Dutta et al. *EPJ Quantum Technology* (2025) 12:4
https://doi.org/10.1140/epjqt/s40507-024-00303-4

EPJ Quantum Technology
a SpringerOpen Journal

RESEARCH Open Access

Hybrid quantum neural networks: harnessing dressed quantum circuits for enhanced tsunami prediction via earthquake data fusion

Shivanya Shomir Dutta[1], Sahil Sandeep[1], Nandhini D[2,3] and Amutha S[1*]

*Correspondence:
amutha.s@vit.ac.in
[1]School of Computer Science and Engineering, Vellore Institute of Technology, Chennai, 600127, Tamil Nadu, India
Full list of author information is available at the end of the article

Abstract

Tsunami is one of the deadliest natural disasters which can occur, leading to great loss of life and property. This study focuses on predicting tsunamis, using earthquake dataset from the year 1995 to 2023. The research introduces the Hybrid Quantum Neural Network (HQNN), an innovative model that combines Neural Network (NN) architecture with Parameterized Quantum Circuits (PmQC) to tackle complex machine learning (ML) problems where deep learning (DL) models struggle, aiming for higher accuracy in prediction while maintaining a compact model size. The hybrid model's performance is compared with the classical model counterpart to investigate the quantum circuit's effectivity as a layer in a DL model. The model has been implemented using 2-6 features through Principle Component Analysis (PCA) method. HQNN's quantum circuit is a combination of Pennylane's embedding (Angle Embedding (AE) and Instantaneous Quantum Polynomial (IQP) Embedding) and layer circuits (Basic Entangler Layers (BEL), Random Layers (RL), and Strongly Entangling Layers (SEL)), along with the classical layers. Results show that the proposed model achieved high performance, with a maximum accuracy up to 96.03% using 4 features with the combination of AE and SEL, superior to the DL model. Future research could explore the scalability and diverse applications of HQNN, as well as its potential to address practical ML challenges.

Keywords: Hybrid quantum neural networks; Parameterized quantum circuits; Principal component analysis; Tsunami prediction; Earthquake data

1 Introduction

Quantum Machine Learning (QML) integrates quantum computing and ML to tackle challenges in both fields. The complexity of quantum information offers opportunities to enhance machine learning and deep learning models using quantum computing technique, offering greater accuracy with smaller amount of training data for prediction. Tsunamis, which are known to be one of the most catastrophic natural disasters leave in their way extensive loss of life, property damage, and economic upheaval near coastal regions. Displacement of survivors becomes inevitable, exacerbating the humanitarian cri-

sis. Environmental ecosystems suffer significant degradation, impacting biodiversity and ecological resilience. Hence, this study focuses on predicting tsunamis, a crucial task in disaster management, using the earthquake data over different regions. Since tsunamis are often caused by underwater earthquakes, particularly in subduction zones where tectonic plates converge, we have chosen the earthquake data for their prediction. Relying solely on the tsunami data for prediction is problematic due to the delay in data availability and insufficient coverage. Greater earthquake energy can shift vast amounts of water, which is a major contributing component to the creation of tsunamis. Tsunami waves travel slower than seismic waves, meaning data from buoys or tide gauges might indicate a tsunami only when it is already close to the shore, leaving little time for effective warnings. Furthermore, the network of tsunami detection buoys is not as comprehensive as seismic networks, leading to gaps in coverage. Initial detection of a potential tsunami relies on identifying the earthquake that might generate it, recording characteristics such as magnitude, depth and location, allowing for immediate analysis and warning. Hence, while tsunami data is essential for confirming and tracking the waves, earthquake data provides the crucial early warning needed for timely and effective tsunami prediction and response, allowing for timely evacuation, enhancing community preparedness, mitigating risks, and reducing losses, limiting property damage and potentially saving countless lives. The authors in [1] explore potential tsunami hazards associated with subaqueous landslides, that are majorly a result of earthquakes, in Lake Zurich, utilizing modeling techniques and historical data analysis to assess potential hazards and their implications for the local population. They emphasize that a tsunami hazard exists, particularly in the context of significant seismic events. Various methods exist to predict tsunamis based on earthquake data. Traditional approaches involve seismic networks that detect and analyze earthquake activity in real-time. These systems use algorithms to estimate the potential for tsunami generation based on the earthquake's magnitude, depth, and location. However, recent advancements have incorporated ML and DL techniques. ML models, such as Support Vector Machines (SVMs) and Random Forests (RF), have been used to predict the likelihood of tsunami occurrence by analyzing patterns in historical seismic data. DL methods, including Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), have further improved prediction accuracy by learning complex patterns and temporal dependencies from vast datasets. Li et al. [2] have used a RF model for early detection of Tsunamis. The RF model predicts tsunamis by evaluating various potential tsunami wave patterns. It analyzes tsunami wave amplitudes detected by S-net sensors along with earthquake data, including magnitude, epicenter latitude, and longitude. On the other hand, Willy et al. [3] have used a RNN, specifically the Gated Recurrent Unit (GRU) model for the early warning of tsunamis. To predict tsunamis accurately, the RNN model underwent a series of preprocessing steps. These steps included two-step feature scaling, the use of robust and min-max scalers to capture wave distribution, and applying a Butterworth filter with specific parameters to filter out ambient noise, ensuring the model's accuracy in identifying tsunami spikes. Also, Eugenio et al. [4] have used Regression Trees for tsunami predictions. This approach is evaluated using the 2003 M 6.8 Zemmouri-Boumerdes earthquake as a case study. This earthquake triggered a tsunami with observed wave heights ranging from centimeters to 3 meters, causing casualties and damages in the Mediterranean basin. The model aims to reliably estimate the maximum height of waves arriving at the coast in the event of a seismic occurrence. However, despite

these advancements, significant challenges remain in predicting tsunamis with high accuracy and reliability with limited data. One of the major research gaps identified is to predict tsunami from earthquake which has minimal data points and less features. This task is not significantly managed by the state-of-the-art models. Additionally, the variability in tsunami generation and propagation mechanisms makes it difficult to generalize predictions across different regions and tectonic settings. Thus, our research aims to address these gaps by exploring the use of HQNN in predicting Tsunamis. The study explores the potential acceleration of quantum computing in machine learning and aims to determine if a hybrid quantum neural network can outperform classical models in accuracy, model size, and training speed for binary classification tasks with high-dimensional datasets. The hypothesis suggests that such a hybrid network could notably enhance performance, with lesser amount of training data, where classical models struggle to achieve high accuracy. The hypothesis, in addition, is based on the concept that the hybrid quantum deep neural network could provide a significant advancement in the area where classical neural networks struggle to deliver excellent results in which the state-of-the-art models can't correctly have an accuracy of more than 90 percent. Hence, the hybrid quantum neural network offer advantages such as improved accuracy and reduced model size. In pursuit of the outlined research objectives, the methodology involves the design and implementation of a hybrid quantum neural network architecture built upon the foundation of state-of-the-art models. This architecture incorporates three distinct layers atop a quantum neural network, with subsequent experimental validation against classical models serving to establish performance standards. The primary research objectives include designing a hybrid quantum neural network and validating it against classical models. The study concludes by evaluating the utility of quantum deep learning and assessing the effectiveness of the hybrid quantum model using metrics like classification accuracy, model size, and training speed. HQML algorithms combine classical and quantum processing, offering new avenues for exploration. We analyze how different algorithm variations affect accuracy and robustness under various architectural variations. This comprehensive analysis is crucial for understanding the relationship between circuit architectures, their robustness, and their utility in QML.

2 Related works

QML integrates quantum computing with classical ML to improve the efficiency of conventional algorithms by utilizing quantum phenomena. QNNs are the quantum analog of classical NNs and are being investigated for their potential superiority over classical counterparts in practical applications. Our research demonstrates that the accuracy of the hybrid quantum neural network method can surpass that of its classical counterpart. In our hybrid model, we have implemented a combination of Pennylane's embedding (AE and IQP Embedding) and layers circuit (BEL, RL, and SEL), along with a Basic Neural Network model, with the combination of Angle Embedding and Random Layers delivering highest accuracy of 97.69% (with 4 features) surpassing all standard and state of the art models. Various papers that have been referred to for the preparation of this paper are summarized in the following Table 1. Makinoshima, Fumiyasu, et al. [5], using CNNs, demonstrated excellent performance in numerical tsunami forecasting experiments for the Tohoku region, with low errors in maximum tsunami amplitude and arrival time forecasting. Mulia, Iyan E., Aditya Riadi Gusman, and Kenji Satake [6] applied a novel approach to tsunami inundation forecasting by applying a DL algorithm to a database of

Table 1 Summary of Related Studies Using ML/DL Predictions

Literature	Model	Remarks	Input Dataset	Predicted Parameter
Makinoshima, Fumiyasu, et al. (2021) [5]	CNN	Proposed a model based on CNNs for early warning, demonstrating excellent performance in forecasting tsunami amplitude and arrival time.	Tsunami observation data from the 2011 Tohoku event	Tsunami amplitude and arrival time
Mulia, Iyan E., Aditya Riadi Gusman, and Kenji Satake (2020) [6]	DNN	Proposed a novel approach to tsunami inundation forecasting by applying a DL algorithm to a database of megathrust earthquakes.	532 source scenarios of interplate earthquakes along the Japan Trench Subduction Zone	Maximum tsunami elevation and flow depth
Listiani, Amalia, and Fuji Lestari (2023) [7]	ANN	An ANN model is used to predict tsunamis in shallow seas with an accuracy of 81%.	General Tsunami Parameters	Maximum inundation map - Oregon
Novianty, Astri, et al. (2022) [8]	ANN	Backpropagation ANN is used to predict tsunami potential in P-wave seismic signals, validated by K-fold cross-validation.	Seismic features - P-wave seismic signals	Rupture duration and P-wave dominant period
Siek, Michael, and Alfriyadi Rafles (2022) [9]	FCNN, CNN, RNN with LSTM	Research aimed to find appropriate ML algorithms for fast and accurate tsunami forecasts using FCNN, CNN, and RNN with LSTM.	Spatiotemporal data - Aceh, 26 December, 2004	Wave height, intensity, and propagation
Gálvez, Juan Francisco Rodríguez, et al. (2023) [10]	MLP, NN	Proposed NN and MLPs to predict maximum height and arrival time of tsunamis.	Tsunamis near Horseshoe fault in Northeastern Atlantic	Tsunami amplitude and arrival time
Andraud, Pierre, et al. (2023) [11]	MLP, CNN	Utilized deep learning techniques, specifically MLPs and CNNs, to predict tsunami impact parameters in real-time.	Seismic scenarios - CENALT fault database	Maximum water heights, runups, retreats, and currents
Korolev, Yury (2021) [12]	Reciprocity principle	Introduces a new method based on the reciprocity principle in forecasting the arrival of tsunamis.	ETOPO2 global dataset, DART stations, and deep-ocean station data	Arrival time of tsunamis
Xu, Hang, and Huan Wu (2023) [13]	LSTM, Bi-LSTM, CNN-LSTM	Developed tsunami prediction models utilizing DL techniques for forecasting tsunami waves, evaluated against numerical models and observational data.	NOAA data	Tsunami wave height
Sebastianelli, Alessandro, et al. (2021) [14]	HQCNN	Proposed a circuit-based HQCNN for remote sensing image classification, applied to LULC classification.	EuroSAT dataset	Land cover types - urban areas, forests, farmland, etc.
Paquet, Eric, and Farzan Soleymani (2022) [15]	HDQNN	Proposed a system based on HDQNN for financial prediction, including a DQN to predict future density matrix and a classical network to extract maximum future price of a security.	Financial time series data for 24 securities	Density matrix at a later time, maximum price reached by a security
Sagingalieva, Asel, et al. (2023) [16]	HQNN	Proposed HQNN for drug response prediction using convolutional, graph convolutional, and DQN layers of 8 qubits with 363 layers.	Genomics of Drug Sensitivity in Cancer dataset	IC50 drug effectiveness values
Jeong, Seon-Geun, et al. (2023) [17]	HQCNN	Proposed HQCNN as a novel method over classical CNN for UWB signal classification.	UWB Channel Impulse Response (CIR) data	Signal strength variations, LOS/NLOS distinctions
Heidari, Hanif, and Gerhard Hellstern (2022) [18]	HQNN, HRFQNN	Proposed quantum machine learning methods, HQNN and HRFQNN, for early detection of heart disease.	Cleveland dataset and Statlog dataset	Presence of heart disease

