INDIRA GANDHI DELHI TECHNICAL UNIVERSITY FOR WOMEN

Quantum Computing Project
Quantum Random Forest and its Applications
BCS-410

Submitted by-
Diksha Malik (08901012020)
Kuhuk Agarwal (09001012020)
BTech CSE (8th semester)

Submitted to-
Ms. Karuna Kadian
Assistant professor
CSE Dept, IGDTUW

Comparative Analysis of Quantum and Traditional Random Forests for Heart Attack Prediction

PROBLEM STATEMENT

Heart disease is a leading cause of global mortality, underscoring the critical need for effective early detection methods. Existing diagnostic tools often lack the accuracy needed to reliably identify individuals at risk of heart attacks. To address this challenge, we aim to leverage machine learning techniques using the Heart Attack Analysis & Prediction Dataset. Specifically, we will compare the effectiveness of two methodologies: the conventional Random Forest algorithm and the emerging Quantum Random Forest approach. By evaluating the predictive performance of these models, we seek to determine which method offers superior capabilities in identifying individuals at risk of heart attacks.

PROPOSED SOLUTION

Our proposed solution involves conducting a comparative analysis of Random Forest and Quantum Random Forest models using the Heart Attack Analysis & Prediction Dataset.

Dataset:
https://www.kaggle.com/datasets/rashikrahmanpritom/heart-attack-analysis-prediction-dataset

Random Forest has proven to be a robust and widely used machine learning technique, known for its ability to handle complex datasets and provide reliable predictions. However, with the advent of quantum computing, there is growing interest in exploring the potential advantages of Quantum Random Forests, which harness quantum mechanics principles to enhance computational power and potentially improve predictive performance.

By conducting a comprehensive comparative analysis, we seek to determine which approach - Random Forest or Quantum Random Forest - demonstrates superior predictive accuracy in identifying individuals at risk of heart attacks. Our evaluation will consider various metrics, including accuracy, precision, recall, and f1-score, to provide a comprehensive assessment of model performance.

This research is motivated by the urgent need for advanced predictive models in cardiovascular health assessment. A more accurate and efficient predictive tool could revolutionize early detection efforts, enabling timely interventions and ultimately reducing the morbidity and mortality associated with heart disease. By shedding light on the comparative effectiveness of conventional and quantum-enhanced machine learning techniques in this context, our study aims to contribute valuable insights to the field of cardiovascular medicine and pave the way for future advancements in predictive analytics for heart health.

METHODOLOGY

Implementation of Quantum Random forest

A kernel-based quantum random forest (QRF) is a novel approach to classification tasks that leverages the potential of quantum computing to improve upon classical random forests. Here's a breakdown of the key concepts:

Classical Random Forests:

- Traditional random forests are ensemble classifiers consisting of multiple decision trees.
- Each tree makes a classification prediction based on a series of features in the data.
- By combining the predictions of multiple trees, random forests aim to improve accuracy and reduce overfitting.

Quantum Random Forest (QRF):

- QRF builds upon the idea of random forests but incorporates elements of quantum machine learning (QML).
- It utilizes a concept called a quantum support vector machine (QSVM), which is a quantum analogue of the classical support vector machine (SVM) used for classification.
- The key difference lies in the way QRF handles the kernel function, a mathematical tool used by SVMs to capture complex relationships in the data.

Kernel Function in QRF:

- In classical random forests, the kernel function is computed using classical algorithms.
- QRF employs quantum kernel estimation (QKE) to compute the kernel function. This leverages the unique properties of quantum mechanics to potentially find more intricate relationships within the data.
- However, QKE can be computationally expensive. To address this, QRF incorporates a technique called the Nyström approximation, which reduces the complexity while maintaining reasonable accuracy.

Benefits of QRF:

- The proponents of QRF argue that by using QKE, it can capture more complex data patterns compared to classical random forests.

- This potentially leads to improved classification accuracy, particularly for datasets with non-linear relationships between features.
- Additionally, QRF offers theoretical guarantees on limiting errors due to the Nyström approximation.

Current Stage of Development:

- It's important to note that QRF is a relatively new research area. The current implementations are focused on proof-of-concept and exploring its potential.
- While the research suggests promising results, large-scale practical applications of QRF are yet to be established.
- Overall, kernel-based quantum random forests represent an exciting exploration of how quantum computing can enhance classical machine-learning techniques for classification tasks. However, the field is still evolving, and further research is needed to determine its full capabilities and real-world practicality.

