BCSE408L Course Project

Quantum Anomaly Detection (Cybersecurity/IoT) using IBM Cloud

22BCE2478 **Madhur Pophale**

Under the Supervision of

Prof. Sankar Ganesh L

Assistant Professor

School of Computer Science and Engineering (SCOPE)

B.Tech.

in

Computer Science and Engineering

School of Computer Science and Engineering

November 2025

TABLE OF CONTENTS

1. Project Description and Goals

1.1. Literature Review

Anomaly detection is a fundamental task in various domains, including network intrusion detection, fraud detection, medical diagnostics, and industrial monitoring. Traditional approaches to anomaly detection include statistical models, distance- and density-based methods, and machine learning techniques. Statistical methods, such as Gaussian Mixture Models and z-score analysis, identify anomalies by assuming a probabilistic distribution for normal data. Distance- and density-based methods, including k-Nearest Neighbors (k-NN) and Local Outlier Factor (LOF), detect anomalies based on deviations in density or distance from normal clusters. Machine learning approaches, such as Support Vector Machines (SVMs), Isolation Forests, and Autoencoders, provide more flexible models for separating anomalous data points from normal instances. While effective in many applications, these classical methods face challenges when handling high-dimensional or complex datasets, where anomalies may be hidden in intricate, non-linear feature relationships.

Quantum computing introduces a fundamentally different approach to machine learning, leveraging principles such as superposition, entanglement, and interference to process information in exponentially large Hilbert spaces. Quantum Machine Learning (QML) techniques, particularly **quantum kernel methods**, have emerged as promising tools for anomaly detection. In these methods, classical data is mapped into high-dimensional quantum states via parameterized quantum circuits, known as quantum feature maps. The inner product of quantum states forms a quantum kernel, which can be used with classical algorithms, such as SVMs, to classify data. This approach allows for potentially improved separation of normal and anomalous samples, particularly in datasets with non-linear correlations that are difficult to capture using classical kernels.

Several studies have demonstrated the potential of quantum kernel methods in classification and anomaly detection tasks. Havlíček et al. (2019) showed that quantum kernels could create decision boundaries that are hard to reproduce classically, improving classification accuracy in synthetic datasets. Cong et al. (2019) explored circuit-based feature maps for machine learning and highlighted their ability to represent high-dimensional features efficiently. In the context of anomaly detection, recent works by Schuld et al. (2021) and Li et al. (2022)

applied quantum kernels to detect network anomalies and cyberattacks, achieving competitive performance on small-scale datasets and demonstrating particular strength in cases where classical kernels struggled to separate complex feature patterns.

Despite these advances, several limitations remain. Most studies are restricted to small datasets and low qubit counts due to current hardware constraints. Quantum devices are also affected by noise and decoherence, which can impact kernel estimation and classification accuracy. Additionally, there is limited benchmarking against state-of-the-art classical anomaly detection methods, and practical applications in real-world scenarios are sparse. These challenges highlight the need for further exploration and motivate the current project, which aims to implement quantum anomaly detection, compare its performance with classical baselines, and assess its feasibility on simulators or available quantum hardware.

1.2. Research Gaps

Although quantum anomaly detection has shown promising results, several gaps remain that limit its practical applicability. Some of these include:

- **Scalability limitations:** Most existing quantum anomaly detection studies work with small datasets and very few qubits. This is mainly because current quantum hardware cannot handle large numbers of qubits reliably. As a result, it is unclear how well these methods would perform on bigger, real-world datasets like network traffic logs or industrial sensor data.
- **Impact of hardware noise:** Real quantum devices are not perfect. They suffer from errors caused by decoherence and imperfect gate operations. These errors can reduce the accuracy of quantum kernel calculations and make results inconsistent. Very few studies investigate how this noise affects anomaly detection performance.
- **Limited benchmarking with classical methods:** Many studies only compare quantum kernels with simple classical methods like RBF kernel SVMs. They rarely compare with stronger classical anomaly detection models, such as Isolation Forests, Autoencoders, or ensemble techniques. This makes it difficult to understand the real advantage of quantum methods.

- **Lack of real-world applications:** Most research uses synthetic or small, clean datasets for testing. Very few studies apply quantum anomaly detection to real-life problems such as network intrusion detection, fraud detection, or industrial monitoring. This limits understanding of how the method performs in practical scenarios.
- **Evaluation under realistic conitions:** There is a need to study quantum anomaly detection under realistic conditions, including noisy hardware, limited data, and high-dimensional feature spaces. Understanding these factors can help in making quantum approaches more practical and reliable.

1.3. Objectives

- **Implement quantum anomaly detection:** Build a pipeline using Qiskit to detect anomalies in a given dataset by applying quantum kernel methods. This includes encoding classical data into quantum feature maps and training a kernel-based classifier.

- **Compare quantum and classical methods:** Evaluate the performance of the quantum kernel approach against classical anomaly detection techniques, such as RBF kernel SVM, to understand the advantages and limitations of the quantum method.

