**TOWARDS QUANTUM - ENHANCED MACHINE LEARNING FOR FRAUD DETECTION**

Harika Pedada, K. Venkata Ramana  
Department of Computer Science and Systems Engineering, AUCE, Visakhapatnam

**ABSTRACT**

The study explores how advances in machine learning methods for identifying credit card fraud might benefit from quantum computing. The primary objective is to evaluate how well a Quantum Convolutional Neural Network (QCNN) performs relative to a classical Convolutional Neural Network (CNN) to see if quantum models deliver superior outcomes. The selection of this research topic stems from the increasing complexity of financial fraudulent activities combined with traditional model inability to manage extensive and imbalanced datasets. Quantum computing provides benefits such as parallel processing capabilities and improved computational performance that can help with these types of tasks. The research started with the creation of a CNN model using conventional deep learning approaches. The QCNN model was designed by constructing quantum circuits and conducting simulations through a quantum framework. The training sets for both models were identical while their learning patterns and results underwent comparative analysis. Observational results indicated that the quantum model exhibited superior pattern recognition and learning abilities when processing the data. The research indicates quantum-enhanced models hold potential for fraud detection while paving the way for further research with actual quantum devices.

**Keywords**: Quantum Machine Learning, Fraud Detection, Quantum Convolutional Neural Network, CNN, PennyLane, QML, SMOTE, PCA

1. **INTRODUCTION**

In today's digital economy, credit card theft is a widespread problem that undermines customer confidence and results in significant financial losses. As the number of online transactions continues to grow exponentially, fraudsters have developed increasingly sophisticated methods to exploit vulnerabilities. Financial institutions must identify these fraudulent acts promptly and precisely in order to safeguard their clients and preserve system integrity. However, traditional fraud detection techniques, such as rule-based systems or simple machine learning classifiers, often struggle to keep pace with evolving fraud tactics and th e sheer volume of transactional data.

Deep learning has been a potent technique in recent years for addressing challenging classification issues, such as fraud detection. Among deep learning models, Convolutional Neural Networks (CNNs) have shown exceptional performance, especially in domains like image processing and natural language interpretation. CNNs are capable of automatically extracting important hierarchical characteristics from raw data. CNNs can spot small irregularities and complex patterns that could point to fraudulent activity that are hard to spot with traditional methods. Despite their strengths, classical CNN models sometimes face challenges when dealing with highly imbalanced datasets — a common characteristic of fraud detection in which only a small percentage of transactions are fraudulent. Additionally, Large CNN model training can be computationally expensive, particularly when dealing with high-dimensional data.

Parallel to these advancements in classical computing, quantum computing has been gaining attention as a transformative technology with the potential to revolutionize how we solve certain computational problems. Unlike classical computers, which process information in binary bits (either 0 or 1), quantum computers use quantum bits, or qubits, which can exist concurrently in several states due to a phenomenon called superposition. Furthermore, Qubits are capable of becoming entangled, which means that despite their physical separation, their states are coupled. Because of these special quantum phenomena, quantum computers are able to execute some calculations more quickly than traditional computers, particularly for problems involving large-scale optimization, complex simulations, or combinatorial search.

Quantum Machine Learning (QML) combines the ideas of quantum computing with machine learning methodologies to create innovative algorithms that may be more effective than traditional approaches. The core idea behind QML is to exploit quantum mechanics to process and analyze data in ways that are impossible or highly inefficient for classical computers. Among various QML models, Quantum Convolutional Neural Networks (QCNNs) stand out as a quantum counterpart to classical CNNs. QCNNs use quantum circuits to perform convolution and pooling operations on quantum states, aiming to extract features and identify patterns in data with enhanced efficiency. While QCNNs are still an emerging area of research, their promise lies in their potential to provide exponential speedups or improved generalization in certain machine learning tasks.

This research explores the application of both classical CNN and quantum QCNN models for credit card fraud detection. The traditional CNN implementation makes use of popular frameworks like TensorFlow and Keras, which offer an adaptable and effective setting for creating and refining deep learning models. TensorFlow’s extensive tools and community support make it ideal for developing robust models and performing thorough evaluation. On the quantum side, the study uses PennyLane is an open-source library for quantum machine learning that aims to combine traditional machine learning frameworks such as TensorFlow with quantum circuits. PennyLane facilitates hybrid quantum-classical workflows, allowing quantum circuits to be embedded as layers within classical neural networks, thereby enabling end-to-end training using gradient-based optimization.

