ISSN: 2690- 912X

Research Article

Journal of Genetic Engineering and Biotechnology Research

Exploring Pauli Configurations in Quantum Kernels for Enhanced Binary Classification

Luis Gerardo Ayala Bertel[1]*, Ashish Patel[2] and Jayakumar Vaithiyashankar[3]

[1]*Department of Mathematics, Faculty of Exact and Natural Science, Cartagena University - Colombia*

[2]*Sr. Data Scientist, Cygnet Infotech Pvt Ltd - India*

[3]*School of Computer Science, Presidency University, Bengaluru – India*

***Corresponding Author**
Luis Gerardo Ayala Bertel, Department of Mathematics, Faculty of Exact and Natural Science, Cartagena University – Colombia.

Submitted: 2023, Oct 10; **Accepted**: 2023, Nov 06; **Published**: 2023, Dec 12

Citation: Bertel, A. G. L., Patel, A., Vaithiyashankar, J. (2023). Exploring Pauli Configurations in Quantum Kernels for Enhanced Binary Classification. *J Gene Engg Bio Res,* 5(3), 232-242.

Abstract

This study pretends to set one's sights on advancing the understanding of quantum machine learning's applicability in binary tasks. By systematically assessing quantum kernels and their performance, the primary objective is to determine and evaluate how different Pauli configurations—precisely, ZZ, ZY, and a custom configuration (Z, YY, ZXZ)—impact execution accuracy running in simulator and 7-qubits IBM quantum hardware. The paper encompasses a robust methodology, including data preprocessing, quantum kernel creation, and integration into classical SVMs. It provides essential guidance for discussing the strengths and limitations of quantum methods, highlighting their potential as a valuable tool for developing accurate diagnostic models.

Keywords: Quantum Kernels, Support Vector Machines (SVMs), Dataset, Dimensionality Reduction, Feature Scaling, Quantum Feature Maps

1. Introduction

Within the domain of medical diagnostics, breast cancer classification stands a binary problem, where: *Malignant* (*cancerous*) tumors are represented as the class labeled "*1*". *Benign* (*noncancerous*) tumors are represented as the class labeled "0". The goal of using this dataset is typically to build a model that can accurately classify breast tumors based on the provided features. It is a widely recognized dataset for practicing and evaluating the algorithm's performance [1]. Traditional classification methodologies have valiantly advanced our understanding, achieving commendable milestones. Nevertheless, the relentless pursuit of computational efficiency continues unabated, steering us toward exploring more precise and timely solutions. Although classical artificial intelligence has reached the point that a model is capable of analyzing a breast tomosynthesis and predicting from it whether a patient may develop cancer within the next five years quantum computing is on the brink of a computing revolution [2]. It represents a harbinger of a new era in problem-solving and has the potential to guide us through the complexities and enigmatic principles of quantum mechanics. Quantum algorithms, with their unique capabilities based on superposition and entanglement, are envisioned to achieve the ability to dissect vast datasets such as breast cancer with unparalleled efficiency. They offer a tantalizing promise: faster, more accurate diagnoses that project an entirely new paradigm shift [3].

While quantum algorithms hold great promise, their incorporation into practical quantum hardware is the immersion of scientific validation. Real quantum devices are not without challenges: noise, limited qubit connectivity, and gate errors present formidable obstacles. Empirical experimentation with real quantum hardware is indispensable. This phase of our research anchors the theoretical elegance of quantum algorithms in real-world quantum processors.

Figure 1: The IBM Falcon R5.11H Quantum Processor

A Quantum Marvel with 7 Real Qubits, Scalable to 27 Qubits Through its Innovative Multi-Chip Stack Architecture, Empowering Quantum Computing Advancements [4].

1.1 Paradigm Shift

In binary classification, discerning between two classes is paramount. To understand this quantum advantage, we must delve into the theoretical underpinnings of quantum algorithms and their implications for binary classification.

Classical binary classification typically deals with datasets represented as vectors in a high dimensional feature space [5]. Each data point is denoted as xi, and the corresponding binary label is y_i, where $y_i \in \{0, 1\}$ represents the two classes. The goal is to achieve a decision boundary or hyperplane that best separates these classes [6]. Finding the optimal weights and bias often relies on iterative optimization techniques such as gradient descent. However, as datasets grow in complexity and dimensionality, these classical methods may encounter computational limitations. Qubits (*quantum bits*) are the fundamental information units in Quantum computing [7]. Qubits can exist in a superposition of states, unlike classical bits, which can be either 0 or 1.