Reference: A kernel-based quantum random forest for improved classification

CODE: https://github.com/kuhuk521/Quantum-Random-Forest/tree/main

LITERATURE REVIEW

The paper introduces the Quantum Random Forest (QRF) model, which extends the linear quantum support vector machine (QSVM) with a unique kernel function computed through quantum kernel estimation (QKE). The QRF is constructed from an ensemble of quantum decision trees (QDTs) and employs a low-rank Nyström approximation to the kernel matrix to limit overfitting. The research provides generalisation error bounds, theoretical guarantees, and numerical results demonstrating the QRF's superior classification performance over QSVMs while requiring fewer kernel estimations. The model offers expressional and computational advantages in quantum machine learning and shows potential for practical implementation in real-world applications.

Code implemented-

1. **Initialisation**:
    - Initialise the parameters and hyperparameters of the Quantum Random Forest (QRF) model, such as the number of trees (T), the depth of each tree (D), the Nyström approximation parameters, and any other relevant settings.
2. **Data Preprocessing**:
    - Preprocess the input data for training and testing the QRF model. This may include data normalisation, feature scaling, handling missing values, and splitting the data into training and testing sets.
3. **Training Phase**:
    - For each tree in the ensemble:
        - Train a quantum decision tree (QDT) using a subset of the training data.
        - Construct the QDT by optimising the split functions based on the quantum support vector machines (QSVMs) with the unique kernel function computed through quantum kernel estimation.
        - Apply the low-rank Nyström approximation to the kernel matrix to limit overfitting.
4. **Prediction Phase**:
    - Given a new instance for classification:
        - Pass the instance through each QDT in the ensemble.
        - Aggregate the predictions from all trees to make a final classification decision.
5. **Evaluation**:
    - Evaluate the performance of the QRF model using metrics such as accuracy, precision, recall, F1 score, and confusion matrix on the test data.
6. **Parameter Tuning**:
    - Optionally, perform hyperparameter tuning using techniques like grid search or random search to optimise the model's performance.

7. **Results Analysis**:
   - ○ Analyse the results obtained from the QRF model, compare them with baseline models like QSVMs, and conclude the effectiveness of the QRF for classification tasks.

What paper concluded-

1. **Model Development**: The paper introduces the QRF model, which extends the linear quantum support vector machine (QSVM) with a kernel function computed through quantum kernel estimation (QKE). The QRF is constructed from a decision-directed acyclic graph of QSVM nodes, forming an ensemble of quantum decision trees (QDTs).
2. **Overfitting Limitation:** To address overfitting, the model employs a low-rank Nyström approximation to the kernel matrix, known as Nyström-QKE (NQKE). This approximation strategy helps in reducing overfitting and improving the generalisation performance of the model.
3. **Theoretical Guarantees**: The paper provides generalisation error bounds on the model and theoretical guarantees to limit errors due to finite sampling on the Nyström-QKE strategy. These guarantees ensure the robustness and reliability of the QRF model in real-world applications.
4. **Performance Comparison:** Numerical illustrations demonstrate the effect of varying model hyperparameters and show that the QRF outperforms QSVMs in terms of classification performance while requiring fewer kernel estimations. This highlights the superiority of the QRF model in certain scenarios.
5. **Reduced Sampling Complexity:** The inclusion of the Nyström-QKE approximation not only limits overfitting but also allows for a reduced circuit sampling complexity, making the model more efficient and practical for implementation.
6. **Future Directions:** The paper suggests potential future directions for research, such as exploring boosting techniques for ensembles of QSVMs and further investigating the benefits of ensemble structures in quantum machine learning models.