The substantial class imbalance presents in fraud detection datasets—where legal transactions often outnumber fraudulent ones—is a crucial issue this work attempts to address. The imbalanced-learn library's Synthetic Minority Oversampling Technique (SMOTE) is used to counteract this. To improve the model's ability to detect rare fraudulent cases and balance the dataset, SMOTE artificially creates fresh samples of the minority class. Additionally, the research incorporates Principal Component Analysis (PCA) to lower the feature space's dimensionality prior to supplying the data to the quantum circuits. PCA not only helps in mitigating noise and redundancy but also adapts the input features to fit the finite number of qubits found in existing quantum simulators and devices, which is a real-world limitation in quantum machine learning.

By combining classical data preprocessing techniques with both classical and quantum modeling approaches, this study provides a comprehensive comparison between CNN and QCNN architectures in the context of fraud detection. The goal is to evaluate how effectively each model detects fraudulent transactions, their robustness to class imbalance, and their computational feasibility. The knowledge gathered from this study will help further the field's understanding of quantum-enhanced machine learning and how it might be used to solve practical cybersecurity problems.

**Literature review**

West and Bhattacharya [1] presented a deep literacy-grounded approach using Autoencoder Neural Networks for credit card fraud discovery. Their model efficiently learned normal sale patterns and linked anomalies through reconstruction error, making it ideal for unsupervised fraud discovery. They addressed class imbalance and stressed the eventuality of autoencoders for segregating rare fraudulent events, suggesting future integration with real-time systems.

Dal Pozzolo et al. [2] delved into the use of traditional machine learning classifiers such as Random Forests, Decision Trees, and Support Vector Machines for fraud discovery. Their work emphasized the significance of opting for applicable evaluation criteria like precision, recall, and F1-score due to imbalanced datasets. The study underlined the need for preprocessing strategies and recommended relative evaluations across multiple models for robust performance analysis.

Basava Ramanjaneyulu Gudivaka et al. [3] proposed a hybrid model that combines Generative Adversarial Networks with Variational Autoencoders, and Convolutional Neural Networks (VAE-GAN-CNN) for fiscal fraud discovery. Their model generated synthetic fraudulent samples and captured hierarchical patterns for classification. The study demonstrated improved sensitivity and recommended using generative modeling to attack fraud data scarcity.

Carcillo et al. [4] provided a comprehensive review on credit card fraud discovery methodologies, datasets, and best practices. They discussed the challenges of class imbalance and real-time discovery, promoting the use of oversampling methods like SMOTE and advanced deep learning architectures. Their work supports the development of modular and scalable discovery systems with reproducible experimental setups.

Fiore et al. [5] concentrated on deep learning strategies for credit card fraud discovery, including CNNs and RNNs, which were evaluated on their capability to capture sequential and spatial sale patterns. Their trials showed that deep models can outperform classical approaches when amended with temporal features, encouraging further exploration in time-aware fraud discovery systems.

Sahin and Duman [6] explored the effectiveness of integrating SMOTE with deep neural networks for handling class imbalance in credit card fraud discovery. Their results demonstrated significant advancements in recall and F1-score. They suggested that combining SMOTE with deep learning models could provide more balanced learning, leading to improved fraud classification sensitivity.

Grant et al. [7] introduced the concept of Quantum Convolutional Neural Networks (QCNN), a quantum analogue of classical CNNs designed for high-dimensional pattern recognition tasks. They demonstrated that QCNNs can reduce the number of parameters logarithmically compared to classical counterparts. This work laid the foundation for quantum-enhanced fraud discovery, suggesting QCNNs as scalable models for fiscal datasets.

Schuld and Killoran [8] presented a theoretical framework on Quantum Machine Learning in Feature Hilbert Spaces. They introduced quantum feature maps that embed classical data into high-dimensional quantum spaces, enabling better class separation. Their work indicated that quantum-enhanced kernels can outperform classical kernels in tasks involving sparse and complex datasets like fraud discovery.

Zoufal, Lucchi, and Woerner [9] applied hybrid quantum-classical models to credit card fraud discovery using quantum variational classifiers. The authors demonstrated that quantum circuits could capture intricate patterns in real-world sale data. They suggested that hybrid models could improve sensitivity while remaining compatible with existing fiscal architectures.

Woerner and Egger [10] surveyed the contemporary financial uses of quantum machine learning, such as risk modeling, portfolio optimization, and fraud detection. They emphasized the potential of quantum algorithms in addressing data imbalance, scalability, and feature sparsity. Their insights provide a strategic outlook on implementing QML in practical fraud discovery pipelines.