- **Analyze classification performance:** Measure and report metrics like confusion matrix, precision, recall, and F1-score to quantify how well the system identifies normal and anomalous samples.

- **Explore hardware and simulation effects**: Study the difference between running the model on simulators versus real quantum hardware, and observe the impact of noise and limited qubits on performance.

- **Identify practical limitations and improvements:** Highlight challenges such as scalability, noise, and dataset size, and suggest ways to improve the method for future applications.

2. Requirement Analysis

2.1. Hardware and Software Specifications

A. Hardware Specifications:

- **Processor**: A modern multi-core CPU (Intel i5/i7 or equivalent) is recommended to handle simulations efficiently.

- **RAM**: Minimum 8 GB of memory is required; 16 GB is preferred for larger datasets or faster computation.

- **Storage**: At least 10 GB of free disk space for storing datasets, code, and intermediate outputs.

- **GPU**: Not mandatory for this project, but a GPU can speed up data preprocessing and visualization if available.

- **Quantum Hardware (Optional)**: For experiments on real quantum devices, an IBM Quantum backend (such as ibmq_qasm_simulator or ibmq_quito) can be used. These backends allow the execution of quantum circuits on actual hardware, but are limited by qubit count and noise.

B. Software Specifications:

- **Operating System**: Windows 10, Linux, or macOS can be used.

- **Programming Language**: Python 3.9 or higher.

- **Libraries and Frameworks**:

1. Qiskit: For quantum circuit creation, feature map embedding, and quantum kernel evaluation.

2. scikit-learn: For classical SVM classifier and evaluation metrics.

3. NumPy / Pandas: For data handling and preprocessing.

4. Matplotlib / Seaborn: For visualization of results such as confusion matrices and kernel heatmaps.

- **Development Environment**: Jupyter Notebook or Google Colab is recommended for interactive coding and visualization.

The project is designed to run on simulators for most experiments, which allows testing larger datasets and circuits without hardware constraints. Optional real-device runs can demonstrate the effects of quantum noise and limited qubits, providing insights into practical hardware challenges.