Observation: It is crucial to understand that superposition does not allow us to directly place multiple distinct inputs into a single qubit. A qubit can encode a complex combination of states. Let us consider a quantum state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, where $|0\rangle$ and $|1\rangle$ are the basis states representing classical bits 0 and 1, respectively. The complex amplitudes α and β determine the probabilities of measuring the qubit in the 0 or 1 state [8]. This superposition allows us to perform certain quantum algorithms, such as quantum parallelism, where operations can be applied to multiple states simultaneously.

In the context of binary classification or other quantum machine learning tasks, qubits can represent quantum feature vectors that encapsulate information about the data. While qubits enable quantum computers to process information differently from classical computers, it does not directly represent multiple inputs in a single qubit. The power of quantum computing lies in its ability to manipulate and process quantum states.

1.2 Quantum Principles

By assuming the above observation, it's possible to say, the superposition of data points allows quantum systems to explore multiple data instances simultaneously. This parallelism can significantly accelerate computations when dealing with large datasets, as quantum algorithms can process data points in a superposition, potentially can reduce computational complexity and time [9]. Quantum entanglement is another foundational principle that goes beyond classical correlations. When two or more qubits become entangled, the state of one qubit becomes intrinsically linked to the state of another, regardless of the physical distance between them. This unique correlation introduces a level of interconnectedness that can be harnessed for enhanced data processing [10]. It is mathematically represented as:

$$|\psi\rangle = \alpha|00\rangle + \beta|01\rangle + \gamma|10\rangle + \delta|11\rangle$$

$|00\rangle$, $|01\rangle$, $|10\rangle$, and $|11\rangle$ are the four possible entangled states of two qubits. The complex amplitudes α, β, γ and δ describe the correlations between these states. Entanglement is particularly relevant in the context of high-dimensional data processing. It allows quantum systems to discover intricate relationships and correlations between data points that might not be seen in classical representations. This capability can lead to enhanced classification accuracy, especially when dealing with complex binary classification problems, such as medical diagnostics.

2. Theoretical Overview

Kernel evaluations are essential for mapping data points into a higher-dimensional feature space [11]. The most common kernel function used is the Radial Basis Function (RBF) kernel:

$$\mathrm{K}(x, x') = \exp\left(-\frac{\|x-x'\|^2}{2\sigma^2}\right) \text{ where;}$$

- $K(x, x')$ represents the *RBF* kernel function.
- x and x' are the input vectors or data points.
- σ is a hyperparameter that controls the "*spread*" or "*width*" of the Gaussian function.
- $\|x - x'\|^2$ denotes the Euclidean distance squared between the input vectors x and x'.

Quantum computing introduces a paradigm shift in kernel evaluation [3]. The quantum state vector $|\psi\rangle$ facilitates the parallel computation of kernel evaluations for multiple data points simultaneously:

$$|\psi\rangle = \sum_i \alpha_i |x_i\rangle$$

In the above equality, $|x_i\rangle$ signifies the quantum state of data point x_i, and α_i are complex amplitudes. Quantum algorithms enable the efficient calculation of inner products between these quantum states, leading to quantum kernel evaluations.

Quantum kernel is created by mapping a classical feature vector $\vec{x}$ to a Hilbert space using a quantum feature map $\phi(\vec{x})$.

$$\mathrm{K}_{ij} = |\langle\phi(\vec{x_i})|\phi(\vec{x_j})\rangle|^2 \text{ where;}$$

- K_{ij} is the kernel matrix.
- $\vec{x_i}$ and $\vec{x_j}$ are dimensional inputs.
- $\phi(\vec{x})$ is the quantum feature map.
- $|\langle a|b\rangle|^2$ denotes the overlap of two quantum states a and b.

a. Quantum Parallelism

Quantum computing's primary advantage lies in its ability to compute these kernel evaluations in parallel across multiple data points, thanks to superposition; entanglement introduces a new level of correlation between data points, allowing the quantum model to uncover intricate patterns that might remain hidden in classical representations [12]. Classical computers, by contrast, must calculate them sequentially. For instance, the class *SVM*

exploits the outputs of these quantum phenomena and computes kernel evaluations. This parallelism can drastically reduce computational complexity making it possible to handle high-dimensional datasets efficiently [13].