CODE OVERVIEW

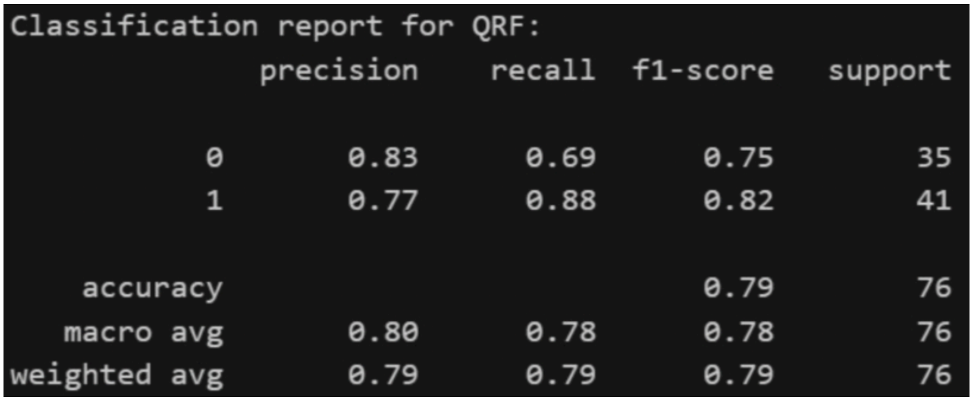
1. **Quantum Random Forest Function:**
    - **Ensemble of Quantum Decision Trees (QDTs):** The QRF consists of a set of quantum decision trees, where each tree is constructed using quantum support vector machines (QSVMs) with a unique kernel function computed through quantum kernel estimation.
    - **Split Function**: The split function in each QDT partitions the data based on a separating hyperplane in a higher-dimensional quantum feature space. This function aims to maximize information gain at each node to effectively separate instances of different classes.
    - **Nyström Approximation:** To limit overfitting, the QRF employs a low-rank Nyström approximation to the kernel matrix, reducing the complexity of the model while maintaining classification accuracy.
    - **Prediction:** During prediction, a new instance is passed through each QDT in the ensemble, and the aggregated predictions from all trees determine the final classification decision.
2. **Data Preprocessing:**
    - **Normalization:** Scale the input features to a standard range to prevent certain features from dominating the model due to their scale.
    - **Handling Missing Values:** Address any missing data points in the dataset through imputation or removal strategies.
    - **Feature Engineering:** Create new features or transform existing ones to enhance the model's performance.
    - **Train-Test Split:** Divide the dataset into training and testing sets to evaluate the model's performance on unseen data.
    - **Quantum Data Encoding:** Convert classical data into quantum states suitable for processing on a quantum computer, if applicable.
    - **Dimensionality Reduction:** Reduce the number of features using techniques like Principal Component Analysis (PCA) to improve computational efficiency and reduce noise.
3. **Split function:**
    - The split function plays a crucial role in the decision-making process of the tree by determining how the instances are divided into different branches.
    - Specifically, the split function in the QRF model is responsible for creating a separating hyperplane in a higher-dimensional quantum feature space. This hyperplane helps distinguish instances of different classes by maximizing the information gain at each node of the decision tree.
    - he goal is to separate the data points effectively so that instances of different classes end up in different leaf nodes, facilitating accurate classification during prediction.

- By employing a split function based on a support vector machine (SVM) that utilizes a quantum kernel, the QRF model aims to enhance the separation of classes in the feature space while also addressing the risk of overfitting.
- The split function guides the construction of the decision tree in the QRF ensemble, contributing to the overall classification performance of the model.

RESULTS

1. **Quantum Random Forest:**

For 6 qubits and 100 trees:

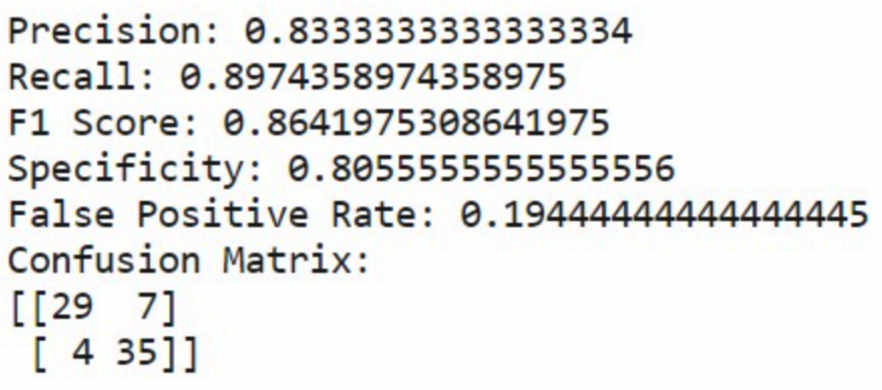


```
Classification report for QRF:
              precision    recall  f1-score   support

           0       0.83      0.69      0.75        35
           1       0.77      0.88      0.82        41

    accuracy                           0.79        76
   macro avg       0.80      0.78      0.78        76
weighted avg       0.79      0.79      0.79        76
```

2. **Traditional Random Forest algorithm:**



```
Precision: 0.8333333333333334
Recall: 0.8974358974358975
F1 Score: 0.8641975308641975
Specificity: 0.8055555555555556
False Positive Rate: 0.19444444444444445
Confusion Matrix:
[[29  7]
 [ 4 35]]
```

ANALYSIS

The Traditional Random Forest algorithm generally performed better than the Quantum Random Forest algorithm regarding precision, recall, and F1 score.