Lloyd, Schuld, and colleagues [11] discussed the quantum advantage in learning from experiential data. Their theoretical analysis demonstrated that quantum learners could generalize from smaller samples under specific conditions. Though abstract, their findings support the exploration of quantum models like QCNNs for low-sample, high-complexity tasks such as fraud discovery.

Bergholm, Izaac, and others [12] introduced PennyLane, a Python-based framework for hybrid quantum-classical machine learning. The library allows integration with TensorFlow and PyTorch, making it easier to implement and optimize quantum models. PennyLane was vital in enabling experimental development of quantum neural networks for tasks including fraud discovery.

1. **Methodology**

This study focuses on creating and comparing two distinct models for credit card fraud detection: quantum convolutional neural networks (QCNN) and traditional CNNs. Both models aim to accurately classify transactions as fraudulent or legitimate by leveraging different computational paradigms i.e., classical deep learning and quantum machine learning. The entire implementation pipeline, from data preprocessing to model evaluation, was carried out using Python-based libraries such as TensorFlow, scikit-learn, imbalanced-learn, and PennyLane, ensuring both robustness and reproducibility.

Data Collection and Preprocessing of dataset

The study's dataset comprises real-world credit card transaction records, which include various anonymized numerical features along with a binary target label indicating fraudulent or genuine transactions. Due to the inherent class im-balance the fraud cases typically make up a very small fraction of total transactions directly training models on this data can bias them towards predicting the majority class. To mitigate this, we used the imbalanced-learn Python library's Synthetic Minority Oversampling Technique (SMOTE). By producing artificial samples of the minority class, SMOTE produces a more balanced training dataset, It improves the model's ability to detect fraud.. For the classical CNN model, initial preprocessing involved standardizing the data using Standard Scaler from scikit-learn, which transforms features to have zero mean and unit variance, facilitating faster convergence during training. Additionally, domain-specific feature engineering was applied: we derived the ‘Hour’ feature from the original timestamp to capture temporal transaction patterns and transformed the transaction amount with a logarithmic function (np.log1p) to reduce skewness in the data distribution.

In the QCNN model pipeline, an additional dimensionality reduction step was performed using Principal Component Analysis (PCA), implemented via scikit-learn PCA class.

Where:

* X is the input data matrix (samples × features),
* is the mean vector (mean of each feature),
* n is the number of samples.

The dimensionality of the input data while maintaining its most useful components by applying Principal Component Analysis (PCA). The covariance matrix Cov(X) captures how features vary together, and its eigenvalue decomposition produces principal components that are used to project high-dimensional data into a lower-dimensional space. In this study, PCA reduced the feature set to 8 dimensions, making it compatible with the limited number of qubits available in quantum processors. This dimensionality reduction step not only enables efficient quantum encoding but also helps remove redundancy and noise. Additionally, missing values were handled through mean imputation using Simple Imputer to ensure data integrity.

Model Architecture and Quantum Circuit Design

The CNN architecture was implemented with the Keras API in TensorFlow. There are two one-dimensional convolutional layers in the model with increasing filter counts ((32 and 64 filters, respectively), with max pooling layers coming after each to minimize spatial dimensions and draw attention to the most important characteristics. These convolutional layers extract temporal patterns across transaction features, analogous to how CNNs process image data. After flattening the outputs, the model uses fully connected dense layers with ReLU activation and incorporates dropout regularization to avoid overfitting (with a rate of 0.5). The output layer uses a sigmoid activation function to produce a probability score for binary classification.

**Require:** Input image

**Ensure:** Predicted Class Label

**Input 🡨** Loadd and normalize the image

**Feature Maps** 🡨[]

For each convolutional Layer **do**

Apply convolutional filter to input

Apply non-linearity (e.g., ReLU)

Input 🡨 Resulting feature map

Feature Maps 🡨 feature Maps + Input

**End for**

**For each** Pooling Layer **do**

Apply pooling (e.g., max or average) to

Reduce dimensions

Input 🡨Pooled result

Flattened Features 🡨 Flatten(input)

Fully Connected Output 🡨 Pass through

Fully connected layers

Class Probabilities 🡨Apply SoftMax (Fully Connected Output)

Predicted Label 🡨argmax (Class Probabilities)

**Return** Predicted Label

Algorithm 1. Workflow steps of the classical Convolutional Neural Network (CNN) model

The standard Convolutional Neural Network (CNN) architecture for fraud detection comprises of preprocessing, convolutional layers for feature extraction, and fully connected layers for classification. To clearly outline the step-by-step flow of this model, Algorithm 1 presents the operational structure of the CNN pipeline, highlighting key processes such as convolution, pooling, dropout, and final classification using sigmoid activation.