3. Methodology

The guidelines in this section are indispensable for the quantum experiment process.

a. Quantum-Ready Preparation

The initial step in our data preprocessing pipeline addresses dimensionality reduction [14]. It is uppermost for quantum workflows due to the intrinsic quantum computing resources limitations. Quantum computers exhibit significant computational power but are currently constrained by the number of qubits and quantum gates they can efficiently handle. By reducing the dimensionality of the dataset using Principal Component Analysis (*PCA*) to the number of real qubits, which are seven, in this study, we mitigate the quantum resources required for subsequent operations. This enhances quantum algorithm efficiency, as it reduces the number of qubits and quantum operations needed to process the data.

b. Ease Quantum Convergence

Normalization plays an essential role in quantum data preprocessing by standardizing the dataset. In quantum algorithms, ensuring data consistency and convergence is crucial to operate reliably on diverse datasets while avoiding numerical instability. When data exhibits widely varying scales, quantum computations can become unstable or converge slowly [15]. Normalization mitigates this issue, fostering the convergence of quantum algorithms by ensuring that the data operates within a stable range.

c. Compatibility with Quantum Feature Maps

Scaling the dataset to a predefined range is another pivotal preprocessing step for quantum computations. Quantum feature maps and quantum kernels, central to many quantum machine learning algorithms, often depend on the input data's range [16]. Scaling the data aligns it with the expectations of quantum components, ensuring compatibility. Non-observance to scale the data appropriately could lead quantum computations to yield inaccurate results.

d. Managing Computational Resources

The selection of train and test size parameters determines the dataset's size, directly impacting quantum computation complexity. By limiting the dataset size, we strike a balance between computational feasibility and classification accuracy [17]. This step is focal to check, as it allows us to tailor their analyses to the available quantum hardware capabilities and desired effectiveness since data selection ensures that quantum resources are utilized efficiently, making quantum computations more manageable.

4. Results

We present the outcomes of our quantum-based classification approach. We explore various aspects of classification performance, including accuracy enhancement, insights from confusion matrices, and a breakdown of correct and incorrect classifications.

a. Quantum Accuracy Enhancement

We delve into the empirical demonstration of quantum advantage in the context of our quantumaugmented (*SVMs*). The quantum exploration is empirically substantiated through specific Pauli configurations namely '*ZZ*', '*ZY* ' and a custom configuration ['Z', '*YY*', '*ZXZ*']. We try to understand the scientific rationale behind each Pauli configuration, including the quantum feature maps involved and how these configurations are intricately linked to the quantum accuracy enhancement [18].

i. Pauli Configuration '*ZZ*'

The 'ZZ' Pauli configuration signifies utilizing the ZZFeatureMap, a quantum feature map designed to capture pairwise correlations between data points. In this configuration, the quantum feature map is defined as follows:

$\Phi ZZ(x) = Q_i \neq j \exp(i\theta ijZi \otimes Zj)$ where;

- $\Phi ZZ(x)$ represents the function.
- $Q_i \neq j$ denotes the product over all pairs (i, j) where i is not equal to j.
- $i\theta ijZi \otimes Zj$ represents the term inside the product, with $\otimes$ denoting the tensor product.

Z_i and Z_j are Pauli-Z operators acting on qubits i and j, and θij represents the entangling parameter between qubits i and j. This feature map captures second-order correlations between the data features, allowing the quantum SVM to learn intricate relationships within the dataset.

Analysis Results for Pauli Configuration 'ZZ'	
Accuracy (Pauli: ZZ):	0.9300
Recall (Pauli: ZZ):	0.9524
F1-score (Pauli: ZZ)	0.9449

While 'ZZ' Pauli configuration leverages the ZZFeatureMap capturing pairwise correlations. The high accuracy (0.9300) demonstrates that this configuration effectively captured second-order correlations within the data. The high recall (0.9524) suggests that the quantum model excelled at identifying true positives, which is crucial in medical diagnosis. The F1-score (0.9449) reflects a balanced precision-recall trade-off, all while executed on a quantum simulator.