2.2. Work Breakdown Structure

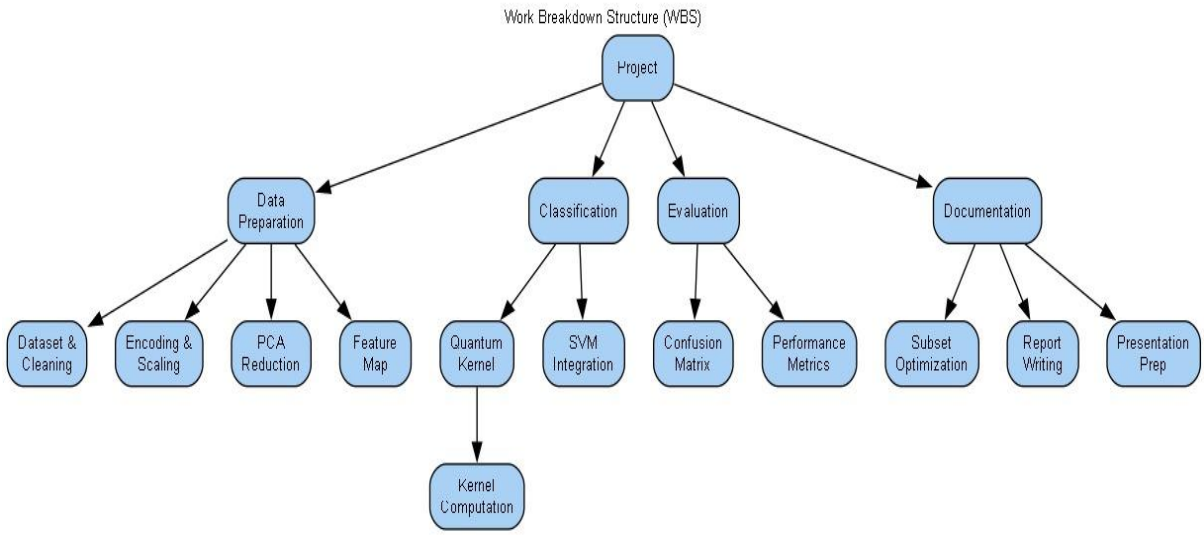


Fig.1 Work Breakdown Structure of the project

2.3. Project Timeline

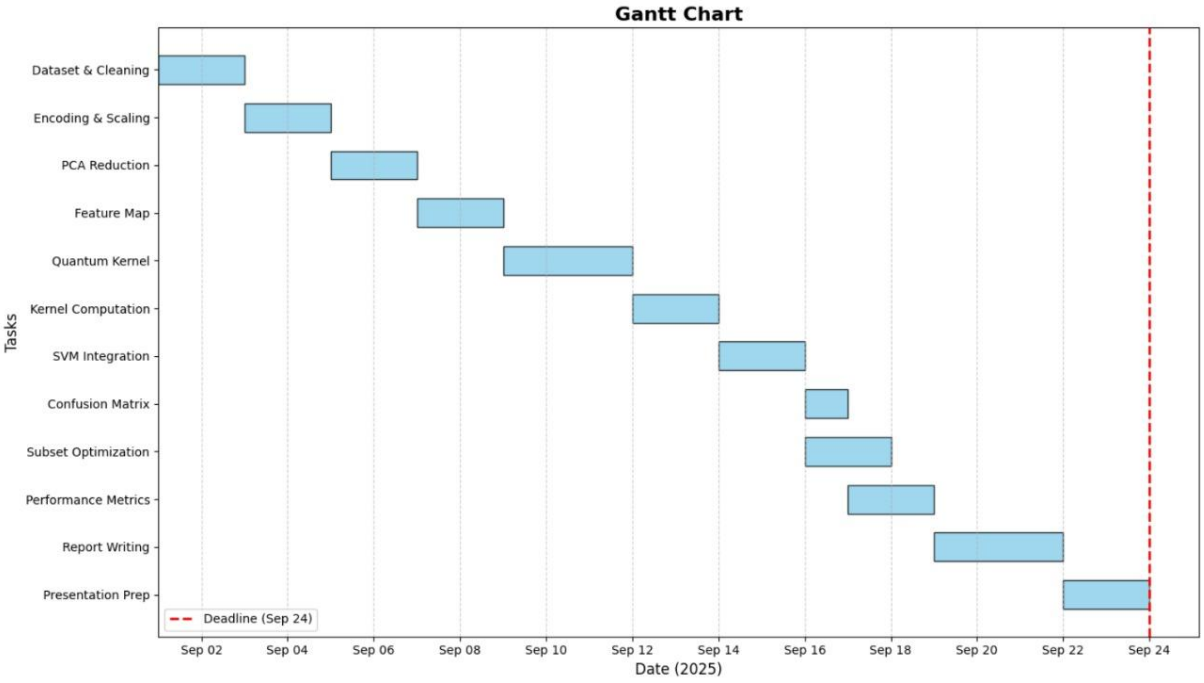


Fig.2. Project timeline using Gantt Chart

2.3. Workflow Diagram

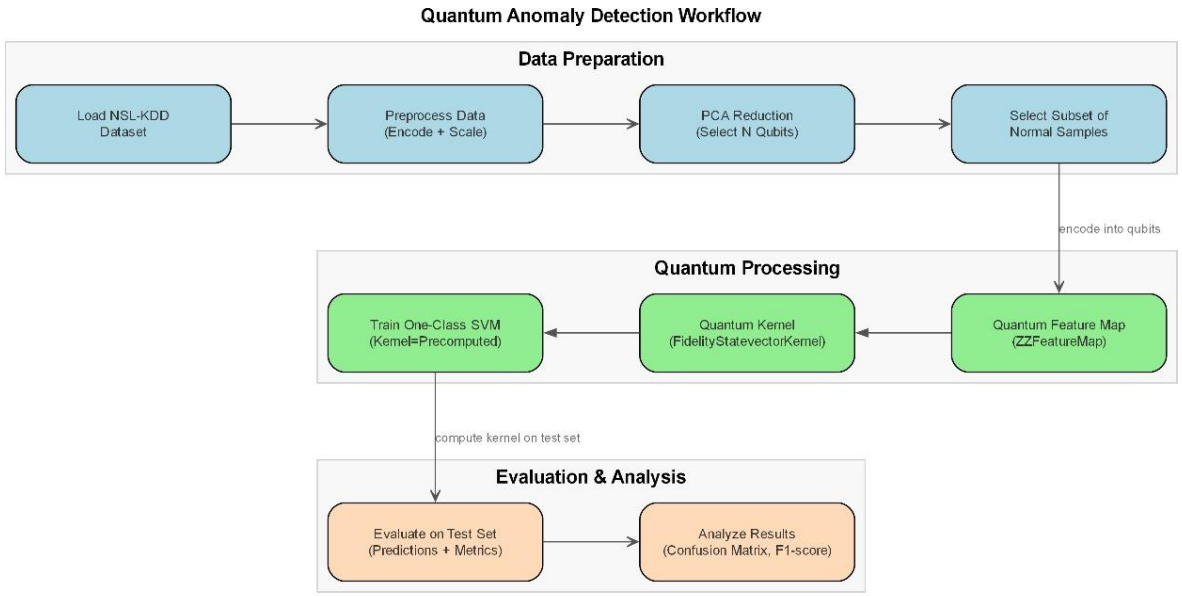


Fig.3. General diagrammatic representation of project workflow

3. Module Design and Implementation

3.1 Module Design

The proposed system is designed as a hybrid classical–quantum anomaly detection framework that integrates traditional machine learning with quantum-enhanced classification. The architecture is divided into a set of well-defined modules, each responsible for a specific stage of the preprocessing, feature engineering, classical model training, quantum kernel computation, and multiclass classification workflow. The overall module design ensures a clear separation of responsibilities, ease of extensibility, and compatibility with both classical and quantum backends.