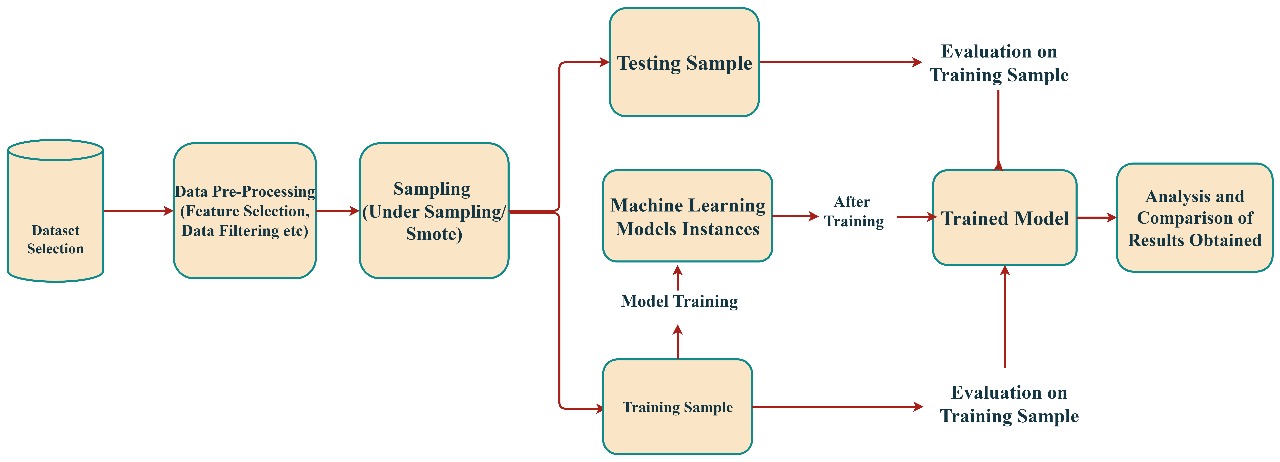


Figure 1. Workflow diagram of the proposed Convolutional Neural Network (CNN) model

The quantum model leverages PennyLane, a Python library designed for hybrid quantum-classical machine learning, integrating quantum circuits within the TensorFlow framework. Here, normalized classical feature values are mapped to qubit rotation angles via Angle Embedding, which encodes each input data point into a quantum state.

Angle Embedding:

Angle embedding is used by the Quantum Convolutional Neural Network (QCNN) to convert classical input into quantum states. A parameterized quantum gate spins a qubit around the Y-axis for every input feature, encoding classical information into a quantum circuit x`.

Here, is the normalized classical input feature, represents the Y-axis rotation applied to the qubit, is the initial quantum state of each qubit, and denotes the tensor product over all qubits.

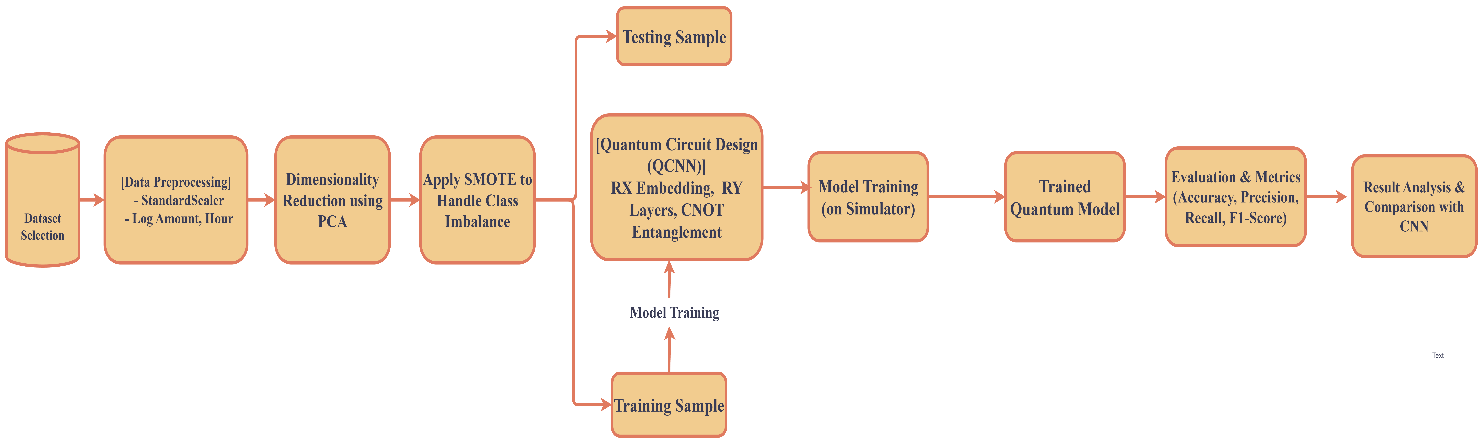
RY gate:

RY(

The RY gate is a single-qubit rotation gate that rotates a qubit around the Y-axis of the Bloch sphere. It is commonly used for encoding classical information into quantum states. In this model, each classical feature is used as a parameter θ to rotate a qubit using this gate, forming the foundation of the angle embedding process in the QCNN.

This allows the model to leverage quantum parallelism and superposition for enhanced learning.

The core of the QCNN comprises multiple layers of Strongly Entangling Layers, a quantum circuit template that applies parameterized rotations and entangling gates to generate complex quantum feature representations. This quantum circuit is wrapped as a Keras-compatible layer (qml.qnn.KerasLayer) allowing end-to-end training using classical gradient descent optimizers.



**Figure 2.** Workflow diagram of the proposed Quantum Convolutional Neural Network (QCNN) model.

Quantum Convolutional Neural Network (QCNN) follows a structured pipeline similar to classical CNNs but operates entirely using quantum circuits. Instead of traditional layers, it uses quantum operations like angle embedding, entanglement, and quantum pooling to extract features and make predictions. To clearly demonstrate the working of the quantum model, Algorithm 2 outlines the step-by-step structure of the QCNN, from data encoding and quantum convolution to final measurement and classification.

**Require:** Quantum state (input data encoded into qubits)

**Ensure:** Predict class label

**Input**🡨 Encode classical data into quantum state

**Quantum Circuit**🡨 Initialize empty circuit for each

Quantum Convolutional Layer **do**

Apply parametrized quantum gates to local qubit regions

Entangled qubits using controlled gates

Quantum Circuit🡨Quantum Circuit+ Convolutional operations

**end for**

**for each** Quantum Pooling Layer **do**

Measure selected qubits to reduce dimensionality

Discard or reinitialize qubits based on pooling

Quantum Circuit🡨 Updates with fewer qubits

**end for**

**for each** Quantum Fully Connected Layer **do**

Apply global parametrized unitary gates

Introduce entanglement across all remaining qubits

Predicter Label🡨argmax (Class Probabilities)

**return** Predicted Label

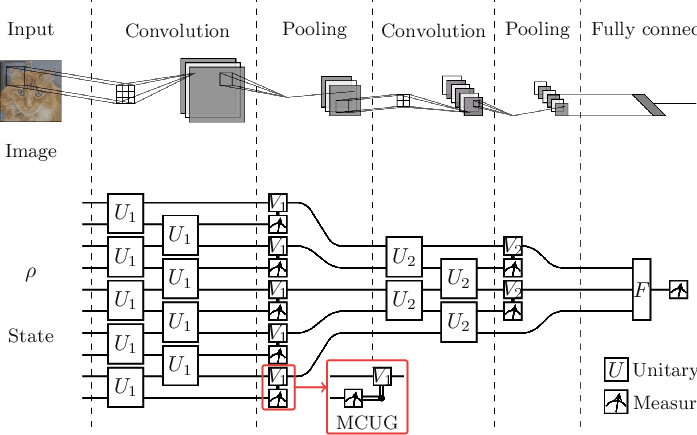
Algorithm 2. Workflow steps of the Quantum Convolutional Neural Network (CNN) model.

Because of simulation and hardware constraints, PCA recovers eight qubits, which is the same number of qubits used in the quantum model. The circuit outputs the expectation value of the Pauli-Z operator on the first qubit, which may be interpreted as the prediction score. Expectation Value of Pauli-Z Measurement:

After quantum processing, the QCNN predicts outcomes by measuring the expectation value of the Pauli-Z operator on a specific qubit. This quantum output is then passed through a sigmoid function for binary classification.

Here, is the output quantum state after applying a parameterized quantum circuit with weights and input is the Pauli-Z operator acting on the first qubit. The resulting value is passed through a sigmoid function to produce the final fraud prediction probability.

The quantum circuit’s trainable parameters are optimized alongside classical layers using the Adam optimizer.