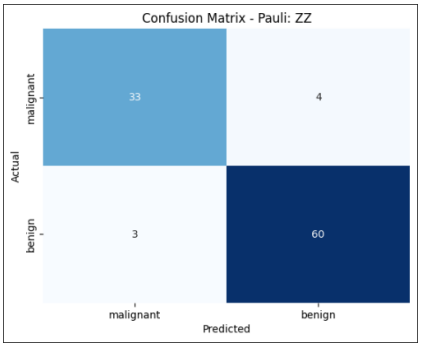

Figure 2: The presence of 33 *TP* indicates that the model

Successfully identified 33 malignant tumors correctly. There were 4 *FP*, which means that the model mistakenly classified 4 benign tumors as malignant. Additionally, there were 3 *FN*, signifying that the model failed to identify 3 malignant tumors correctly. Lastly, there were 60 *TN*, indicating that the model correctly identified 60 benign tumors.

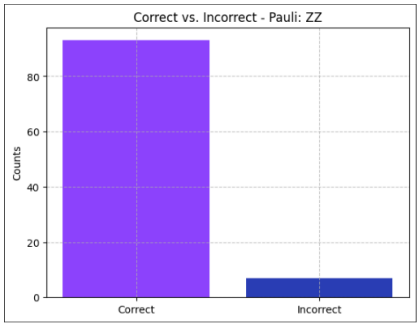

Figure 3: The model successfully discerned and classified the experiment data with a remarkable accuracy of 93% on quantum simulation.

ii. Pauli Configuration 'ZY'

The '*ZY* ' Pauli configuration employs the ZFeatureMap, which focuses on capturing correlations between data points and the individual features of the dataset. The feature map for '*ZY* ' can be expressed as:

$\Phi_{ZY}(x) = \prod_i \exp(i\theta_i Z_i \otimes I)$ where;

- $\Phi_{ZY}(x)$ represents the function.
- $\prod_i$ denotes the product over all terms involving i.
- $i\theta_i Z_i \otimes I$ represents the term inside the product, with $\otimes$ denoting the tensor product, and I representing the identity operator.

In this configuration, Zi is a Pauli-Z operator acting on qubit i, and θi represents the entangling parameter for feature i. This feature map allows the quantum SVM to highlight the significance of individual features while considering their interplay in the classification process.

Analysis Results for Pauli Configuration 'ZY'	
Accuracy (Pauli: ZY):	0.9400
Recall (Pauli: ZY):	0.9841
F1-score (Pauli: ZY)	0.9538

'*ZY* ' Pauli configuration employs the ZFeatureMap, which captures correlations between data points and individual features. The high accuracy (0.9400) showcases the configuration's capability to effectively consider feature importance and data correlations. The impressive recall (0.9841) suggests that this configuration excelled at identifying cancerous instances (true positives), all while executed on a quantum simulator.

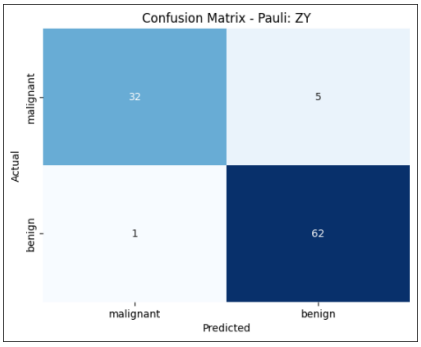

Figure 4: The presence of 32 *TP* indicates that the model

Successfully identified 32 malignant tumors correctly. However, there were 5 FP, which means that the model mistakenly classified 5 benign tumors as malignant. Additionally, there was 1 *FN*, signifying that the model failed to identify 1 malignant tumor correctly. Lastly, there were 62 *TN*, indicating that the model correctly identified 62 benign tumors.