megathrust earthquakes. The study explores the use of a Deep Neural Network (DNN) to estimate high-resolution tsunami inundation from low-resolution simulation results, allowing for rapid and accurate predictions. Listiani, Amalia, and Fuji Lestari [7] also used an Artificial Neural Network (ANN) model to predict tsunamis with an accuracy of 81%. The ANN model utilized a Multi-Layer Perceptron with 5 input parameters, 2 hidden layers, and 1 output. Using similar methodology, Novianty, Astri, et al. [8] also implemented a backpropagation ANN to predict the tsunami potential in P-wave seismic signals and validated the accuracy result by K-fold cross-validation. Another research by Siek, Michael, and Alfriyadi Rafles [9] aimed to find an appropriate machine learning algorithm for fast and accurate tsunami forecasts. Three ML models, namely Fully Connected Neural Network (FCNN), CNN and RNN with Long Short-Term Memory (LSTM) were utilized and compared. In a set of papers, by Gálvez, Juan Francisco Rodríguez, et al. [10] and Andraud, Pierre, et al. [11], the authors using Multi-Layer Perceptrons (MLPs) with NN and CNN, respectively, predicted the maximum height and arrival time, and the real-time impact parameters of tsunamis. Another paper by Korolev, Yury [12] combines numerical experiments, data assimilation, software computations, and mathematical algorithms to develop a new method for short- term tsunami forecasting based on the reciprocity principle. In the same way, the authors have also implemented HQNN for various predictive/detective tasks. Xu, Hang, and Huan Wu [13] proposed novel tsunami prediction models using DL techniques for forecasting tsunami waves. The LSTM model and its variants were compared against numerical models and observational data to develop an early warning system for tsunamis. Sebastianelli, Alessandro, et al. [14] utilized a circuit-based Hybrid Quantum Convolutional Neural Network (HQCNN) for remote sensing image classification, which was used in land use and land cover (LULC) classification. Paquet, Eric, and Farzan Soleymani [15] developed a HQNN for predicting financial trends in the coming years. It includes an encoder that converts partitioned financial time series into a sequence of density matrices, a Deep Quantum Network (DQN) that predicts the future density matrix, and a classical network that extracts the maximum future price of a security from the output density matrix. Sagingalieva, Asel, et al. [16] proposed a model for effective drug response prediction, incorporating a deep quantum computing circuit within a ML framework to simultaneously analyze the cell line and chemical properties to predict their effects. This hybrid quantum architecture demonstrated a 15% improvement in prediction accuracy compared to its classical counterpart. Jeong, Seon-Geun, et al. [17] suggested another model based on HQCNN as a novel method over classical CNN in UWB signal classification. The proposed model for tsunami prediction using earthquake dataset is built upon similar method using various architectures for prediction tasks. Heidari, Hanif, and Gerhard Hellstern [18] proposed two quantum machine learning methods, HQNN and hybrid random forest quantum neural network (HRFQNN), for early detection of heart disease. However, the existing literature lacks concrete evidence about the superiority of QNNs over classical counterparts, especially in practical applications. The challenges faced during this task particularly included the complexity of integrating diverse data sources such as seismic data, oceanographic data, and geographical data. It emphasizes the need for effective data fusion techniques to improve the accuracy and reliability of tsunami forecasts. This paper empirically demonstrates the greater capacity of HQNNs over their classical counterparts in binary classification. The models' accuracy is used as a measure of capacity, and the results consistently show a quantum advantage,

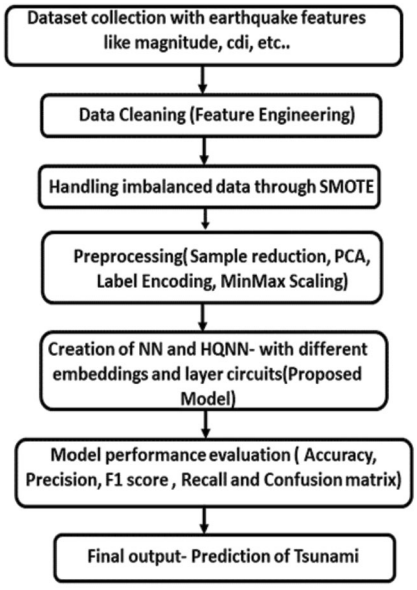


Figure 1 Flow chart of the present study representing the methodology

with HQNNs achieving better accuracy. This study serves as a foundational step towards further exploring the quantum advantage of HQNNs in real-world applications.

3 Methodology

Figure 1 is the flowchart demonstrating the methodology of our study.

3.1 Dataset description

The earthquake dataset, acquired from Kaggle [19], consists of an imbalanced dataset spanning from January 1, 1995, to January 1, 2023, originally comprises of 1000 data points with 675 data points of no labels for Tsunami and 325 data pints with yes labels for tsunami. A description of features has been provided in Table 2.

Figure 2 illustrates the correlation between earthquake magnitude and significance levels, differentiated by tsunami occurrence. Using data from the research dataset, the scatter plot shows that earthquakes triggering tsunamis (red points) generally have higher significance levels compared to those that do not (blue points). Both categories display a positive correlation between magnitude and impact, as evidenced by the upward trend lines. This indicates that higher magnitude earthquakes tend to have greater overall impacts, with tsunamis further amplifying their significance. Figure 3 presents the annual count of tremors from 1995 to 2023, comparing global occurrences with specific countries: Indonesia, Japan, Papua New Guinea, and a category for minor activity countries. The maroon line represents global tremors, consistently showing higher counts with peaks around 2010 and 2015. Indonesia (orange line) and Japan (red line) exhibit more fluctuations, with notable increases in activity post-2010. Minor activity countries (blue line) and Papua New Guinea (green line) show relatively low and stable tremor counts throughout the period. This comparison highlights the variability in seismic activity across different regions and emphasizes Indonesia and Japan's higher seismic activity within the analyzed timeframe.

Table 2 Feature Description

Sno.	Feature	Feature Description
1	Title	Title name given to the earthquake
2	Magnitude	The magnitude of the earthquake
3	Date_time	Date and time of the occurrence of earthquake
4	Cdi	The maximum reported intensity for the event range
5	Mmi	The maximum estimated instrumental intensity for the event
6	Alert	The alert level - "green", "yellow", "orange", and "red"
7	Tsunami	"1" for events in oceanic regions and "0" otherwise
8	Sig	A value indicating the significance of an event, where larger numbers denote greater significance. This value is determined by factors such as magnitude, maximum Modified Mercalli Intensity (MMI), felt reports, and estimated impact.
9	Net	The ID of a data contributor. Identifies the network considered to be the preferred source of information for this event.
10	Nst	The total number of seismic stations used to determine earthquake location.
11	Dmin	Horizontal distance from the epicenter to the nearest station
12	Gap	The largest azimuthal gap between adjacent stations (in degrees). Generally, a smaller azimuthal gap indicates a more reliable calculation of the earthquake's horizontal position. Earthquake locations with an azimuthal gap exceeding 180 degrees often have significant uncertainties in their location and depth.
13	MagType	The method or algorithm used to calculate the preferred magnitude for the event
14	Depth	The depth where the earthquake begins to rupture
15	Latitude/Longitude	A coordinate system used to determine and describe the position or location of any place on Earth's surface.
16	Location	Location within the country
17	Continent	Continent of the earthquake-hit country
18	Country	Affected country
19	Year	Year in which the earthquake had occurred

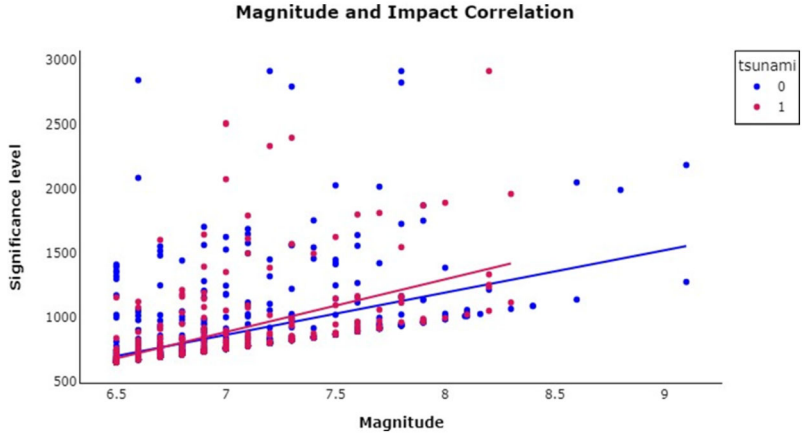


Figure 2 Earthquakes with and without Tsunamis - Impact Analysis

3.2 Feature engineering

Feature engineering is the process of transforming raw data into features that more effectively represent the underlying problem for predictive models, leading to improved accuracy on unseen data. It involves selecting, creating, or modifying features (variables) in the dataset to make them more suitable for machine learning algorithms. We particularly used feature engineering with feature selection. Feature selection focuses on identifying and retaining the most relevant features in the dataset while discarding redundant or irrelevant ones.