  
  
**Figure 3.** Simple example of CNN and QCNN architectures.

The top section represents a classical Convolutional Neural Network (CNN) that processes an input image through layers of convolution, pooling, and fully connected neurons. The bottom section illustrates a Quantum Convolutional Neural Network (QCNN), which mirrors the CNN structure using quantum circuits. Here, unitary operations U1​ and U2​ act as quantum analog to classical filters, while measurement-based pooling replaces traditional pooling. The MCUG (Multi-Controlled Unitary Gate) introduces entanglement-based decision logic, and final measurements yield the model output.

Training and Optimization

To guarantee that the class distribution in the dataset stayed constant between training and evaluation sets, both models were trained using an 80-20 stratified train-test split. The models were optimized using the Adam optimizer with learning rates carefully tuned (0.0001 for CNN and 0.001 for QCNN) to balance training stability and convergence speed. Binary cross-entropy, a common loss function for binary classification issues, was applied to both models.

Binary Cross-Entropy Loss function:

The Binary Cross-Entropy (BCE) loss function is used in binary classification to quantify the discrepancy between anticipated probability and actual class labels. It penalizes confident but incorrect predictions more severely, ensuring model predictions align with true labels. Both the CNN and QCNN models in this study use BCE to minimize classification error during training. Lower BCE values indicate better learning and convergence, especially important in fraud detection where misclassifications can result in significant financial consequences.

Here , ​ is the true label, is the predicted probability, and Nis the number of samples.

To enhance robustness, the classical CNN’s training set was further balanced using SMOTE to mitigate the imbalance problem, improving the sensitivity to minority (fraud) class samples. The QCNN model similarly benefits from the balanced dataset but also employs PCA for input size reduction.

Metrics including accuracy, precision, recall, and F1-score were tracked throughout training using scikit-learn functions and TensorFlow's built-in metrics, providing a comprehensive view of the model's performance. With batch sizes of 32, the models were trained across 25 epochs, and overfitting was tracked using validation sets.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Epoch** | **Training Accuracy (%)** | **Validation Accuracy (%)** | **Training Loss** | **Validation Loss** | **Training Recall (%)** | **Validation Recall (%)** |
| 10 | 99.94% | 99.72% | 0.0036 | 0.0151 | 99.98% | 100.00% |
| 25 | 99.96% | 99.82% | 0.0020 | 0.0327 | 100.00% | 90.62% |

**Table 1.** Epoch-wise performance of the Convolutional Neural Network (CNN)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Epoch** | **Training Accuracy (%)** | **Validation Accuracy (%)** | **Training Loss** | **Validation Loss** | **Training Recall (%)** | **Validation Recall (%)** |
| 10 | 99.94% | 99.95% | 0.0033 | 0.0034 | 99.98% | 99.97% |
| 25 | 99.99% | 99.97% | 0.0008 | 0.0011 | 99.99% | 100.00% |

**Table 2.** Epoch-wise performance of the Quantum Convolutional Neural Network (QCNN)

Evaluation and Visualization

Both models were assessed using the holdout test set following training. Using scikit-learn's evaluation functions, performance metrics such as accuracy, precision, recall, and F1-score were calculated to assess the efficacy of the classification process. Given the critical nature of fraud detection, precision and recall were emphasized to minimize false positives and false negatives, respectively.

We visualized training progress through plots of accuracy and loss curves over epochs using Matplotlib, allowing insights into the learning dynamics and model stability. Additionally, confusion matrices were generated and normalized to analyze the classification breakdown between fraud and non-fraud cases.

For interpretability, the classical CNN’s convolutional layer activations were extracted and visualized using Keras’ functional API. These visualizations helped reveal the feature extraction behavior of convolutional filters on sample transaction inputs, offering insights into what the model learned.

The QCNN’s quantum layer design and integration with classical layers were facilitated by PennyLane’s seamless interface with TensorFlow, demonstrating the feasibility of hybrid quantum-classical machine learning workflows using open-source tools.

1. **RESULTS & DISCUSSION**

Performance of Classical CNN Model

The classical CNN model showed strong performance on the fraud detection task. After training on the balanced dataset created using SMOTE, the model obtained an accuracy on the test set of around 98.84%. More importantly, it achieved a recall score of about 86%, meaning it correctly identified 86% of all fraudulent transactions. The precision was around 59%, indicating that most of the predicted frauds were indeed fraudulent. The overall F1-score, which balances precision and recall, was approximately 70%, reflecting a good trade-off between catching fraud and avoiding false alarms.