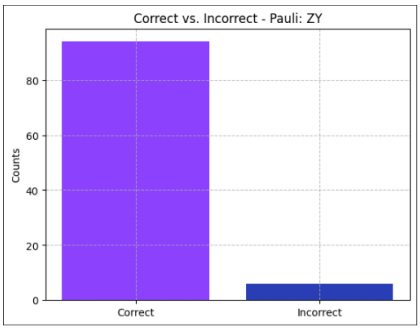

Figure 5: The model successfully discerned and classified the experiment data with a remarkable accuracy of 93% on quantum simulation.

iii. Custom Pauli Configuration

[‘*Z*’,‘*Y Y* ’,‘*ZXZ*’]

The custom Pauli configuration [‘*Z*’, ‘*YY*’, ‘*ZXZ*’] is a versatile choice that combines the Pauli-Z operator, Pauli-Y operators, and a combination of Pauli-Z and Pauli-X operators. The custom feature map can be represented as:

$\Phi_{[Z',YY',ZXZ']}(x) = \prod_i \exp(i\theta_i Z_i \otimes I + i\phi_i Y_i \otimes Y_i + i\chi_i Z_i \otimes X_i)$ where;

- $\Phi[Z',Y\,Y',ZXZ'](x)$ represents the function.
- Q_i denotes the product over all terms involving i.
- $i\theta_i Z_i \otimes I, i\phi_i Y_i \otimes Y_i$, and $i\chi_i Z_i \otimes X_i$ represent the terms inside the product, with $\otimes$ denoting the tensor product. I represents the identity operator.

Y_i represents Pauli-Y operators, X_i represents Pauli-X operators, and θ_i, ϕ_i, and χ_i are the corresponding entangling parameters.

This setup experiment aims to hypothetically balance both first-order correlations and complex higher-order interactions within the data [18].

Analysis Results for Custom Pauli Configuration	
Accuracy (Pauli: [‘Z’, ‘YY’, ‘ZXZ’]):	0.6300
Recall (Pauli: [‘Z’, ‘YY’, ‘ZXZ’]):	1.0000
F1-score (Pauli: [‘Z’, ‘YY’, ‘ZXZ’])	0.7730

This custom Pauli configuration [‘*Z*’, ‘*YY*’, ‘*ZXZ*’] combines Pauli-Z, Pauli-Y, and Pauli-X operators, looking or pretending to offer a balanced approach to capturing correlations within the data. While this configuration got high recall (1.0000) probably due bias, the accuracy (0.6300) and F1-score (0.7730) are relatively lower. The lower accuracy suggests that the model may have incurred more false positives or negatives compared to the other configurations. The high recall indicates all true cancer cases evaluated were identified but potentially at the expense of higher false positives, all while executed on a quantum simulator.

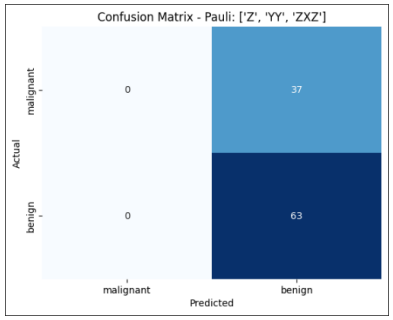

Figure 6: This particular confusion matrix pattern, where there is no TP and a relatively high number of FP, suggests a potential bias in the model. Bias models occur when they are skewed towards one class, typically the majority class, and perform poorly on the minority class. It appears that Pauli [‘*Z*’, ‘*YY*’, ‘*ZXZ*’] configuration may have a bias towards classifying tumors as malignant, resulting in a high number of false positives. The bias could be due to several factors, including the choice of the Pauli feature map, hyper parameters of the SVM, or characteristics of the breast cancer dataset. It is essential to investigate and fine-tune these factors to address the bias and improve the model’s performance, especially in scenarios where correctly identifying malignant tumors.

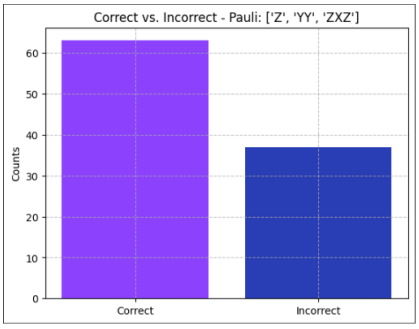

Figure 7: The model study has unveiled an accuracy of 63% on quantum simulation.