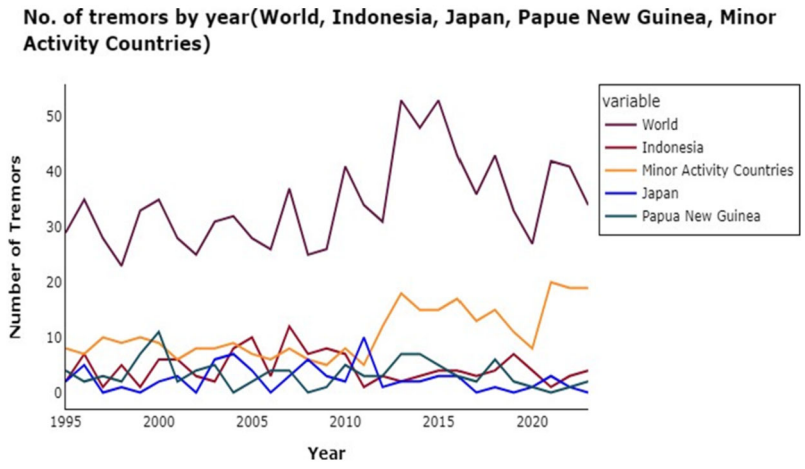


Figure 3 Number of tremors by year

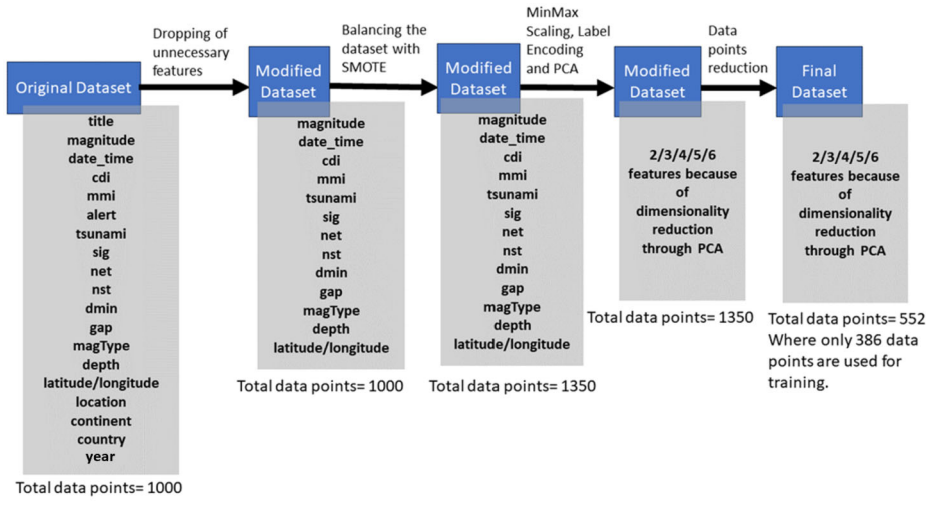


Figure 4 Illustration of feature engineering, handling imbalanced data and data preprocessing

This process involves analyzing the dataset to understand the importance and contribution of each feature to the predictive task. Techniques such as correlation analysis, feature importance, or domain expertise are employed to select features that have the greatest impact on the model's performance. By reducing the dimensionality of the feature space and retaining only the most informative features, feature selection helps improve model accuracy, reduce overfitting, and enhance interpretability. Hence we dropped the features like title, country(as we have latitude and longitude), continent, alert and location after performing exploratory data analysis (EDA), reducing the overall features to 14.

3.3 Handling imbalanced data

As our data with label 1 was less than the label 0, for predicting tsunami we balanced it using SMOTE (Synthetic Minority Over-sampling Technique). [20] or Synthetic Minority Over-sampling Technique, is a pivotal method uti-lized in machine learning to combat challenges posed by imbalanced datasets. In many real-world scenarios, such as fraud de-

Figure 5 Before Applying SMOTE

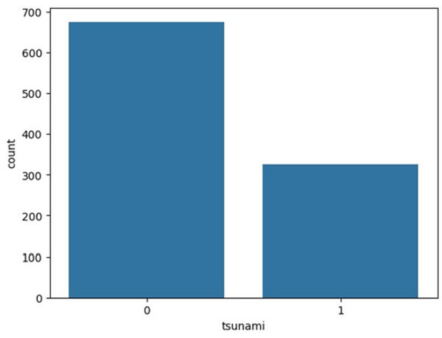
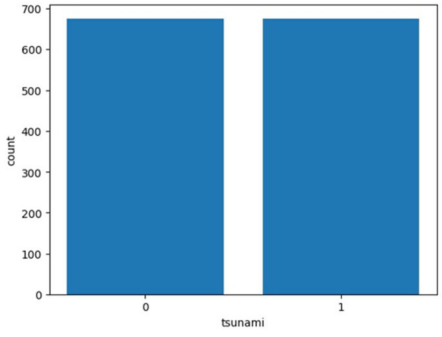


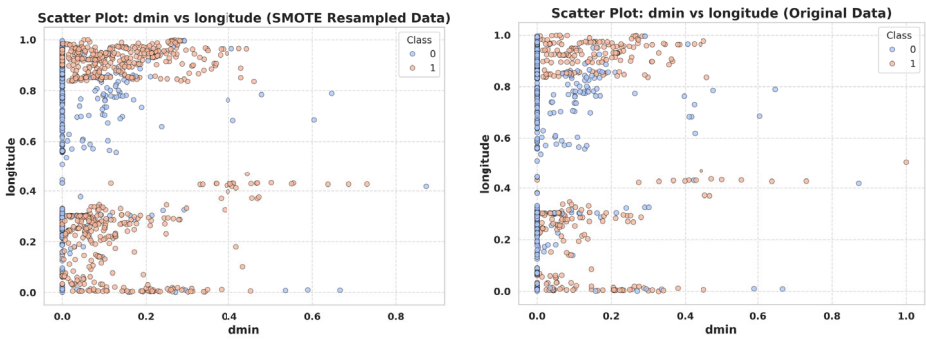
Figure 6 After Applying SMOTE



tection or medical diagnosis, one class is severely underrepresented compared to others. This skew can significantly impair the performance of traditional ML algorithms, as they tend to prioritize the majority class due to its prevalence. SMOTE ingeniously tackles this issue by synthetically generating new samples for the minority class, thereby rectifying the imbalance within the dataset. What sets SMOTE apart is its ability to create these synthetic samples not through mere duplication of existing minority class instances, but rather by interpolating between them in the feature space. This approach ensures that the synthetic samples retain the essential characteristics of the minority class, preserving the integrity of the data distribution. By strategically oversampling the minority class, SMOTE empowers machine learning models to better grasp the intricacies of these underrepresented in- stances, consequently enhancing their predictive performance, particularly in accurately identifying and classifying minority class instances. Thus Fig. 5 shows the counts of the labels for the prediction of Tsunmai-0 (No Tsunmai) and 1(Possibility of tsunami) before SMOTE and Fig. 6 shows the same after applying SMOTE.

After addressing the issue of class imbalance in the dataset using SMOTE, Random Forest was then used to rank feature importance, identifying earthquake magnitude, depth, and longitude as the top three influential features for tsunami generation. A detailed explanation of the validation process with the current dataset is provided below for clarity.

Random Forest ranks feature importance by evaluating how much each feature contributes to reducing the impurity in decision trees across the forest. When training each tree, the algorithm makes splits based on different features at each node. The quality of these splits is measured using a criterion such as Gini impurity or entropy, which quantifies how well the feature divides the data into distinct classes. Each time a feature is used



(a) Scatter plot for dmin vs longitude (Resampled Data).

(b) Scatter plot for dmin vs longitude (Original Data).

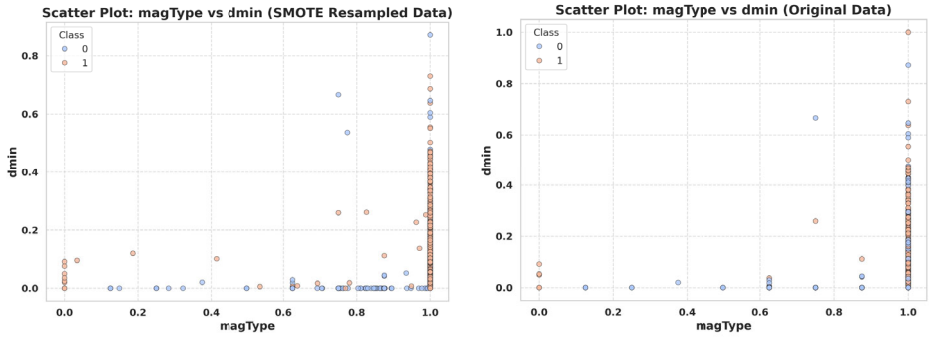
Figure 7 Comparison of dmin vs longitude in Resampled and Original Data

for splitting a node, the reduction in impurity is recorded. The more a feature helps reduce impurity, the more important it is considered.

The overall importance of a feature is determined by summing up the impurity reduction across all trees in the forest. This gives a measure of the feature's contribution to improving the model's ability to make accurate predictions as described by the varImp() method in Chen et al. [21]. The varImp() method is a widely used technique for determining variable importance in machine learning models, particularly for decision-tree-based models such as Random Forests. This method evaluates the contribution of each feature to the predictive performance of the model, typically by calculating metrics such as the mean decrease in accuracy or the mean decrease in impurity. The mean decrease in accuracy assesses the reduction in the model's accuracy when a specific feature is excluded, while the mean decrease in impurity measures the reduction in node impurity (e.g., Gini impurity or entropy) resulting from splits based on the feature across all trees in the forest. By ranking features according to their importance scores, the varimp() method helps identify the most influential features in the dataset.

Hence to validate the geophysical plausibility of the synthetic data, scatter plots were created for all pairwise combinations of these top 3 features (earthquake magnitude, depth, and longitude) ranked according to the feature importance by Random Forest, comparing patterns between the original and synthetic samples. The consistency in clustering patterns, feature distributions, and the spatial arrangement of data points across features, as shown in these plots, indicate that the inter-feature dependencies remain intact and that the synthetic data generated through SMOTE preserves meaningful and realistic relationships with real-dataset features, consistent with established seismological principles.

The scatter plots in Fig. 7 compare the relationship between dmin (minimum distance to the seismic event) and longitude for the original (Fig. 7 b) and SMOTE-resampled (Fig. 7 a) data. In both plots, distinct clusters for the two classes (Class 0 and Class 1) are observed, and the general distribution patterns are preserved. The SMOTE-resampled plot introduces additional synthetic points for the minority class (Class 1), filling sparse regions while maintaining geophysical trends visible in the original data. This validates that the synthetic data aligns with the natural relationship between dmin and longitude observed in the original dataset.



(a) Scatter plot for magType vs dmin (Resampled Data).

(b) Scatter plot for magType vs dmin (Original Data).

Figure 8 Comparison of magType vs dmin in Resampled and Original Data

(a) Scatter plot for magType vs longitude (Resampled Data).

(b) Scatter plot for magType vs longitude (Original Data).

Figure 9 Comparison of magType vs longitude in Resampled and Original Data

Similarly in Fig. 8 and 9, it is clearly visible that class 1 is resampled properly, maintaining geophysical trends visible in the original data. This validates that the synthetic data aligns with the natural relationship between magType vs dmin and magType vs longitude observed in the original dataset.