However, the Quantum Random Forest algorithm, demonstrated comparable performance to the Traditional Random Forest algorithm, indicating its potential utility despite its current performance gap.

Further optimisation and exploration of Quantum Random Forest algorithms, potentially with more qubits or different configurations, may improve predictive performance and bridge the performance gap with traditional methods.

OPTIMISATION TECHNIQUES EMPLOYED

In the context of the Quantum Random Forest (QRF) model described in the paper, an optimization technique known as **Randomized Node Optimization (RNO)** is used to ensure that the correlation between Quantum Decision Trees (QDTs) in the ensemble is minimized.

Here is how the Randomized Node Optimization technique works in the QRF model:

- **Randomness Injection**: RNO introduces randomness by selecting landmark data points, which results in optimized hyperplanes that vary depending on the subset chosen.
- **Hyperparameter Variation**: The technique allows for the variation of hyperparameters, such as Φ and L, both within and across trees in the ensemble. This variation helps in creating diverse and uncorrelated weak classifiers.
- **Kernel Embeddings**: The specific kernel defined by an embedding Φ implies a unique Hilbert space of functions, enhancing the expressiveness of the model.
- **Distinct Kernels**: The QRF approach employs distinct kernels at each depth of the tree, providing a more expressive enhancement compared to a simple ensemble of independent QSVMs.

By utilizing Randomized Node Optimization, the QRF model aims to create a diverse ensemble of Quantum Decision Trees with unique split functions, enhancing the overall classification performance and robustness of the model.

What else can be done?

Paper referenced- https://link.springer.com/article/10.1007/s00500-020-05274-z

Quantum Walk- A quantum walk is a quantum mechanical analogue of classical random walks, where a particle moves through a graph or lattice according to quantum rules. In a quantum walk, the particle is described by a quantum state that evolves over time, influenced by quantum operations. Quantum walks have been studied extensively in quantum computing and quantum information theory due to their potential for designing novel algorithms with applications in various fields.

Quantum walks exhibit unique properties compared to classical random walks, such as high non-linearity, parallelism, and potential exponential speedup in certain tasks. They have been shown to be efficient for tasks like searching, sampling, and matching in graph analysis. Quantum walks can be implemented in different physical systems, such as optical setups, quantum circuits, and quantum computers, to explore complex relationships and patterns in graphs or networks.

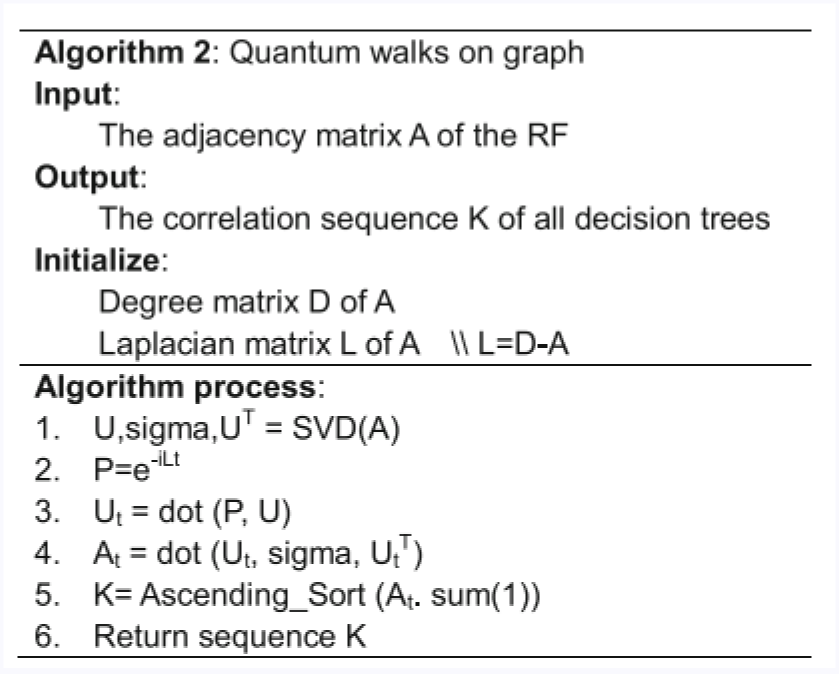
Overall, quantum walks play a significant role in quantum algorithms and quantum information processing, offering advantages over classical random walks in terms of computational efficiency and problem-solving capabilities.