3.1.1 Overview of System Modules

- **Data Preprocessing Module**
    - Handles dataset ingestion, cleaning, normalization, and feature encoding.
    - Converts categorical attributes into numerical form and scales continuous features.
    - Ensures that the dataset is transformed into a unified representation suitable for both classical and quantum pipelines.
- **Feature Selection and Transformation Module**
    - Performs reduction or selection of relevant features to improve model efficiency.
    - Provides two parallel processing paths:
        I. **Classical Feature Path:** Supports direct numerical features for SVM/SVC.
        II. **Quantum Feature Path:** Applies quantum-compatible transformations such as amplitude or angle encoding.
- **Classical Model Module (SVM / SVC)**
    - Implements Scikit-learn's Support Vector Classifier (SVC) as the baseline classical anomaly detection model.
    - Supports multiclass classification using the built-in One-vs-One (OvO) scheme in SVC.

- Responsible for classical model training, hyperparameter configuration, and prediction.

- **Quantum Kernel Module**

  - Implements Qiskit's QuantumKernel to compute kernel matrices using parameterized quantum feature maps.
  - Encodes classical samples into quantum states using a custom or predefined feature map.
  - Generates K_train and K_test kernel matrices for use with the Quantum SVM classifier.
  - Includes optimizations to handle computationally heavy kernel evaluations.

- **Quantum SVM + One-vs-Rest Classification Module**

  - Performs multiclass anomaly detection using a Quantum SVM (QSVM) wrapped in a One-vs-Rest (OvR) strategy.
  - For each class, trains a binary QSVM classifier using quantum kernel matrices.
  - Combines outputs from all OvR models to produce a final multiclass prediction.
  - Evaluates performance on accuracy, precision, recall, and F1-score.

- **Evaluation and Visualization Module**

  - Generates confusion matrices, classification reports, and performance comparisons.
  - Produces graphical outputs including bar charts, ROC curves (if applicable), and distribution plots.
  - Supports side-by-side comparison of classical vs quantum model behaviour.

3.1.1 Inter-Module Workflows

The modules interact through a structured pipeline:

1. Input dataset → Preprocessing Module
2. Cleaned features pass into Classical Model Module, and Quantum Kernel Module

3. Classical SVC and Quantum OvR-SVM make predictions independently
4. Results are passed to the Evaluation Module for metric computation and visual analysis
5. Final outputs include accuracy, F1-scores, confusion matrices, and comparative insights

This modular design ensures that the classical and quantum components are loosely coupled, allowing easy integration, debugging, and future upgrades, including running the quantum pipeline on actual quantum hardware.

3.2 Implementation

The proposed hybrid anomaly detection system was implemented using Python, combining classical machine learning techniques with quantum kernel-based classification. The implementation follows the modular design described earlier and integrates preprocessing, model training, and performance evaluation into a unified pipeline.

3.2.1 Environment Setup

The system was developed in Python using the following libraries:

a) Scikit-learn for classical SVM (SVC) classification
b) Qiskit Machine Learning for quantum kernel and Quantum SVM
c) NumPy/Pandas for data handling
d) Matplotlib/Seaborn for visualization

All experiments were conducted in Jupyter Notebook/Google Colab using a CPU-based quantum simulator.

3.2.2 Data Preprocessing

The dataset was cleaned, encoded, and normalized before model training. Categorical features were converted into numeric form, and continuous attributes were scaled to maintain consistency across classical and quantum models. A standard train–test split was applied. For

the quantum model, only four target classes (normal, neptune, satan, ipsweep) were retained to reduce quantum computational cost.

3.2.3 Classical SVM Implementation

The classical baseline was implemented using Scikit-learn's SVC with an RBF kernel. The model was trained on the preprocessed feature vectors and evaluated using accuracy, precision, recall, F1-score, and confusion matrix. Training and prediction were extremely fast (<0.05 seconds), highlighting the efficiency of the classical pipeline.

3.2.4 Quantum SVM Implementation

The quantum model uses Qiskit's QuantumKernel and a Quantum SVM (QSVM) wrapped in a One-vs-Rest strategy for multiclass classification. Feature vectors were encoded using a quantum feature map and used to compute kernel matrices (K_train, K_test) on a quantum simulator. Quantum kernel computation is the most time-intensive step, taking ~95 seconds for training and ~80 seconds for testing. The OvR Quantum SVM was then trained on these matrices and evaluated with the same metrics as the classical model.

3.2.5 Evaluation and Visualization

Both models were evaluated using standard classification metrics. Confusion matrices, bar plots, and metric comparisons were generated to visualize performance differences between classical and quantum models.

4. Results and Discussion

4.1 Results

This section presents the performance of the Classical SVM (SVC) model and the Quantum SVM with One-vs-Rest (OvR) classifier on the test dataset. Both models were evaluated using accuracy, precision, recall, F1-score, and confusion matrices.