However, for dimensionality reduction of the dataset to reduce computational complexity of the proposed HQNN model as the number of qubits required increases with the dimensionality of the dataset as described later, we opted for Principal Component Analysis (PCA) instead of selecting specific features, as PCA considers all features and transforms them into a set of orthogonal principal components that capture the maximum variance. This approach ensures that no feature is entirely discarded, allowing the combined contribution of all features to be represented in the reduced-dimensional space.

3.4 Data pre-processing

In this proposed model of using HQNN in the prediction of Tsunami, the input data are the features of earthquakes occurring over different regions which is available in the form of a CSV file from Kaggle. The data couldn't be used directly hence all the null values or errors were eliminated. There were 19 (excluding the title) features pertaining to each earthquake

record. Then the unecessary features were dropped reducing the total number of features to 14. The 14 features included magnitude, date and time of occurrence, maximum reported intensity (cdi), maximum estimated instrumental intensity (mmI), indication of tsunami occurrence, significance level (sig), data contributor ID (net), number of seismic stations used (nst), horizontal distance from the epicenter to the nearest station (dmin), azimuthal gap between adjacent stations (gap), magnitude calculation method (magtype), rupture depth (depth), latitude, and longitude coordinates.

Figure 4 details the preprocessing steps applied to the dataset throughout the research. Starting with the Original Dataset containing 1000 data points and multiple features such as title, magnitude, and location details, unnecessary features were first dropped to create the Modified Dataset. This dataset, still with 1000 data points, retained key features like magnitude, date_time, and tsunami. Next, the dataset was balanced using SMOTE (Synthetic Minority Over-sampling Technique), increasing the total to 1350 data points. Following this, the data underwent MinMax scaling, label encoding, and PCA (Principal Component Analysis), preparing it for further dimensionality reduction. The resulting dataset maintained 1350 data points but was reduced to 2 to 6 features due to PCA. Finally, after reducing the number of data points for training purposes, the Final Dataset comprised 552 data points, with only 386 used for training, retaining 2 to 6 features as determined by PCA. The prerprocessing steps are further described in detail.

3.4.1 Min-max scaling

The normalization of features has been executed through Min-max scaling [22], following the pro- vided equation (1) This process ensures that all features are rescaled to fit within the range [0,1], thereby establishing that the minimum and maximum values of any feature or variable will be 0 and 1, respectively.

$$\text{Cscaled} = \frac{C - C_{\min}}{C_{\max} - C_{\min}} \tag{1}$$

where Cmax and Cmin represent the maximum and minimum values of the earthquake feature vectors respectively

3.4.2 Principal component analysis

As we intend to use HQNN for the prediction process, keeping the computation complexity in mind, we applied Principal component analysis (PCA) technique to choose the highly significant features among the 14 input features after feature engineering (before transforming them into quantum-state representations), making sure that the model was made to run on limited data with the highest possible accuracy. PCA [23] stands as a method for linearly reducing dimensionality, adept at identifying linear correlations within data. Commonly employed in data processing, PCA techniques serve the purpose of feature extraction. This dimensionality reduction was particularly critical for scaling the hybrid quantum-classical workflow, as discussed in [24, 25] and [26]. After feature engineering, the 14 features scaled to Cscaled, we condensed it to a 2D vector [C1, C2], highlighting two key features. Employing PCA, we iterated this process for 3D, 4D,5D and 6D vectors, ultimately extracting principal features [Ci]. These specifically chosen attributes via PCA were utilized as inputs for subsequent tsunami predictions.

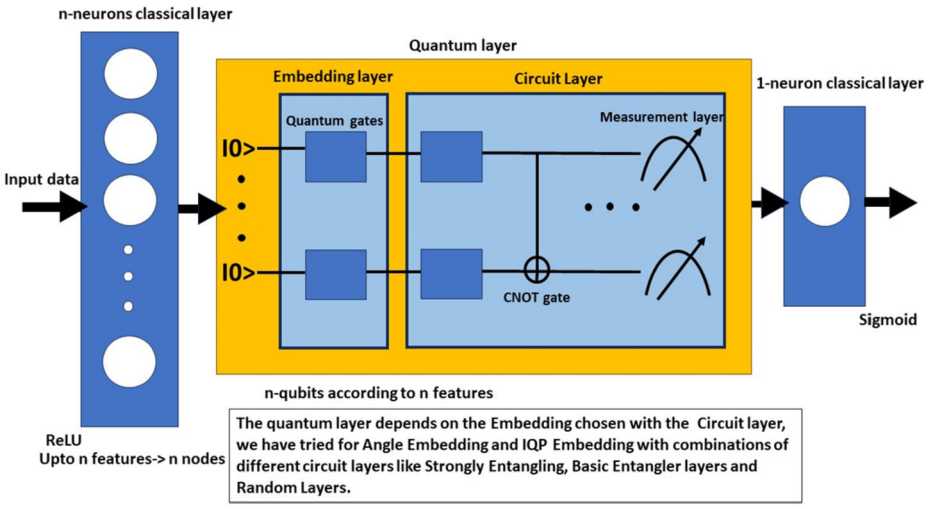


Figure 10 Detailed Model Architecture of the proposed model HQNN

3.4.3 Label encoding

Label Encoding [27] serves as a crucial pre-processing method in machine learning projects, as it transforms categorical columns into numerical formats, making them compatible with machine learning models that exclusively operate with numerical data. This conversion is essential for enabling these models to utilize the categorical information within the dataset effectively. Here, for pre-processing we have used sklearn's pre-processing module.

3.5 Model architecture

The architecture flow diagram of the proposed HQNN model is as illustrated in Fig. 10. The proposed model is a combination of deep learning layers and a quantum layer which has been made by defining a Qnode.

The input data is first fed into a n-neuron classical layer which has neurons equal to the number of features or PCA being tested. The layer after the input layer consists of the ReLU activation function.

Mathematically, the ReLU function [28] is defined as in equation (2).

$$\mathrm{ReLU}(x) = \max(0, x) \tag{2}$$

Where x is the input to the ReLU function. The function outputs 0 for any negative input and leaves positive inputs unchanged. Graphically, the ReLU function looks like a ramp, where the function is zero for all negative values of x and increases linearly with x for positive values.

The next layer is the quantum layer consisting of a Qnode The next layer in the model is a quantum layer comprising a QNode, which combines Pennylane's embedding method to encode classical input data into a quantum state and a parameterized circuit layer that adjusts parameters with each iteration. This QNode is converted into a TensorFlow-compatible quantum node to function within a Keras layer. The experiments in the study utilize six combinations of ansatz: Angle Embedding paired with Basic Entangler Layers,

Angle Embedding with Random Layers, Angle Embedding with Strongly Entangling Layers, IQP Embedding with Basic Entangler Layers, IQP Embedding with Random Layers, and IQP Embedding with Strongly Entangling Layers.

Angle Embedding encodes classical data by using the data points as rotation angles for single-qubit gates. IQP Embedding, on the other hand, represents features in qubits using diagonal gates from an IQP circuit. The Basic Entangler Layer consists of single-qubit rotations on each qubit, followed by CNOT gates that connect neighboring qubits. The Random Layer is formed by selecting single-qubit rotations and two-qubit entangling gates randomly. Inspired by the circuit-centric classifier approach, the Strongly Entangling Layer includes single-qubit rotations combined with entanglers. Additionally, a measurement layer using the Pauli-Z operator is incorporated to assess the expectation value of the observable along the z-axis of individual qubits, thereby providing insights into the quantum state.

The interplay between the measurement layer and the classical layer in a Quantum Neural Network (QNN) is crucial for its ability to handle complex tasks. The QNN first runs a quantum circuit, producing measurement outcomes by observing specific observables on the qubits with the Pauli-Z operator, represented as classical bits. These measurements are subsequently passed on to the classical layer for further processing.

The embedding techniques of Pennylane mentioned above and used for testing are as follows:

3.5.1 Angle embedding

Angle embedding involves representing classical data by directly encoding them as rotation angles of single-qubit gates.

Rotation gates: The rotation gate [29] performs rotation operations on single qubits. It uses a specified rotation angle θ to rotate the qubit around the x, y, or z axis. The different types of rotation gates are Rx, Ry, and Rz, each corresponding to rotations around the respective axes. For example, if the input value to the Rx gate is π, the qubit will rotate by π radians around the x-axis. A rotation gate can be expressed in 2X2 matrix [30] as in equation (3).

$$R(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \tag{3}$$

Where θ is the rotation angle.

In the context of QNNs, this typically involves applying rotations around different axes (RX, RY, RZ gates) based on the input data. The angles of these rotations serve as the parameters that encode the classical information into the quantum state.

3.5.2 IQP embedding

IQP (Instantaneous Quantum Polynomial) [29] embedding utilizes Hadamard gates to create superpositions and embeds data through rotations and entanglement within these superposed qubits. This method is capable of embedding n data points into n qubits.

Hadamard gate: It operates on a single qubit and changes the target qubit to a superposition state [29]. Superposition is a state where a qubit can simultaneously be 0 and 1 with certain probabilities, represented as a point on the Bloch sphere. Unlike classical bits

that can only be 0 or 1, a qubit can represent any value on this sphere, allowing a quantum computer with n qubits to represent 2^n values simultaneously. This capability enables quantum computers to perform many calculations at once, vastly increasing computation speed compared to classical computers.

The Hadamard gate can be represented [31] as in equation (4).

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{4}$$

Currently, quantum neural networks implemented with Pennylane can operate with up to 16 qubits, regardless of the embedding method. Embedding 16 data points requires 16 qubits, and since each qubit involves a Hadamard gate, the circuit becomes more time-consuming to run. While data embedding using superposition is possible, it poses challenges in reflecting high-dimensional data effectively. This limitation impacts the performance of quantum neural networks, as using low-dimensional data can degrade performance for complex tasks.

3.5.3 Basic entangler layer

The basic entangler layer introduces entanglement among qubits by applying a sequence of entangling gates, typically Controlled-NOT (CNOT) gates, across neighbouring qubits.

CNOT gate: The CNOT gate [29] operates on two qubits: a control qubit and a target qubit. If the control qubit is in the state 1, it flips the state of the target qubit (i.e., changes 0 to 1 or 1 to 0). If the control qubit is in the state 0, the state of the target qubit remains unchanged. Thus, the CNOT gate modifies the state of the target qubit based on the current state of the control qubit, while the control qubit itself retains its state after the operation.

The CNOT gate can be mathematically represented as 4X4 matrix [29] as equation (5).

$$\text{CNOT} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{5}$$

3.5.4 Strongly entangling layer

The strongly entangling layer enhances entanglement among qubits by applying a more complex pattern of entangling gates compared to the basic entangler layer.

In addition to the CNOT gates between neighbouring qubits, a CNOT gate is also applied between the last qubit and the first qubit, closing the loop and ensuring full connectivity.

3.5.5 Random layer

The random layer introduces randomness into the quantum circuit by applying a sequence of randomly chosen single-qubit rotations and entangling gate.