Quick overview of the given paper:

- **Quantum Walks on the Graph:** The document introduces the concept of quantum walks on a graph, where a continuous-time quantum walk algorithm is used to explore complex relationships among decision trees in a random forest (RF). Quantum walks help in selecting decision trees with high margin scores and good diversity from the RF.
- **Algorithm and Methodology:** The algorithm involves initializing the quantum state of each vertex on the graph and applying an evolution operator based on continuous time. The adjacency matrix of the RF is used along with singular value decomposition (SVD) to express the quantum state of each vertex.
- **Ensemble Pruning:** The proposed method focuses on ensemble pruning in deep forest models to create a pruned deep forest (PDF) with a simplified model and improved performance. By optimizing individual forests in each layer of the deep forest, the PDF achieves higher accuracy and reduced ensemble size.
- **Experimental Validation:** The PDF file describes experiments conducted on 15 datasets from the UCI repository to validate the effectiveness of the proposed ensemble pruning method. Results show that the PDF outperforms the deep forest and other classic ensemble learning methods on most datasets.
- **Future Work:** The document suggests future work to verify the proposed pruning method on non-forest-style ensembles and dynamically determine the reserved proportion of base classifiers for enhanced generalizability on different datasets.

The document discusses the use of continuous-time quantum walks on the graph to select decision trees with high margin scores and good diversity from the random forest. The quantum walks algorithm initializes the quantum state of each vertex on the graph and applies an evolution operator to govern the change of the quantum state based on continuous time. The algorithm involves the use of the adjacency matrix of the random forest and singular value decomposition (SVD) to express the quantum state of each vertex.

<u>The algorithm suggested for writing quantum walk on a graph</u>



```
Algorithm 2: Quantum walks on graph
Input:
    The adjacency matrix A of the RF
Output:
    The correlation sequence K of all decision trees
Initialize:
    Degree matrix D of A
    Laplacian matrix L of A   \\ L=D-A
Algorithm process:
1.  $U, sigma, U^T$ = SVD(A)
2.  $P=e^{-iLt}$
3.  $U_t$ = dot (P, U)
4.  $A_t$ = dot ($U_t$, sigma, $U_t^T$)
5.  K= Ascending_Sort ($A_t$. sum(1))
6.  Return sequence K
```

Quantum State Initialization:

- Each decision tree in the random forest (RF) is represented as a vertex on a graph.
- Initially, the quantum state of each decision tree (vertex) is prepared using a technique called singular value decomposition (SVD) of the adjacency matrix of the RF.

2. Quantum State Evolution:

- The quantum state of each vertex evolves over time in a continuous manner.
- This evolution is governed by an evolution operator, which is determined based on the Laplacian matrix of the graph.

3. Adjacency Matrix at Time t:

- At each time step, an updated adjacency matrix is computed to represent the connections between vertices.
- This matrix reflects the evolved relationships between decision trees in the RF.

4. Quantum Walks Score (QW):

- The Quantum Walks Score (QW) of each decision tree (vertex) is calculated based on its connectivity with other decision trees in the RF.
- Higher QW scores indicate higher redundancy or similarity with other decision trees.

5. Vertex Sorting and Selection:

- Decision trees are sorted based on their QW scores, with higher scores indicating higher redundancy.
- Trees with high redundancy are candidates for pruning, as they contribute less diversity to the model.

6. Pruning and Model Optimization:

- Pruning involves removing decision trees with high redundancy, simplifying the RF while retaining unique contributions.
- This process aims to create a pruned deep forest (PDF) with improved simplicity and potentially enhanced accuracy.

7. Experimental Validation:

- The algorithm's effectiveness is evaluated through experiments on datasets from the UCI repository.
- Performance metrics are compared against other common algorithms such as Support Vector Machine (SVM), original deep forest, random forest, and bagging.