4.1.1 Classical SVM (SVC) Results

The classical SVC achieved an overall accuracy of **65.9%**.
Key observations include:

- High recall for majority classes, particularly **neptune (1.00 recall)** and **satan (0.97 recall)**.
- Poor performance on minority classes such as **ipsweep (F1 = 0.20)**.
- Weighted F1-score remained strong at **0.80**, indicating good performance on dominant labels.

Selected Metrics

- Accuracy: 0.659
- Macro F1-score: 0.60
- Weighted F1-score: 0.80

The confusion matrix shows that the model correctly classified most *normal* and *neptune* samples but struggled with rare attacks.

SVC Classification Report:

	precision	recall	f1-score	support
normal	0.69	0.90	0.78	452
neptune	0.82	1.00	0.90	215
satan	0.35	0.97	0.51	33
ipsweep	0.11	1.00	0.20	6
micro avg	0.66	0.93	0.77	706
macro avg	0.49	0.97	0.60	706
weighted avg	0.71	0.93	0.80	706

Fig. 4 Classical SVM Result metrics

4.1.2 Quantum SVM (OvR) Results

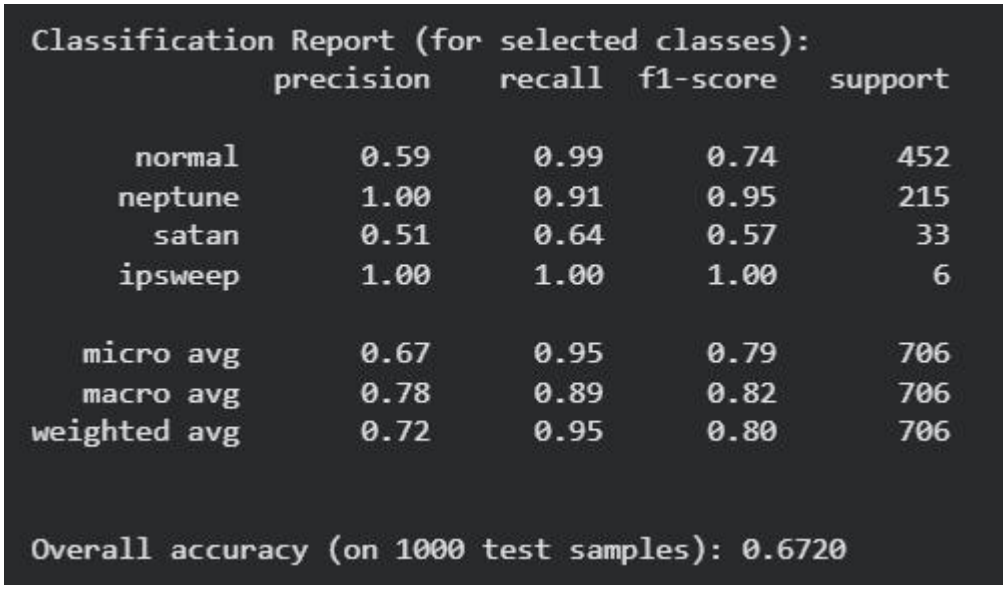
The Quantum SVM achieved a slightly higher accuracy of 67.2%.
Notable improvements were observed in class balance and minority detection:

- Perfect classification for ipsweep (Precision = 1.00, Recall = 1.00).
- Higher macro-level performance (Macro F1 = 0.82) indicating better treatment of all classes.
- Competitive results for *neptune* (F1 = 0.95) and *satan* (F1 = 0.57).

Selected Metrics

- Accuracy: 0.672
- Macro F1-score: 0.82
- Weighted F1-score: 0.80

The confusion matrix shows strong separation across all four classes, with significantly fewer misclassifications for minority attacks.



```
Classification Report (for selected classes):
              precision    recall  f1-score   support

      normal       0.59      0.99      0.74       452
     neptune       1.00      0.91      0.95       215
       satan       0.51      0.64      0.57        33
     ipsweep       1.00      1.00      1.00         6

   micro avg       0.67      0.95      0.79       706
   macro avg       0.78      0.89      0.82       706
weighted avg       0.72      0.95      0.80       706


Overall accuracy (on 1000 test samples): 0.6720
```

Fig. 5 Quantum Kernel (SVM + OvR) result metrics

4.2 Discussion

The comparative analysis between the Classical SVM (SVC) and the Quantum SVM with One-vs-Rest classification reveals several important insights regarding model behaviour, strengths, and limitations.

First, although the overall accuracy of both models is similar, the quantum classifier demonstrates a clear advantage in macro-level performance, achieving a macro F1-score of 0.82, compared to 0.60 for the classical SVM. This indicates that the quantum model provides more balanced classification across all classes, especially in scenarios with uneven class distributions.