The Figs. 11-16 provided in this paper illustrate various quantum nodes designed for integration into an artificial neural network (ANN), each utilizing different embeddings and layer circuits. These visualizations demonstrate a single block of the circuit, but in practice, six blocks were used for enhanced performance. While the figures show 2 or 3

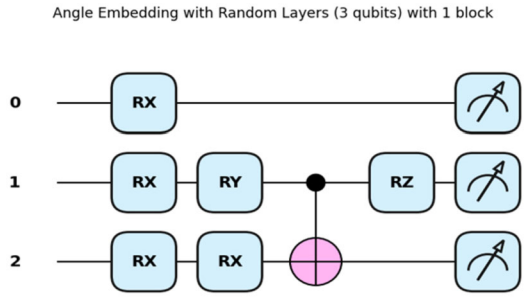


Figure 11 Quantum Node of AE with RL (3 qubits)

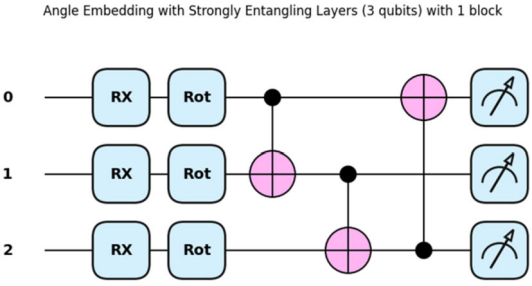


Figure 12 Quantum Node of AE with SEL (3 qubits)

qubits for demonstration purposes, our results indicate that using 4 qubits yields better outcomes. No optimization was performed to reduce the gate count in the quantum nodes, as this study utilized the standard embedding techniques and layer circuits provided by the PennyLane library and our primary focus was on demonstrating the quantum model's capabilities and performance. While these standard methods are effective for demonstrating the quantum model's capabilities, the lack of gate optimization could be considered as a limitation in terms of efficiency, particularly in larger-scale applications.

Figure 11 presents a visualization of a quantum node with three qubits, employing angle embedding with random layers. The circuit begins with rotation gates around the x-axis (RX) for each qubit. Qubit 0 only has an RX gate followed by a measurement. Qubit 1 undergoes an RX gate, followed by a rotation around the y-axis (RY), and then an RZ gate, before being measured. Qubit 2 has two consecutive RX gates before its measurement. Additionally, a Controlled-NOT (CNOT) gate creates entanglement between qubit 1 (control) and qubit 2 (target). All qubits are then measured. This sequence of gates forms one block of the quantum node. For integration into an ANN with six blocks, this sequence is repeated or extended with different parameters across multiple blocks. In this hybrid setup, classical data is embedded into the quantum circuit through the rotation angles, and the resulting measurements are used as inputs for subsequent layers in the ANN, combining quantum and classical computation to potentially enhance machine learning model performance.

Figure 12 shows a quantum node with angle embedding and strongly entangling layers for three qubits. Each qubit is first subjected to an RX gate, followed by an additional rotation (denoted as "Rot"), representing rotations around any axis or a combination of axes. Qubits are then interconnected through a series of CNOT gates, creating entangle-

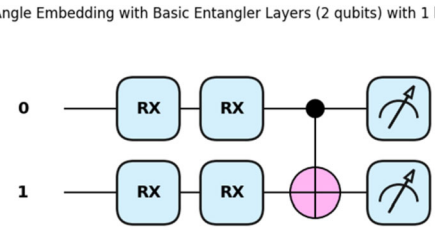


Figure 13 Quantum Node of AE with BEL (2 qubits)

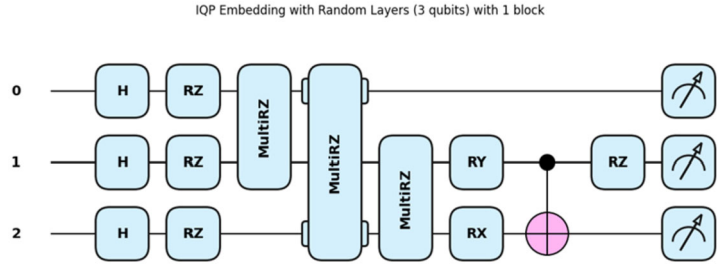


Figure 14 Quantum Node of IQP Embedding with RL (3 qubits)

ment between the qubits. Specifically, qubit 0 controls a CNOT gate with qubit 1 as the target, qubit 1 controls a CNOT gate with qubit 2 as the target, and another CNOT gate has qubit 0 as the control and qubit 2 as the target. Additionally, a controlled rotation gate applies further entanglement. After these operations, each qubit is measured. This circuit forms one block of the quantum node. When integrated into an ANN with six blocks, this structure is replicated or extended, with each block potentially having different parameters. Classical data is embedded into the quantum circuit through rotation angles, and the measurements are used as inputs for subsequent classical layers in the ANN. This hybrid approach leverages quantum mechanics' entanglement and superposition properties to enhance machine learning models' computational capabilities.

Figure 13 illustrates a quantum node designed for angle embedding with basic entangler layers, utilizing two qubits and forming one block. Each qubit undergoes two sequential RX gates to embed classical data into the quantum states by adjusting the rotation angles. After these rotations, a CNOT gate creates entanglement between the qubits, with qubit 0 as the control and qubit 1 as the target. Following the entanglement operation, each qubit is measured, collapsing the quantum states into classical bits. These measurements can then be used as outputs or features for subsequent ANN layers. In an ANN with multiple blocks, this circuit structure would be replicated or extended with different parameters for each block. By integrating quantum circuits as layers within classical neural networks, the model leverages quantum mechanics principles to enhance its computational power and potentially achieve better performance in machine learning tasks.

Figure 14 illustrates a quantum node employing IQP embedding with random layers. This quantum node uses a 3-qubit quantum circuit with one block to embed classical data into quantum states, enhancing data processing capabilities. The circuit begins with Hadamard gates applied to each qubit, creating superposition states that enable parallel processing power. RZ gates then rotate the qubits around the Z-axis, introducing phase

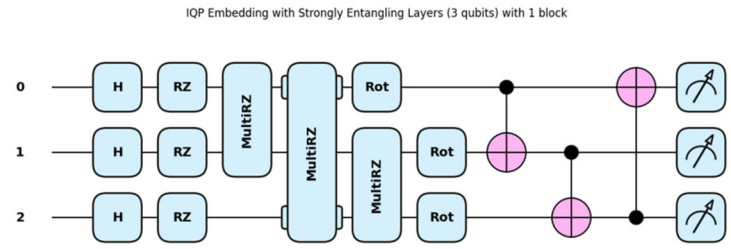


Figure 15 Quantum Node of IQP Embedding with SEL

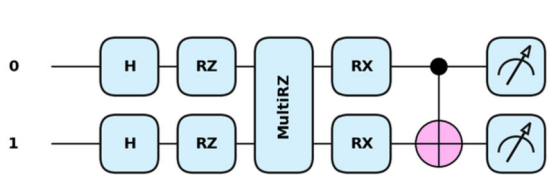


Figure 16 Quantum Node of IQP Embedding with BEL

shifts crucial for data embedding. MultiRZ gates entangle the qubits by applying simultaneous Z-axis rotations to all three, establishing complex inter-qubit relationships. An RX and an RY gate are applied to the second qubit in the subsequent layer, performing rotations around the X and Y axes. A CNOT gate with qubit 2 as the target introduces additional entanglement. Additional RZ gates refine phase relationships before measurements collapse the quantum states to classical binary outcomes, providing data integrated into the ANN. By embedding classical data into quantum states through these layered quantum operations, the quantum node enhances the ANN's ability to process complex, high-dimensional data, leveraging quantum computational advantages.

Figure 15 illustrates a quantum node employing IQP embedding with strongly entangling layers. This 3-qubit, one-block circuit processes and embeds classical data into quantum states. The circuit begins with Hadamard gates on each qubit, creating superpositions, followed by RZ gates introducing phase shifts. MultiRZ gates then apply simultaneous Z-axis rotations to entangle the qubits. Next, generic rotation gates (Rot) modify the qubit states. Multiple CNOT gates entangle the qubits further, flipping the state of the target qubit based on the control qubit's state. These operations culminate in measurements collapsing the quantum states into classical binary outcomes. Integrating this into an ANN aims to leverage quantum entanglement and superposition, enhancing the network's capability to handle complex, high-dimensional data for advanced machine learning tasks.

Figure 16 illustrates a quantum node employing IQP embedding with basic entangler layers. This quantum circuit utilizes 2 qubits with one block to transform classical data into quantum states. The process begins with Hadamard gates applied to each qubit, creating superpositions. Following this, RZ gates rotate the qubits around the Z-axis, introducing phase shifts. A MultiRZ gate entangles the qubits by applying simultaneous Z-axis rotations. RX gates are then applied, rotating the qubits around the X-axis. A CNOT gate, with the qubit 1 as the control and the qubit 2 as the target, introduces additional entanglement. Finally, measurements collapse the quantum states into classical binary outcomes.

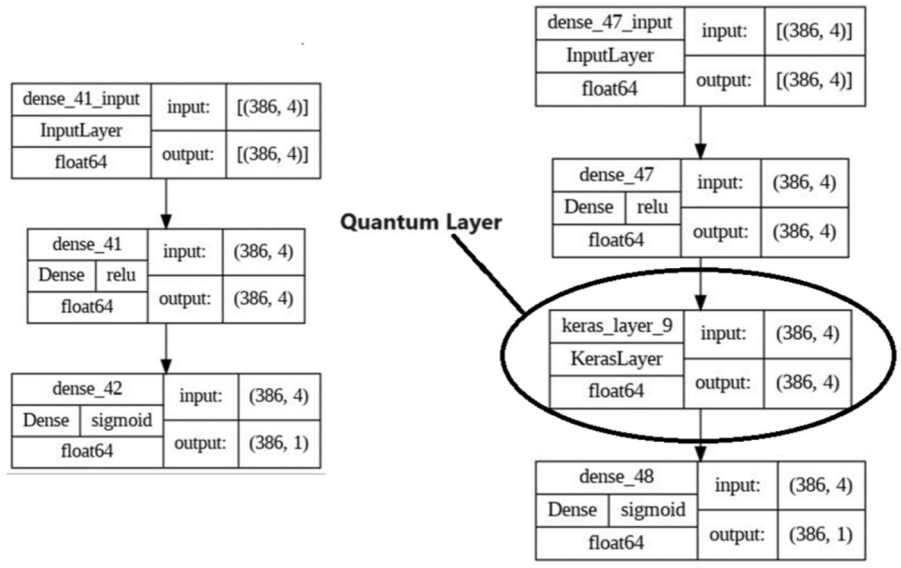


Figure 17 Classical model vs Proposed model

This embedded data is then integrated into the ANN, enhancing its ability to process complex and high-dimensional data by leveraging quantum superposition and entanglement properties.

