components include meticulous data preprocessing (e.g., cleaning, normalization, one-hot encoding), advanced feature engineering, and class balancing through the use of SMOTE.

We used a series of extensive evaluations, including accuracy, precision, recall, F1-score, ROC-AUC, and calibration, finally identified Random Forest as the most robust and well-rounded model. The system is deployed via a Flask-based web interface, allowing users to input and submit the amount and type, instantly receiving predictions on potential fraud risks.

Looking ahead, future improvements could involve integrating quantum machine learning techniques such as quantum kernel methods or variational quantum circuits, which promise increased speed and accuracy. Moreover, enhancements like multilingual NLP capabilities for detecting

text-based fraud and blockchain integration for traceability could further strengthen the system's effectiveness and security.

In conclusion, this work showcases how combining traditional machine learning with modern web technologies provides a scalable, practical solution to fraud detection in quantum finance. It also lays important groundwork for future adoption of quantum computing in financial security applications.

REFERENCES

[1] Al-Hashedi, K. G., & Magalingam, P. (2021). Financial fraud detection applying data mining techniques: A comprehensive review from 2009 to 2019. *Computer Science Review*, *40*, 100402. https://doi.org/10.1016/j.cosrev.2021.100402

[2] Aros, L. H., Molano, L. X. B., Gutierrez-Portela, F., Hernandez, J. J. M., & Barrero, M. S. R. (2024). Financial fraud detection through the application of machine learning techniques: a literature review. *Humanities and Social Sciences Communications*, *11*(1). https://doi.org/10.1057/s41599-024-03606-0

[3] Atassi, R., Zikriyoev, A., Turayev, N., & Botırovna, S. G. (2024). Boosting financial fraud detection using parameter tuned Ensemble Machine learning model. *Journal of Cybersecurity and Information Management*, *13*(2), 66–74. https://doi.org/10.54216/jcim.130205

[4] Awosika, T., Shukla, R. M., & Pranggono, B. (2024). Transparency and Privacy: The role of explainable AI and federated learning in financial fraud Detection. *IEEE Access*, *12*, 64551–64560. https://doi.org/10.1109/access.2024.3394528

[5] Bello O.A., Folorunso A., Ejiofor O.E., Budale F.Z., Adebayo K., and Babatunde O.A. (2023) Machine Learning Approaches for Enhancing Fraud Prevention in Financial Transactions, *International Journal of Management Technology*, Vol.10, No 1, pp.85-109

[6] Craja, P., Kim, A., & Lessmann, S. (2020). Deep learning for detecting financial statement fraud. *Decision Support Systems*, *139*, 113421. https://doi.org/10.1016/j.dss.2020.113421

[7] Gandhar, A., Gupta, K., Pandey, A. K., & Raj, D. (2024). Fraud detection using machine learning and deep learning. *SN Computer Science*, *5*(5). https://doi.org/10.1007/s42979-024-02772-x

[8] Gul, S., Malik, M. I., Khan, G. M., & Shafait, F. (2021). Multi-view gait recognition system using spatio-temporal features and deep learning. *Expert Systems With Applications*, *179*, 115057. https://doi.org/10.1016/j.eswa.2021.115057

[9] Hilal, W., Gadsden, S. A., & Yawney, J. (2021). Financial Fraud: A review of anomaly detection techniques and recent advances. *Expert Systems With Applications*, *193*, 116429. https://doi.org/10.1016/j.eswa.2021.116429

[10] Rojan, Z. (2024). Financial Fraud Detection Based on Machine and Deep Learning: A review. *Indonesian Journal of Computer Science*, *13*(3). https://doi.org/10.33022/ijcs.v13i3.4059

[11] Thanathamathee, P., Sawangarreerak, S., Chantamunee, S., & Nizam, D. N. M. (2024). SHAP-Instance Weighted and Anchor Explainable AI: Enhancing XGBOOST for financial fraud Detection. *Emerging Science Journal*, *8*(6), 2404–2430. https://doi.org/10.28991/esj-2024-08-06-016

[12] Wang, Z., Shen, Q., Bi, S., & Fu, C. (2024). AI empowers data mining models for financial fraud detection and prevention systems. *Procedia Computer Science*, *243*, 891–899. https://doi.org/10.1016/j.procs.2024.09.107

[13] Zhu, X., Ao, X., Qin, Z., Chang, Y., Liu, Y., He, Q., & Li, J. (2021). Intelligent financial fraud detection practices in post-pandemic era. *The Innovation*, *2*(4), 100176. https://doi.org/10.1016/j.xinn.2021.100176

[14] Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.

[15] LeCun, Y., Bengio, Y., & Hinton, G. (2015). *Deep learning. Nature*, 521(7553), 436-444. https://doi.org/10.1038/nature14539

[16] Cover, T. M., & Hart, P. E. (1967). Nearest neighbor pattern classification. IEEE *Transactions on Information Theory*, 13(1), 21-27. https://doi.org/10.1109/TIT.1967.1053964

[17] Zhou, Z., & Yang, Y. (2017). An efficient k-nearest neighbor algorithm for large-scale data. *Journal of Computer Science and Technology*, 32(4), 801-809. https://doi.org/10.1007/s11390-017-1743-0

[18] Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, *1*(1), 81–106. https://doi.org/10.1023/A:1022643204877

[19] Boriah, S., Chandola, V., & Kumar, V. (2010). An evaluation of anomaly detection techniques for time series data. *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1–10.

[20] Hosmer, D. W., Lemeshow, S., & Sturdivant, R. X. (2013). Applied logistic regression (3rd ed.). Wiley. https://doi.org/10.1002/9781118548387

[21] Nguyen, T. N., Ngo, Q. H., & Kim, K. (2020). Credit card fraud detection: A hybrid approach using deep learning and logistic regression. *Journal of Ambient Intelligence and Humanized Computing*, 11, 5709–5720. https://doi.org/10.1007/s12652-020-01821-4

[22] Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5–32. https://doi.org/10.1023/A:1010933404324

[23] Zhou, Z.-H. (2012). *Ensemble methods: Foundations and algorithms*. CRC Press.

[24] Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794. https://doi.org/10.1145/2939672.2939785

[25] Li, Z., Liu, D., & Wang, W. (2020). Quantum machine learning for data classification based on XGBoost. *Quantum Information Processing*, 19(9), 309. https://doi.org/10.1007/s11128-020-02808-z