A significant observation is the quantum model's superior handling of minority and rare attack classes. The Quantum SVM achieved perfect precision and recall for the *ipsweep* class,

while the classical SVM struggled considerably (F1 = 0.20). This suggests that the high-dimensional feature space generated by quantum feature maps enables better separation of subtle or low-frequency anomalies.

Both models performed strongly on dominant classes such as *normal* and *neptune*, although the classical model achieved slightly higher F1-scores for the majority class. However, it tended to overfit to frequent patterns, resulting in weaker identification of less represented attack types.

From a computational standpoint, the quantum approach incurs substantially higher overhead. Quantum kernel computation required over 170 seconds in total, while the classical model trained and predicted in under 0.05 seconds. This highlights a practical limitation of quantum methods when executed on simulators or without hardware acceleration.

Overall, the results indicate that while classical SVM remains fast and reliable, the quantum-enhanced model offers better class balance, improved minority detection, and stronger generalization across multiple anomaly types. These findings support the potential of quantum machine learning as a complementary tool to classical methods in anomaly detection tasks.

5. Conclusion

This project presented a hybrid approach to multiclass anomaly detection by integrating classical machine learning with quantum kernel-based classification. A Classical SVM (SVC) model and a Quantum SVM using an OvR strategy were implemented and evaluated on selected attack classes of the NSL-KDD dataset. The results demonstrate that both models achieve comparable overall accuracy; however, the quantum model provides significantly better performance on minority classes and achieves a substantially higher macro F1-score.

The Quantum SVM showed strong capability in distinguishing low-frequency anomalies such as ipsweep, achieving perfect classification where the classical model struggled. This highlights the potential advantage of quantum feature spaces in capturing subtle patterns and improving class balance in uneven datasets. On the other hand, the classical SVM remained far more computationally efficient, training and predicting in milliseconds compared to the much longer kernel computation times required by the quantum model.

Overall, the findings suggest that quantum machine learning can complement classical techniques, particularly in scenarios involving class imbalance or complex decision boundaries. While current quantum simulators incur high computational cost, future advancements in quantum hardware are expected to make such models more practical. This project demonstrates that quantum-enhanced approaches can offer meaningful improvements in anomaly detection tasks and represent a promising direction for future research in cybersecurity and machine learning.

6. Future Scope

While this project demonstrates the potential of quantum machine learning for multiclass anomaly detection, several avenues remain open for future enhancement and exploration:

- **Execution on Real Quantum Hardware**
  The current implementation relies on quantum simulators, which introduce significant computational overhead. Running the Quantum SVM on actual quantum hardware (IBM Quantum backends) can provide more accurate performance insights and may reduce kernel computation times.
- **Expansion to Full Dataset Classes**
  Due to quantum computational limitations, only four classes were considered. As quantum hardware evolves, evaluating the model on all NSL-KDD attack categories will offer a more comprehensive assessment of scalability and robustness.
- **Advanced Quantum Feature Maps**
  Exploring deeper or problem-specific feature maps—such as Pauli-based, ZZFeatureMap, or custom ansatz circuits—may further improve class separability and minority-class detection.
- **Hybrid Quantum–Classical Architectures**
  Integrating quantum kernels with classical neural networks or ensemble methods could combine the strengths of both paradigms, enabling more powerful anomaly detection frameworks.
- **Optimization and Noise-Resilient Techniques**
  Future work may include error-mitigation strategies, circuit optimization, and parameter tuning to make quantum models more stable, especially when executed on noisy intermediate-scale quantum (NISQ) devices.

- **Real-Time or Streaming Anomaly Detection**
  Extending the system to handle real-time data streams can enable practical deployment in network security, IoT monitoring, or cloud infrastructure anomaly detection systems.
- **Comparison with Other Quantum Models**
  Investigation of Quantum Neural Networks (QNNs), Variational Quantum Circuits (VQC), and Quantum Distance-Based classifiers could provide deeper insights into which quantum approaches are most suitable for specific anomaly detection scenarios.

7. References

1. Abe, S. (2005). *Support Vector Machines for Pattern Classification* (2nd ed.). Springer.
2. Alvarez-Rodriguez, U., Sanz, M., Lamata, L., & Solano, E. (2017). Supervised quantum learning without measurements. *Scientific Reports*, 7, 13645. https://doi.org/10.1038/s41598-017-14340-8
3. Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.
4. Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3), 273–297. https://doi.org/10.1007/BF00994018
5. Dua, D., & Graff, C. (2017). *UCI Machine Learning Repository*. University of California, Irvine. https://archive.ics.uci.edu
   (NSL-KDD is derived from KDD'99 in this repository.)
6. Han, S., Wang, J., & Fan, L. (2023). Quantum machine learning for anomaly detection: A review. *Quantum Information Processing*, 22(8), 311. https://doi.org/10.1007/s11128-023-03951-9
7. Havlíček, V., Córcoles, A. D., Temme, K., Harrow, A. W., Kandala, A., Chow, J. M., & Gambetta, J. M. (2019). Supervised learning with quantum-enhanced feature spaces. *Nature*, 567(7747), 209–212. https://doi.org/10.1038/s41586-019-0980-2
   (Foundational paper behind quantum kernel methods used in QSVM.)
8. IBM Quantum. (2023). *Qiskit Machine Learning Documentation*. https://qiskit.org/documentation/machine-learning/
9. Khan, A., Shams, R., & Khan, A. (2019). A survey on the NSL-KDD dataset: Re-sampling and classification using SVM. *International Journal of Computer*

Applications, 975, 8887.
(Useful reference for SVM on NSL-KDD.)