These descriptions detail the various quantum nodes used in our paper, demonstrating different quantum embedding and layer circuits, which form the building blocks of our approach to integrating quantum layers into ANNs. While the figures show single blocks and fewer qubits for simplicity, our results indicate that using six blocks and four qubits yields superior performance.

Now we will discuss about the Measurement Layer, which has been added to any chosen layer circuit using the PauliZ operator. The measurement layer involves measuring certain observables, often represented by Pauli operators like PauliZ, to extract information from the quantum state for further processing or decision-making. The PauliZ gate is used here to measure the expectation value of individual qubits, providing valuable information about the quantum state during the computation process. Thus, the measurement layer gets connected to the next classical layer which is a 1 neuron layer with sigmoid activation function.

The integration of quantum layers into neural network architectures represents a groundbreaking advancement in computational efficiency and performance. Figure 17 contrasts a traditional neural network with a quantum-enhanced model. In the conventional architecture, the model consists of an input layer, a dense layer with ReLU activation, and an output layer with a sigmoid activation. The quantum-enhanced model introduces a significant innovation by incorporating a quantum layer between the dense layers. This 'KerasLayer' leverages quantum computing principles to maintain the input and output dimensions while potentially enhancing feature extraction and transformation capabilities. The quantum layer promises improved computational speed and accuracy, especially in handling complex data patterns (Fig. 17)

Mathematically, the sigmoid function is defined as in equation (6).

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{6}$$

Where x is the input to the sigmoid function and e is the base of the natural logarithm (approximately equal to 2.71828) [32]. The sigmoid function squashes the input value x into the range [0,1]. When x is large and positive, $e^{-}x$ approaches zero, making the denominator of the fraction close to 1, and consequently, the sigmoid output approaches 1. Conversely, when x is large and negative, $e^{-}x$ approaches infinity, making the denominator large, and the sigmoid output approaches 0.

3.5.6 Model training

The model comprises 6 blocks, where each block includes an Embedding layer followed by a layer circuit, facilitating complex quantum feature transformations. Each of these blocks is repeated with a single layer of embedding to enhance entanglement and data representation capacity. The training of this model employed a learning rate of 0.02, optimized using the Adam optimizer from TensorFlow, configured with $\beta 1$=0.9 and $\beta 2$=0.999, to ensure efficient convergence and mitigate the risk of vanishing gradients. We trained the model on batches of size 16, iterating over 25 epochs to allow the model sufficient opportunity to learn from the training data. The model was compiled with a binary cross-entropy loss function and the AUC metric, ensuring a robust evaluation of its performance in distinguishing between classes. The training history was captured by fitting the model on the training dataset, validating on a separate test dataset, and shuffling the data between epochs to enhance generalization.

The training time for the proposed 6-block HQNN is 15 minutes, whereas the classical neural network requires only 3 minutes for training. In this study, the quantum model is implemented on a classical computer equipped with an i5 processor and an Nvidia GTX GPU, using the PennyLane library, rather than on quantum simulators. The aim of this paper is to prove quantum advantage using the Pennylane library over the classical models. While quantum simulation with Pennylane becomes exponentially more expensive with an increasing number of qubits, potentially limiting scalability for larger datasets or real-time predictions, the future of quantum computing offers promising solutions. Real-time predictions, particularly in the context of inference with pre-trained models, are not significantly affected by the training time, as the model is already trained and only requires efficient inference. Furthermore, the main goal of this study is to demonstrate that quantum models can effectively learn from small datasets with limited features, outperforming classical methods. While larger datasets may result in longer training times, the scalability challenge can be mitigated as quantum computing hardware evolves, with more qubits and faster processing available. Besides this study has shown that even smaller-scale quantum computations can achieve significant speed-ups for specific tasks. Therefore, as quantum technology continues to advance, quantum models are expected to overcome current limitations and offer substantial advantages, particularly in real-time applications and problems that leverage quantum speed-ups.

3.5.7 Evaluation of model

Metrics from Eqs. (7)–(10) such as precision, recall, accuracy, and F1-score were used to evaluate the model's performance. Below TP stands for True Positives, TN stands for True

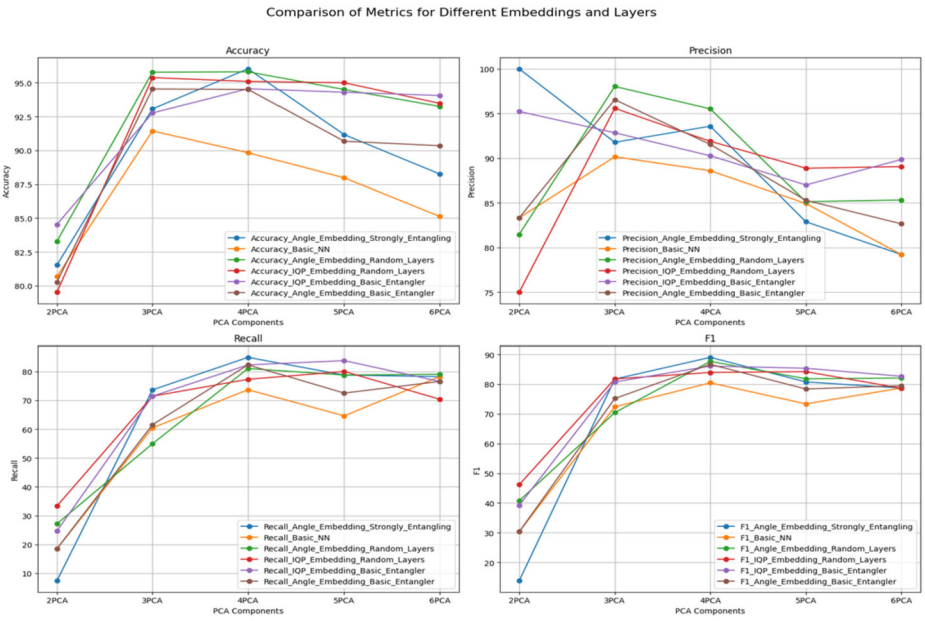


Figure 18 Comparison of metrics

Negatives, FN for False Negatives and FP stands for False Positives.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{7}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{8}$$

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{FP} + \text{FN} + \text{TP} + \text{TN}} \tag{9}$$

$$\text{F1 Score} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{10}$$

4 Summary and detailed analysis

The results of our study present a comprehensive performance comparison of several quantum-classical hybrid models and a basic neural network model across different levels of dimensionality reduction using PCA. The models evaluated include AE with SEL, a Basic Neural Network, AE with RL, IQP Embedding with RL, IQP Embedding with BEL, AE with BEL, and IQP Embedding with SEL. Performance metrics such as Accuracy, Precision, Recall, and F1 Score were recorded for PCA dimensions ranging from 2 to 6 (see Table 3 and Fig. 18).

Table 3 provides a detailed comparison of the models, showing how each model's performance varies across different PCA dimensions. The data reveal that quantum-classical hybrid models generally outperform the Basic Neural Network in terms of accuracy, particularly at higher PCA dimensions. For example, the model with Angle Embedding and Strongly Entangling Layers achieves the highest precision at lower PCA dimensions but experiences a notable decline at higher dimensions. Conversely, models with IQP Embedding and Basic Entangler Layers exhibit more stable precision across different PCA com-

Table 3 Comparative Numerical Analysis Of The Test Set

Type	PCA	Accuracy	Precision	Recall	F1 Score
Angle Embedding with Strongly Entangling Layers					
	2PCA	81.53%	100%	7.40%	13.79%
	3PCA	93.07%	91.78%	73.62%	81.70%
	4PCA	96.03%	93.58%	84.88%	89.02%
	5PCA	91.17%	82.89%	78.75%	80.76%
	6PCA	88.26%	79.22%	78.20%	78.70%
Basic Neural Network model					
	2PCA	80.68%	83.33%	18.51%	30.30%
	3PCA	91.44%	90.16%	60.43%	72.36%
	4PCA	89.83%	88.60%	73.68%	80.45%
	5PCA	87.98%	84.93%	64.58%	73.37%
	6PCA	85.12%	79.22%	78.70%	78.70%
Angle Embedding with Random Layers					
	2PCA	83.28%	81.48%	27.16%	40.74%
	3PCA	95.78%	98.03%	54.94%	70.42%
	4PCA	95.81%	95.52%	81.01%	87.67%
	5PCA	94.50%	85.13%	78.75%	81.81%
	6PCA	93.25%	85.33%	79.01%	82.05%
IQP Embedding with Random Layers					
	2PCA	79.52%	75.00%	33.33%	46.15%
	3PCA	95.38%	95.58%	71.42%	81.76%
	4PCA	95.10%	91.89%	77.27%	83.95%
	5PCA	95.00%	88.88%	80.00%	84.21%
	6PCA	93.48%	89.06%	70.37%	78.62%
IQP Embedding with Basic Entangler Layers					
	2PCA	84.53%	95.23%	24.69%	39.21%
	3PCA	92.77%	92.85%	71.42%	80.74%
	4PCA	94.56%	90.27%	82.27%	86.09%
	5PCA	94.30%	87.01%	83.75%	85.35%
	6PCA	94.06%	89.85%	76.54%	82.66%
Angle Embedding with Basic Entangler Layers					
	2PCA	80.28%	83.33%	18.51%	30.30%
	3PCA	94.54%	96.55%	61.53%	75.16%
	4PCA	94.51%	91.54%	82.27%	86.66%
	5PCA	90.68%	85.29%	72.50%	78.37%
	6PCA	90.35%	82.66%	76.54%	79.48%
IQP Embedding with Strongly Entangling Layers					
	2PCA	84.91%	95.23%	24.69%	39.21%
	3PCA	93.56%	92.98%	58.24%	71.62%
	4PCA	94.52%	93.15%	77.27%	84.47%
	5PCA	91.11%	86.76%	73.75%	79.27%
	6PCA	91.66%	83.56%	75.30%	79.22%

ponents. Recall and F1 Score metrics indicate that quantum models maintain a higher balance between precision and recall compared to the basic neural network.

Figure 18 visually represents these metrics, showing the trends in Accuracy, Precision, Recall, and F1 Score for each model as the PCA components increase. The plots highlight that models incorporating quantum layers, especially those with entangling mechanisms, are more effective in handling dimensionality-reduced data, resulting in higher overall performance metrics. Specifically, the Angle Embedding with Strongly Entangling Layers model achieves the highest recall and F1 scores at 4 PCA and 5 PCA, demonstrating its superior capability to manage complex, reduced-dimensional data. But 4 PCA shows an enhanced performance than any other PCA so hence we would focus on analysing the

results obtained from 4 PCA HQNNs, specifically the HQNN with Angle embedding and Strongly Entangling layers which gave us the highest accuracy.

The combined insights from Table 3 and Fig. 18 underscore the potential of quantum-classical hybrid models to outperform traditional neural networks in scenarios involving high-dimensional data reduction. These findings suggest that leveraging quantum entangling mechanisms can significantly enhance model performance in classification tasks, particularly when dealing with reduced-dimensional datasets. In our comparative analysis of model performance, we focused on evaluating the effectiveness of two different architectures using the Area Under the Curve (AUC) metric for both training (auc) and validation (val_auc) datasets.