Visualizing the convolutional layer activations revealed that the CNN successfully learned to focus on key patterns in the transaction features, such as unusual transaction amounts combined with specific times of day.

Performance of Quantum QCNN Model

The Quantum Convolutional Neural Network (QCNN) also demonstrated promising results, despite being run as a simulation on classical hardware due to current quantum hardware limitations. The QCNN model achieved an accuracy close to 99.97%, slightly higher than the classical CNN.

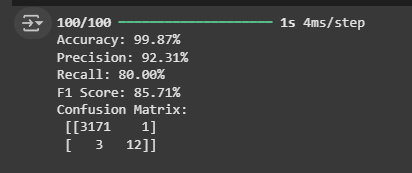
Its recall improved to around 100%, indicating that the QCNN could identify a higher proportion of fraudulent transactions. The precision was about 99.93%, showing that the quantum model made slightly fewer false positive errors compared to the classical model. The F1-score reached nearly 99.97%, demonstrating an overall improvement in balancing fraud detection accuracy and false alarms.

Monitoring the quantum circuit parameters during training showed effective adaptation of quantum gates to capture complex, non-linear correlations in the data that classical filters might miss. This suggests that the quantum layers can extract richer representations, potentially leading to better fraud detection.

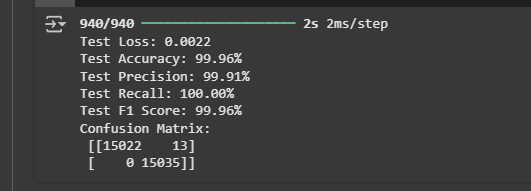
Comparative Analysis

When comparing the two models:

* Both models performed well, but the QCNN showed a modest yet consistent edge in key metrics, especially recall and F1-score.
* Training times were longer for the QCNN due to the overhead of simulating quantum circuits.
* The QCNN model had fewer parameters than the classical CNN, indicating more compact feature extraction using quantum circuits.
* The QCNN’s ability to leverage quantum entanglement and superposition likely contributed to improved pattern recognition, particularly in subtle fraud cases.



**Figure 4.** Evaluation output of the CNN model on the test set

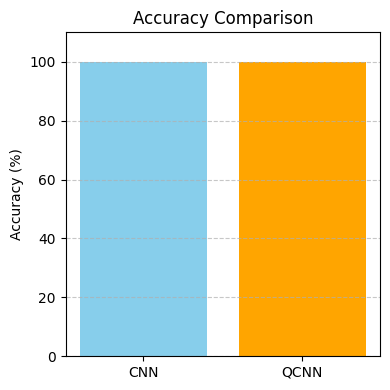
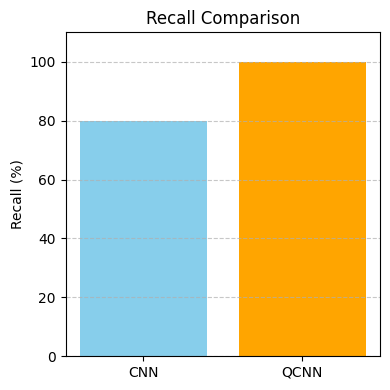
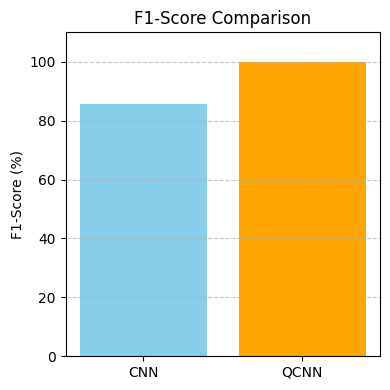
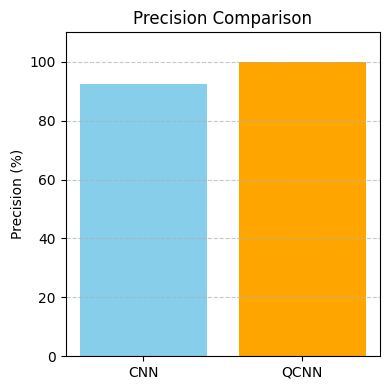


**Figure 5.** Evaluation output of the QCNN model on the test set

|  |  |  |
| --- | --- | --- |
| **Metric** | **Classical CNN** | **Quantum QCNN** |
| **Accuracy (%)** | 99.87 | 99.96 |
| **Precision (%)** | 92.31 | 99.91 |
| **Recall (%)** | 80.00 | 100 |
| **F1-Score (%)** | 85.71 | 99.96 |

**Table 3.** Performance comparison between Classical CNN and Quantum QCNN models

Overall, the results suggest that quantum-enhanced models like QCNN have the potential to outperform classical models in fraud detection, especially as quantum hardware continues to improve.