10. Schuld, M., Bocharov, A., Svore, K. M., & Wiebe, N. (2019). Circuit-centric quantum classifiers. *Physical Review A*, 101(3), 032308. https://doi.org/10.1103/PhysRevA.101.032308
11. Schuld, M., & Petruccione, F. (2021). *Machine Learning with Quantum Computers* (2nd ed.). Springer.

Quantum Anomaly Detection (Cybersecurity/IoT) using IBMQ

Name: Madhur Pophale
Reg. No: 22BCE2478
Course: Cloud Computing
Course Code: BCSE408L
Faculty: Prof. Sankar Ganesh L

Problem Statement

- Anomaly detection is critical in domains such as cybersecurity, healthcare, and finance — identifying unusual patterns that may indicate attacks, faults, or fraud.
- Most traditional anomaly detection systems are binary, labeling data as normal or anomalous, but real-world data often includes multiple types of anomalies → requiring a multiclass approach.
- Classical ML models can struggle with:
  - High-dimensional and non-linear data
  - Imbalanced class distributions
  - Poor generalization on unseen anomaly types
- Quantum computing introduces a new paradigm with the potential for:
  - Exponential feature space representation (via superposition and entanglement)
  - Enhanced pattern separation for complex decision boundaries
  - Potential computational speedup in learning and inference

Motivation

- The motivation behind this project is to find out and explore whether Quantum Machine Learning or QML can improve the Multiclass anomaly detection to detct multiple types of Cybersecurity attacks with a performance that is either equivalent or better than Classical Multiclass anomaly detection methods

Objectives

- Build a multiclass anomaly detection framework using both classical and quantum models.
- Compare their performance, scalability, and interpretability.
- Analyze advantages and challenges of quantum models in anomaly detection.

Dataset & Preprocessing

- Dataset: NSL-KDD (Training and Testing subset)
- Dataset Dimensions:
  - KDDTrain+.txt: 43 Features, 126K Entries
  - KDDTest+.txt: 43 Features, 22.5K Entries
- Source: University of New Brunswick

- Classes involved for benchmarking: Normal, Neptune, Satan, ISweep

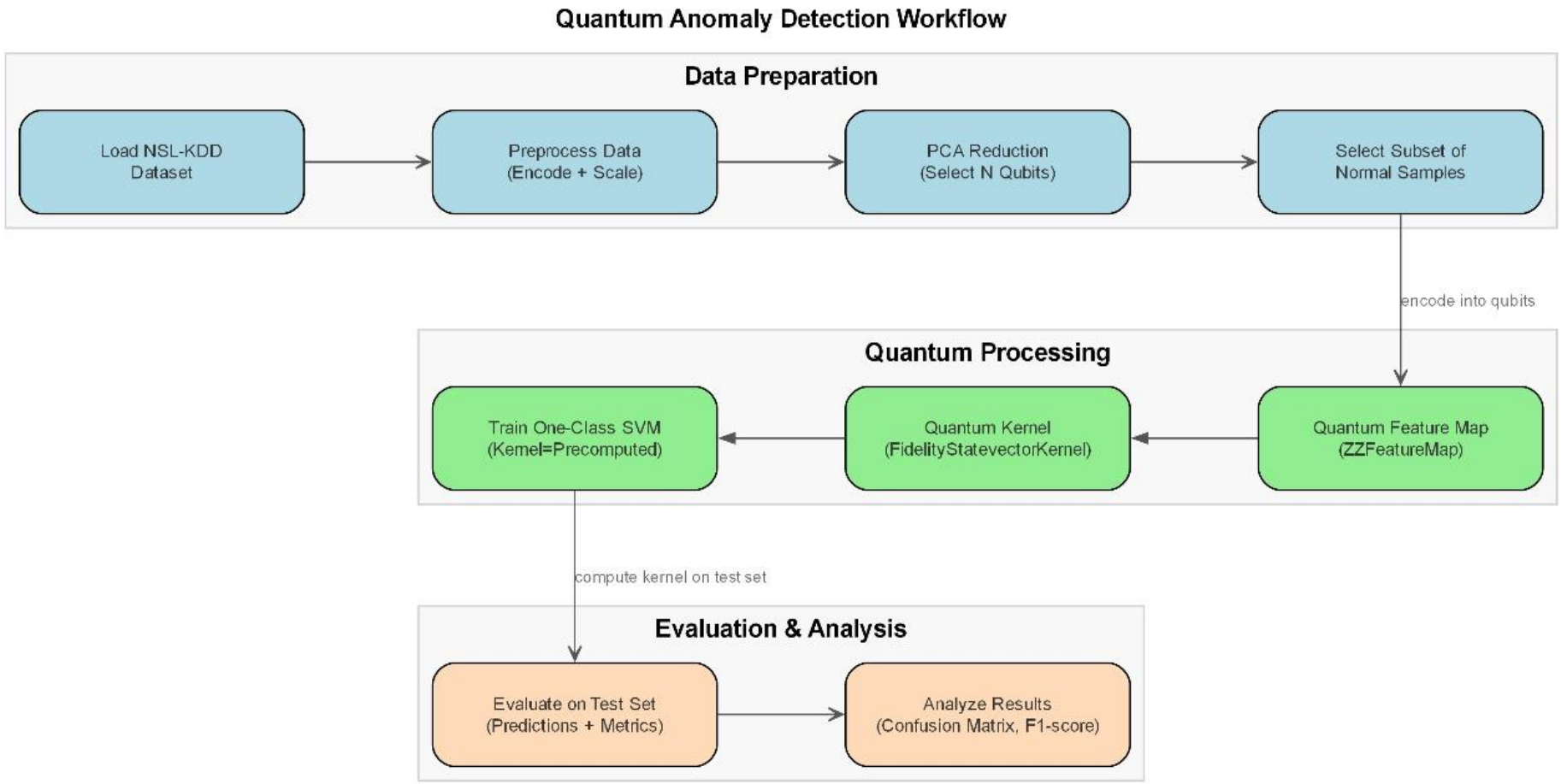
- Preprocessing steps:
  - Normalization/scaling
  - Feature selection/reduction (PCA or Quantum Feature Map)
  - Encoding for quantum model (Amplitude or Angle encoding)