The results in Table 3 and Fig. 18 illustrate the strong performance of the proposed HQNN with 4 PCA components, achieving an accuracy of 96.03% alongside balanced precision, recall, and F1 scores. The decline in performance with 5-6 PCA components can be attributed to the diminishing relevance of the additional features. Higher components in PCA [33] often represent low-variance features, which are less informative or may even capture noise and redundancy. Adding these components makes the model more complicated, which weakens its ability to focus on the key features. This results in a partial diversion of the model's learning capacity to less meaningful details, thereby reducing overall performance. These observations highlight the critical role of selecting an appropriate feature space in quantum machine learning.

The observed discrepancy in the performance of the HQNN model with Angle Embedding and Strongly Entangling Layers (AE+SEL) for 2 PCA, where it achieves 100% precision but a low recall of 7.40%, highlights a key limitation of low-dimensional feature representations. The results reveal that the 2PCA model avoids false positives for the 'No' class, resulting in perfect precision; however, this comes at the expense of misclassifying most 'Yes' class samples as 'No,' leading to drastically low recall. Such performance anomalies are common in models trained with limited feature dimensions, as the low-dimensional representation restricts the model's ability to capture discriminative patterns. In contrast, the proposed 4 PCA model significantly outperforms the 2 PCA model, achieving balanced and superior performance metrics, including an accuracy of 96.03%, precision of 93.58%, recall of 84.88%, and an F1 score of 89.02%. By leveraging an enriched feature set, the 4 PCA model effectively learns meaningful patterns, resulting in a robust decision boundary that accommodates both classes fairly. Though, 4 PCA translates to just 4 features for the model as compared to the original 19 features in the dataset. Still, our proposed HQNN model is able to achieve an improved performance in metrics, whereas, classical models or the state-of-the-art models do not perform well with features as less as the ones experimented in this study because the classical models require more dimensions to learn the patterns in the data but HQNN is doing the same with least number of dimensions. In the PCA technique which we used to reduce the dimensions, more number of features tend to have more noise and less information, whereas most of the information is captured through the first few features extracted by PCA; keeping in mind that the quantum model is a hybrid model with classical layers too, 2 PCA features become too less for the model to learn the pattern. But, among the experimented features from 2-6 PCA, it is already difficult for the state-of-the-art models to learn the pattern of the data with such less features. Experimentally, our proposed HQNN model from among 2-6 PCA learns the best in the case of 4 PCA outperforming its classical counterpart.

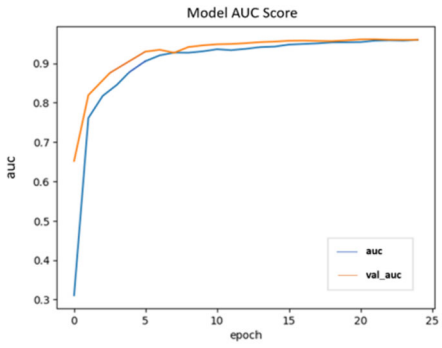


Figure 19 HQNN model with Angle Embedding and a layer circuit of strongly entangling layers (4 PCA)

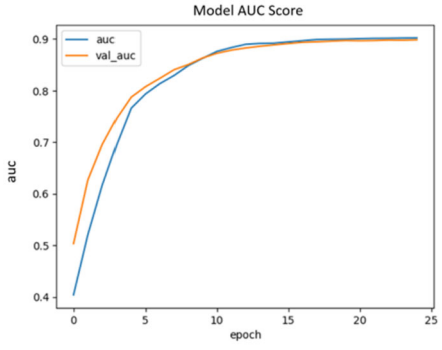


Figure 20 Conventional Neural Network Model (4 PCA)

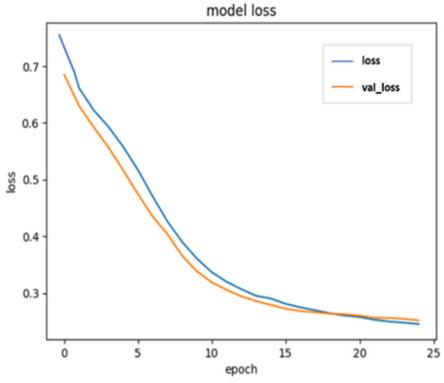


Figure 21 Loss Curve-HQNN model with Angle Embedding and a layer circuit of strongly entangling layers (4 PCA)

This result underscores the quantum advantages in utilizing small, carefully selected feature sets, particularly in data-scarce scenarios. As quantum computing hardware and algorithms continue to evolve, such insights will inform the design of scalable quantum models capable of maintaining robust performance across varying dimensionalities. Moreover, the findings reinforce the potential of HQNNs to address real-world problems effectively, especially in domains where precise feature engineering is possible and quantum advantages can be maximized.

The graph in Fig. 19 represents the performance of a hybrid quantum-classical model with angle embedding and a layer circuit of strongly entangling layers in case of 4 PCA as it gave us the best performance. The graph in Fig. 20 represents a conventional NN model in case of 4 PCA. The graph in Fig. 19 demonstrates that the hybrid quantum-classical

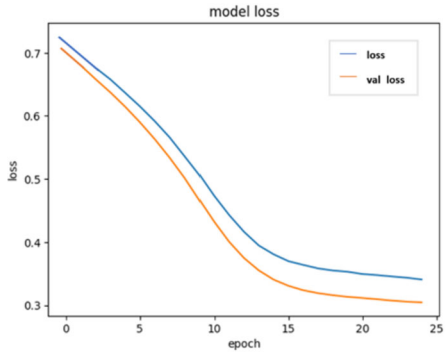


Figure 22 Loss Curve-Conventional Neural Network Model (4 PCA)

model outperforms the conventional NN model across all epochs. The AUC values for both the training and validation datasets are consistently higher in the hybrid model.

This indicates that the hybrid model is more effective at distinguishing between classes, leading to better overall performance. The validation AUC closely tracks the training AUC, indicating that the model generalizes well to unseen data and is likely avoiding overfitting.

In contrast, the graph in Fig. 20 shows lower AUC values for the conventional NN model. Although the training and validation AUCs are closely aligned, indicating good generalization, the overall performance does not match that of the hybrid model. The higher AUC values in the graph in Fig. 19 reflect the hybrid model's superior capability in learning complex patterns and making accurate classifications. This performance is also supported by other metrics, such as precision, recall, and F1-score, which are higher for the hybrid model compared to the conventional neural network model.

Overall, the graph in Fig. 19 clearly indicates that the hybrid quantum-classical model offers a significant performance advantage over the conventional neural network model, making it a more effective choice for tasks requiring high accuracy and robust generalization.

The HQNN (AE and SEL)'s loss curves (in case of 4 PCA which was best), as shown above in Fig. 21 show a more consistent and steady decrease in both training and validation loss compared to the classical NN (4 PCA), as shown in Fig. 22. From the initial epochs, the HQNN maintains a close alignment between the training and validation loss curves, indicating robust generalization and minimal overfitting. This close tracking suggests that the HQNN is effectively learning from the training data without memorizing it, thereby capturing the underlying data patterns more efficiently.

The smooth and gradual decline in loss for the HQNN reflects a stable learning process where the model parameters are being optimized effectively throughout the training period. In contrast, the classical neural network exhibits a more rapid initial decrease in validation loss compared to training loss, which then stabilizes at a slightly lower level. This pattern indicates a strong initial learning phase but also suggests potential overfitting as the training progresses. The widening gap between training and validation loss in the middle epochs is a sign that the model might be starting to memorize the training data rather than learning to generalize from it. While the classical neural network does eventually stabilize, the validation loss being lower than the training loss points towards a less stable and potentially less reliable learning process compared to the HQNN.

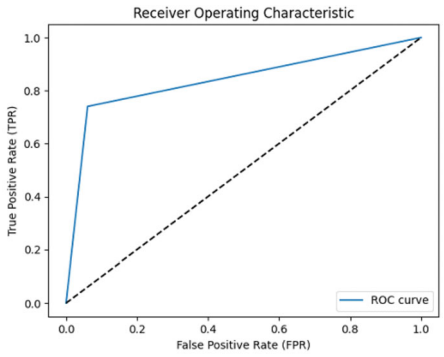


Figure 23 ROC Curve of HQNN model with Angle Embedding and a layer circuit of Strongly Entangling Layers (4 PCA)

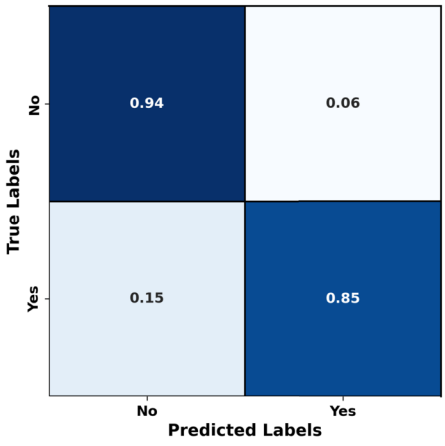


Figure 24 Normalized Confusion Matrix - HQNN model with Angle Embedding and a layer circuit of Strongly Entangling Layers (4 PCA)

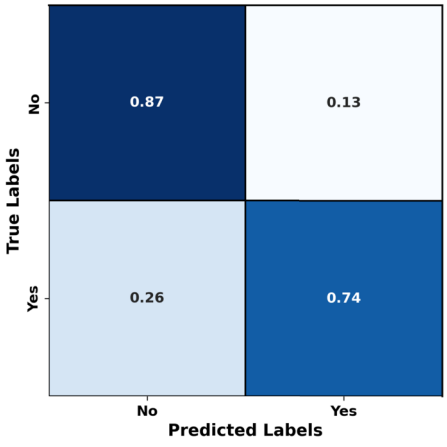


Figure 25 Normalized Confusion Matrix - Conventional Neural Network Model (4 PCA)

Consequently, the HQNN's loss curves, combined with its superior performance metrics (Table 3), underscore the potential of hybrid quantum-classical approaches in enhancing neural network performance, particularly for tasks that benefit from the complex pattern recognition capabilities of quantum computing. The superior performance of the HQNN is further highlighted by its better precision, recall, F1 score, accuracy, and AUC.

These metrics demonstrate that the HQNN not only learns more effectively but also generalizes better, making it a more reliable model for predictive tasks. The use of an-

gle embedding and strongly entangling layers in the HQNN allows it to explore a richer feature space, leading to improved understanding and representation of the data.

The suggested model's performance is measured using the ROC (Receiver Operating Characteristics) curve. This ROC includes the total positive rate (TPR) (denotes "specificity" of the model) and false positive rate (FPR) (denotes "sensitivity" of the model) values of the trained model.