Step-by-step integration of quantum random forest and quantum walk

Step 1: Quantum Random Walk Setup

- Discrete Quantum Random Walk (DQRW): Choose a graph structure where nodes represent states or decisions in your decision tree. For each node, define unitary operations (coin and shift operations) that represent the evolution of the walker (data point navigating the decision tree).

Step 2: Building Quantum Decision Trees

- Node Decisions via QRW: At each decision node within the tree, use the outcome of a quantum random walk to decide the splitting criterion (e.g., based on the probability distribution of the quantum walker's position).

- State Preparation and Measurement: Prepare the initial state of the quantum walker based on the features of the data point. Measurement outcomes after evolving the walk can determine the path taken by the data through the decision tree.

Step 3: Quantum Forest Ensemble

- Multiple Trees with QRW Variations: Construct multiple quantum decision trees, varying the initial conditions or the parameters of the quantum random walk (like different quantum coins or initial superpositions).
- Aggregation of Outputs: Use a quantum or classical method to aggregate the outputs of different quantum trees. This could involve a quantum voting mechanism where the most common class label among the measured outcomes of the QRWs determines the final prediction.

Step 4: Quantum Computation and Classical Integration

- Quantum Circuit Execution: Implement the quantum random walk and decision tree logic on a quantum processor or through a quantum simulation. Each tree in the forest might require multiple runs of a quantum circuit.
- Classical Coordination: Manage the ensemble learning aspect classically, where the results from each quantum decision tree are combined to produce the final output.

Challenges and Considerations

- Technological Limitations: Current quantum technology might not yet support complex operations and large-scale integrations required by a full quantum random forest.
- Error Rates and Decoherence: Quantum operations are prone to errors and decoherence, which can affect the stability and reliability of the quantum decision processes.
- Complexity vs. Gain: Evaluate whether the quantum approach provides significant advantages over classical methods in terms of speed or accuracy for your specific application.

APPLICATIONS OF QRF

1. **Classification and Regression Tasks**: Like classical random forests, QRFs can be used for both classification and regression tasks in various domains such as finance, healthcare, and natural language processing.
2. **Quantum Chemistry:** Quantum random forests can be employed in quantum chemistry for tasks such as molecular property prediction, molecular dynamics simulation, and drug discovery. The quantum nature of QRFs can potentially offer advantages in handling quantum states of molecules more efficiently.
3. **Quantum Image Processing:** QRFs can be applied to quantum image processing tasks, including image classification, object detection, and image segmentation. This can find applications in fields like quantum imaging and quantum communication.
4. **Financial Modeling and Forecasting:** QRFs can be used in financial modelling and forecasting to analyse stock market data, predict asset prices, detect anomalies, and optimise investment portfolios. The quantum capabilities of QRFs may offer advantages in capturing complex patterns in financial data.
5. **Bioinformatics:** In bioinformatics, QRFs can be utilised for tasks such as genomic data analysis, protein structure prediction, and biomarker discovery. The ability of QRFs to handle high-dimensional data and extract meaningful features can be beneficial in biological research.
6. **Quantum Sensing and Metrology:** QRFs can play a role in quantum sensing and metrology applications by processing data obtained from quantum sensors. This includes tasks such as signal processing, pattern recognition, and anomaly detection in quantum sensor measurements.
7. **Natural Language Processing (NLP):** Quantum random forests can be applied to NLP tasks such as text classification, sentiment analysis, named entity recognition, and machine translation. The quantum-enhanced capabilities may enable more accurate and efficient processing of natural language data.
8. **Healthcare Analytics:** QRFs can be employed in healthcare analytics for tasks such as disease diagnosis, patient outcome prediction, and medical image analysis. The quantum approach may offer advantages in handling large-scale healthcare datasets and extracting relevant insights.
9. **Cybersecurity**: QRFs can be utilised in cybersecurity applications for tasks such as intrusion detection, malware classification, and network traffic analysis. The quantum capabilities of QRFs may enhance the detection of sophisticated cyber threats and improve overall system security.
10. **Climate Modeling:** QRFs can contribute to climate modelling efforts by analysing environmental data, predicting weather patterns, and assessing the impact of climate change. The quantum approach may enable more accurate and detailed simulations of complex climate systems