System Requirements

- Software:
  - IBM Qiskit (qiskit, qiskit-aer, qiskit-machine-learning)
  - Numpy, Pandas, Scikit-learn, Matplotlib, Seaborn
  - Python 3.9 or higher, Google Colab Environment

- Hardware:
  - Intel Core i3, 8GB Memory
  - Google Colab TPU, 12GB Memory

Workflow Diagram



Results and Analysis

Performance Metrics for Quantum Kernel

```
Classification Report (for selected classes):
              precision    recall  f1-score   support

      normal       0.59      0.99      0.74       452
     neptune       1.00      0.91      0.95       215
       satan       0.51      0.64      0.57        33
     ipsweep       1.00      1.00      1.00         6

   micro avg       0.67      0.95      0.79       706
   macro avg       0.78      0.89      0.82       706
weighted avg       0.72      0.95      0.80       706


Overall accuracy (on 1000 test samples): 0.6720
K_test shape: (1000, 1600)  computed in 80.7 s

Confusion Matrix (rows=actual, cols=predicted) for selected classes:
[[449   0   3   0]
 [ 18 196   1   0]
 [ 12   0  21   0]
 [  0   0   0   6]]
```

Performance Metrics for Classical Model

```
SVC Classification Report:
              precision    recall  f1-score   support

      normal       0.69      0.90      0.78       452
     neptune       0.82      1.00      0.90       215
       satan       0.35      0.97      0.51        33
     ipsweep       0.11      1.00      0.20         6

   micro avg       0.66      0.93      0.77       706
   macro avg       0.49      0.97      0.60       706
weighted avg       0.71      0.93      0.80       706

SVC train_time=0.02s predict_time=0.01s
SVC Accuracy: 0.659
SVC Confusion Matrix (selected classes):
 [[406   1  41   4]
 [  0 215   0   0]
 [  1   0  32   0]
 [  0   0   0   6]]
```

Discussion

- The Quantum SVM+OvR Model achieved the accuracy levels of 67.2% when compared with the classical SVM which yields an accuracy of 65.9%
- This Proof of concept model tells us that if not for hardware limitations, the Quantum SVM+OvR model would outperform the classical methods, showing better generalixation capabilities on Multiclass Data
- The Quantum Classifier achieved a perfect precision and recall score for a minority “isweep” class, which is a very common challenege in an Classical Anomaly Detection Model
- The Quantum approach took about ~95s for training of the Quantum Kernel and also about ~81s for its testing whereas the timings show that the classical approach took only about ~0.5s for entire training and prediction on the same data subset as the quantum model
- This tells us that the Quantum accuracy gains comes with a significantly higher computational overhead

Conclusion

- In cunclusion while the classical model remains far more simpler and faster for deployment and setup, the Quantum Approach provides us an improved class balance and a rare-class recognition, but at the cost of extensive optimization and hardware acceleration to scale efficiently

- The current hardware capabilities of the IBMQ quantum computers and simulators make it difficult to scale and implement these techniques at Utility scale, but this Proof of Concept model serves as a proof that it is not only possible, but may also be adapted in the coming future

THANK YOU