Figure 23 projects the ROC curve for the present HQNN model with AE and a layer of circuit of SEL. (4 PCA). The ROC curve presented for the proposed HQNN model (AE and SEL) shows a strong performance with a significant area under the curve (AUC) close to 1. This indicates that the model has a high TPR and a low FPR across various threshold levels. The steep rise to a TPR of 1.0 at a low FPR suggests that the model is highly effective at distinguishing between the positive and negative classes. The curve's proximity to the top left corner further reinforces the model's excellent discriminative ability, making it a robust choice for classification tasks in the given context. The confusion matrix provides a breakdown of true positives, true negatives, false positives, and false negatives. It is essential for evaluating the accuracy of tsunami predictions using the earthquake dataset.

You can see the representation of the confusion matrix in Fig. 24 and 25 for the HQNN model with AE and SEL and the CNN models, respectively (4 PCA). It can be observed from the above Figs. 24 and 25, that the HQNN model is more accurate in predicting tsunamis as compared to the conventional NN model.

5 Results and conclusion

This research aims to predict tsunamis, a crucial task in disaster management, using the earthquake data over different regions. Initial detection of a potential tsunami relies on identifying the earthquake that might generate it, recording characteristics such as magnitude, depth and location, allowing for immediate analysis and warning of the upcoming disaster. In this study, we assess the effectiveness of the proposed HQNN model by employing various kernels, including AE and IQP embedding along with different entangling layers, namely, SEL, BEL and RL, and further to compare its performance with the Classical NN model in the domain of prediction of tsunami, a natural disaster using earthquake dataset.

The suggested research and supplementary tests were carried out utilizing the Pennylane library. The quantum kernel circuit (QKC) was designed with a qubit-based default qubit simulator architecture in the Microsoft Visual Studio environment. To assess the performance of the HQNN model proposed, we have developed various versions of the model, each with a different number of qubits, spanning from 2 to 6 qubits, to examine its behavior in diverse scenarios. We implemented PCA dimensionality reduction technique to choose the top few significant features. While comparing the results of the proposed HQNN with classical QNN model as shown in Table 3, it was found that at 4 PCA, the model of QNN with AE with SEL, surpasses all other models, including that of Classical NN Model with the greatest accuracy of 96.03%. Furthermore, at 5 and 6 PCA, it was observed that the model with AE with RL and IQP Embedding with BEL surpasses all other models with accuracies of prediction as 95.00% and 94.06%, respectively. Also, at 2 and 3 PCA, it was found that the model with IQP Embedding with SEL and AE with RL surpasses all other models with accuracies of prediction as 84.91% and 95.78%, respectively. However, it was also noted that at 2 PCA, the accuracy of Classical NN Model is slightly higher

than QNN with IQP Embedding with RL and AE with BEL. Hence, the outcomes of this research indicate that the HQNN model outperforms, or in some cases, such as in case of 2 PCA, with IQP Embedding with RL, is almost equal to the classical QNN model in the area of Tsunami prediction. Compared to other variants of the proposed model, the model with AE with BEL, was found to be the best-performing model. Hence, this paper demonstrates the transformative potential of quantum computing in revolutionizing the field of disaster management, enabling people to prepare for upcoming disasters through early warning systems, hence mitigating damage and loss of lives. Future research should focus on improving the integration of earthquake data with advanced computational models to further enhance prediction accuracy and reliability. Expanding the dataset to include more diverse seismic events and incorporating additional environmental factors could also help improve model performance. Additionally, exploring real-time implementation and optimization of quantum computing resources for faster and more accurate predictions remains a vital area for future exploration.

Acknowledgements
The first author and the second author would like to thank Vellore Institute of Technology (VIT), Chennai for the computational facilities and for the motivation.

Author contributions
S.S.D. carried out the implementation of the proposed model and also executed all other models for extensive result analysis. S.S.D. also wrote about the proposed model and detailed analysis. The rest of the manuscript was well organized and written by S.S. D.N. and A.S. did the conceptualization, verification, and reviewing.

Funding
Open access funding provided by Vellore Institute of Technology. Not Applicable.

Data Availability
The earthquake dataset used in this study is publicly available on Kaggle and can be accessed at https://www.kaggle.com/datasets/warcoder/earthquake-dataset.

Declarations

Competing interests
The authors declare no competing interests.

Author details
[1]School of Computer Science and Engineering, Vellore Institute of Technology, Chennai, 600127, Tamil Nadu, India. [2]Department of Ocean Engineering, Indian Institute of Technology Madras, Chennai, 600036, Tamil Nadu, India. [3]Institute of Hydraulic Engineering and Water Resources Management, RWTH Aachen University, Aachen, 52074, Germany.

Received: 9 November 2024 Accepted: 29 December 2024 Published online: 10 January 2025

References

1. Strupler M, Hilbe M, Kremer K, Danciu L, Anselmetti FS, Strasser M, Wiemer S. Subaqueous landslide-triggered tsunami hazard for lake Zurich, Switzerland. Swiss J Geosci. 2018;111:353–71.
2. Li Y, Goda K. Risk-based tsunami early warning using random forest. Comput Geosci. 2023;179:105423.
3. Dharmawan W, Diana M, Tuntari B, Astawa IM, Rahardjo S, Nambo H. Tsunami tide prediction in shallow water using recurrent neural networks: model implementation in the Indonesia tsunami early warning system. J Reliab Intell Environ. 2024;10(2):177–95.
4. Cesario E, Giampà S, Baglione E, Cordrie L, Selva J, Talia D. Forecasting tsunami waves using regression trees. New York: IEEE Press; 2023. p. 1–7.
5. Makinoshima F, Oishi Y, Yamazaki T, Furumura T, Imamura F. Early forecasting of tsunami inundation from tsunami and geodetic observation data with convolutional neural networks. Nat Commun. 2021;12(1):2253.
6. Mulia IE, Gusman AR, Satake K. Applying a deep learning algorithm to tsunami inundation database of megathrust earthquakes. J Geophys Res, Solid Earth. 2020;125(9):2020–019690.
7. Listiani A, Lestari F. Tsunami potential prediction with artificial neural network. 2023.
8. Novianty A, Machbub C, Widiyantoro S, Meilano I, Daryono Q. Tsunami potential prediction using seismic features and artificial neural network for tsunami early warning system. Int J Adv Sci Eng Inf Technol. 2022;12(1):16–22. https://doi.org/10.18517/ijaseit.12.1.14237.
9. Siek M, Rafles A. Data-driven modelling for tsunami forecasting using computational intelligence. In: 2022 IEEE international conference on cybernetics and computational intelligence (CyberneticsCom). New York: IEEE Press; 2022. p. 185–90.

10. Gálvez JFR, Sáncez JM, Díaz MJC, Asunción M, Sánchez-Linares C. Use of neural networks for tsunami maximum height and arrival time predictions. Technical report, Copernicus Meetings. 2023.
11. Andraud P, Gailler A, Dias F, Vayatis N. Deep learning approach for real-time tsunami impact forecasting in near field context-application to the French Mediterranean coastline. In: EGU general assembly conference abstracts. 2023. p. 7763.
12. Korolev Y. A new approach to short-term tsunami forecasting. Edited by Gloria I Lopez, 141. 2012.
13. Xu H, Wu H. Accurate tsunami wave prediction using long short-term memory based neural networks. Ocean Model. 2023;186:102259.
14. Sebastianelli A, Zaidenberg DA, Spiller D, Le Saux B, Ullo SL. On circuit-based hybrid quantum neural networks for remote sensing imagery classification. IEEE J Sel Top Appl Earth Obs Remote Sens. 2021;15:565–80.
15. Paquet E, Soleymani F. Quantumleap: hybrid quantum neural network for financial predictions. Expert Syst Appl. 2022;195:116583.
16. Sagingalieva A, Kordzanganeh M, Kenbayev N, Kosichkina D, Tomashuk T, Melnikov A. Hybrid quantum neural network for drug response prediction. Cancers. 2023;15(10):2705.
17. Jeong S-G, Do Q-V, Hwang H-J, Hasegawa M, Sekiya H, Hwang W-J. Hybrid quantum convolutional neural networks for uwb signal classification. Ieee Access. 2023.
18. Heidari H, Hellstern G. Early heart disease prediction using hybrid quantum classification. 2022. arXiv preprint. arXiv:2208.08882.
19. Chauhan C. Kaggle. https://www.kaggle.com/datasets/warcoder/earthquake-dataset. Accessed; 2022.
20. Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP. Smote: synthetic minority over-sampling technique. J Artif Intell Res. 2002;16:321–57.
21. Chen R-C, Dewi C, Huang S-W, Caraka RE. Selecting critical features for data classification based on machine learning methods. J Big Data. 2020;7(1):52.
22. Kappal S, et al. Data normalization using median absolute deviation mmad based z-score for robust predictions vs. min–max normalization. London J Res Sci Nat Form. 2019;19(4):39–44.
23. Hasan BMS, Abdulazeez AM. A review of principal component analysis algorithm for dimensionality reduction. J Soft Comput Data Min. 2021;2(1):20–30.
24. Sridevi S, et al. Quantum kernel-aided support vector machine classifier for improved speech classification. In: 2023 14th international conference on computing communication and networking technologies (ICCCNT). New York: IEEE Press; 2023. p. 1–6.
25. Sridevi S, Indira B, Dutta SS, Sandeep S, Sreenivasan A. Quantum enhanced support vector machine with instantaneous quantum polynomial encoding for improved cyclone classification. In: 2023 6th international conference on recent trends in advance computing (ICRTAC). New York: IEEE Press; 2023. p. 748–52.
26. Dutta SS, Sandeep S, Sridevi S, Sridhar MR, Singh C. Quantum kernel based support vector machine with quantum approximate optimization algorithm embedding for improved lung cancer prediction. In: 2024 15th international conference on computing communication and networking technologies (ICCCNT). New York: IEEE Press; 2024. p. 1–6.
27. Khadka N. General machine learning practices using python. 2019.
28. Agarap A. Deep learning using rectified linear units (relu). 2018. arXiv preprint. arXiv:1803.08375.
29. Kim H, Lim S, Baksi A, Kim D, Yoon S, Jang K, Seo H. Quantum artificial intelligence on cryptanalysis. Cryptology ePrint Archive. 2023.
30. Xiong H, Wu Z, Fan H, Li G, Jiang G. Quantum rotation gate in quantum-inspired evolutionary algorithm: a review, analysis and comparison study. Swarm Evol Comput. 2018;42:43–57.
31. Lee CY. Numerical simulations of quantum neural network and quantum circuit. 2022.
32. Pratiwi H, Windarto AP, Susliansyah S, Aria RR, Susilowati S, Rahayu LK, Fitriani Y, Merdekawati A, Rahadjeng IR. Sigmoid activation function in selecting the best model of artificial neural networks. Journal of physics: conference series. vol. 1471. Bristol: IOP Publishing; 2020. p. 012010.
33. Wold S, Esbensen K, Geladi P. Principal component analysis. Chemom Intell Lab Syst. 1987;2(1–3):37–52.

Publisher's Note