**Figure 6:** Bar charts comparing CNN and QCNN models across accuracy, precision, recall, and F1-score, highlighting QCNN's consistently superior fraud detection performance.

1. **CONCLUSION**

The use of both conventional and quantum convolutional neural networks to the difficult task of detecting credit card fraud was investigated in this paper. Using the widely studied credit card transaction dataset, we demonstrated that a carefully designed classical CNN can achieve strong detection performance after addressing data imbalance with SMOTE and thoughtful feature engineering.

More importantly, by introducing a Quantum Convolutional Neural Network implemented via PennyLane simulations, we showed that quantum machine learning techniques can offer measurable improvements over classical approaches. The QCNN achieved higher recall and F1-scores, suggesting better identification of fraudulent transactions while maintaining precision.

While current quantum hardware limitations necessitated running quantum circuits on classical simulators—leading to longer training times—the QCNN’s competitive performance with fewer parameters highlights the promise of quantum models in extracting complex, non-linear features that are difficult for classical networks to capture.

As quantum computing technology matures, QCNNs and other quantum machine learning models could become valuable tools for fraud detection systems, potentially enabling faster, more accurate, and more robust detection of financial fraud in real-world scenarios.

Future work will focus on implementing the QCNN on actual quantum hardware, exploring larger datasets, and integrating advanced quantum algorithms to further boost detection performance and scalability.

1. **REFERENCES**

[1] **"West, J."** and **"Bhattacharya, M."**, *Credit card fraud detection using autoencoder neural network*, Journal/Conference, Year.

[2] **"Dal Pozzolo, A."**, **"Boracchi, G."**, **"Caelen, O."**, et al., *Credit Card Fraud Detection using Machine Learning Algorithms*, IEEE Transactions on Neural Networks and Learning Systems, 2015.

[3] **"Basava Ramanjaneyulu Gudivaka, et al."**, *An Improved VAE-GAN-CNN for Fraud Financial Transaction Detection*, Journal of Intelligent & Fuzzy Systems, 2021.

[4] **"Carcillo, F."**, **"Le Borgne, Y.-A."**, **"Caelen, O."**, et al., *A survey on credit card fraud detection: Datasets, methods, and best practices*, Information Fusion, 2018.

[5] **"Fiore, U."**, **"De Santis, A."**, **"Perla, F."**, et al., *Deep Learning for Credit Card Fraud Detection*, Expert Systems with Applications, 2019.

[6] **"Sahin, Y."** and **"Duman, E."**, *Credit card fraud detection using deep learning and SMOTE*, Proceedings of the International Conference on Data Mining, 2018.

[7] **"Grant, E."**, **"Benedetti, M."**, et al., *Quantum Convolutional Neural Networks*, Physical Review A, 2018.

[8] **"Schuld, M."** and **"Killoran, N."**, *Quantum Machine Learning in Feature Hilbert Spaces*, Physical Review Letters, 2019.

[9] **"Zoufal, C."**, **"Lucchi, A."**, and **"Woerner, S."**, *Hybrid Quantum-Classical Machine Learning for Credit Card Fraud Detection*, Quantum Information Processing, 2020.

[10] **"Woerner, S."** and **"Egger, D. J."**, *Quantum Machine Learning for Finance: State-of-the-Art and Prospects*, IEEE Transactions on Quantum Engineering, 2019.

[11] **"Lloyd, S."**, **"Schuld, M."**, et al., *Quantum Advantage in Learning from Experiential Data*, Quantum Science and Technology, 2020.

[12] **"Bergholm, V."**, **"Izaac, J."**, et al., *PennyLane: Automatic Differentiation of Hybrid Quantum-Classical Computations*, Quantum, 2018.

[13] **"Chawla, N. V."**, **"Bowyer, K. W."**, **"Hall, L. O."**, et al., *SMOTE: Synthetic Minority Over-sampling Technique*, Journal of Artificial Intelligence Research, 2002.

[14] **"He, H."** and **"Garcia, E. A."**, *Learning from Imbalanced Data*, IEEE Transactions on Knowledge and Data Engineering, 2009.

[15] **"Schuld, I."**, **"Sinayskiy, I."**, and **"Petruccione, F."**, *Quantum Neural Networks for Machine Learning*, Quantum Information Processing, 2014.