b. ROC and AUC Calculation Values

The Receiver Operating Characteristic and Area under the Curve presented in the following results were calculated by leveraging the quantum feature maps, quantum kernels, and classical *SVMs* with quantum kernels as outlined. Here are the values that were determined:

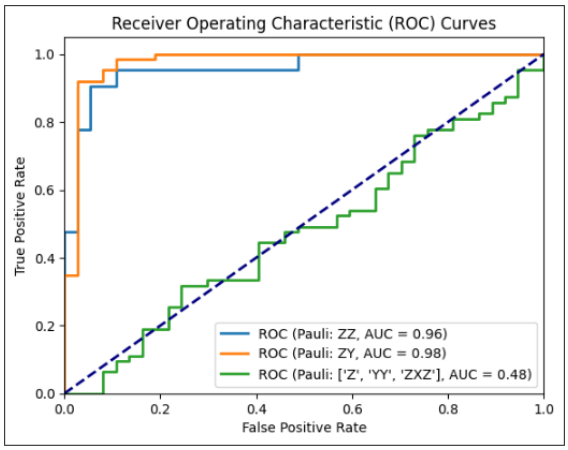

Figure 8: The code calculated the false positive rate (*FPR*) and true positive rate (*TPR*) at various decision thresholds. These rates were used to plot the ROC curve. The AUC was then calculated as the area under this curve.

The high *AUC* value (96) indicates that the model's true positive rate (sensitivity) is considerably higher than its false positive rate (1-specificity), showcasing its ability to correctly classify cancer cases while minimizing misclassifications. In quantum terms, this suggests that the quantum feature map and quantum kernel derived from the '*ZZ*' Pauli configuration have effectively encoded and processed the dataset's information, enabling the quantum SVM to make perceptible predictions. In quantum terms, the high AUC value (98) implies that the quantum model's feature map effectively utilizes both data point correlations and individual feature information to make highly accurate predictions. Quantum *SVM* based on the '*ZY* ' configuration has learned to navigate the complex relationships within the data, leading to superior classification performance.

The scenario with a custom Pauli configuration [*'Z', 'YY', 'ZXZ'*] resulting in an AUC value (48) can be indicative bias or an imbalance in the quantum data encoding or processing. This bias may arise from several quantum-related factors:

• Our custom configuration might not be wellsuited for the specific quantum problem we are trying to solve. Quantum algorithms often rely on carefully selecting quantum gates (*Pauli operators*) to encode and manipulate data. If these operators do not capture the relevant information in the data, it can lead to a biased model.
• Quantum systems exhibit entanglement and interference effects which can be both an advantage and a disadvantage. '*YY* ' operator represents a form of entanglement between qubits, which might not be appropriate for the experiment. This entanglement could lead to correlations that are not relevant to the classification task, introducing noise and bias into the model [19].
• Quantum computing algorithms involve various design options, such as ansatz (*quantum circuit structure*), optimization algorithms, and encoding techniques, among others. If these options are not appropriate or if an experiment is performed just to find out what results it can give, biases can occur.

c. Importance of Quantum Simulators

Quantum simulators like "*statevector_simulator*" offer a controlled and resource-efficient environment for developing, debugging, and validating quantum algorithms. They ensure the quantum model behaves as expected before transitioning to actual quantum hardware [20]. This step is crucial to avoid resource wastage and validate the correctness of the quantum code. Simulators serve as benchmarks to assess the potential quantum advantage of an algorithm or configuration when compared to classical methods. This benchmarking provides a baseline for evaluating quantum approaches' effectiveness and helps make informed decisions about their real-world deployment.

Running on quantum simulators allows us to estimate the computational requirements of quantum algorithms. This estimation is essential for planning resource allocation, especially when transitioning to quantum hardware with limited qubits and gates. It helps in optimizing resource usage and managing the cost-effectiveness of quantum computations.

5. Real-World IBM Quantum Provider

The selection of an appropriate quantum backend is a crucial aspect of any quantum computing experiment, as it directly impacts the results' quality and the research objectives' feasibility.

In our real-world experiment, we carefully considered the choice of the quantum backend, and we opted for the "*ibmq_jakarta*" is hosted on the Falcon r5.11H processor, which is a state-of-the-art quantum computing system designed for practical quantum computations. This processor has several important attributes that make it an attractive choice for our experiment.

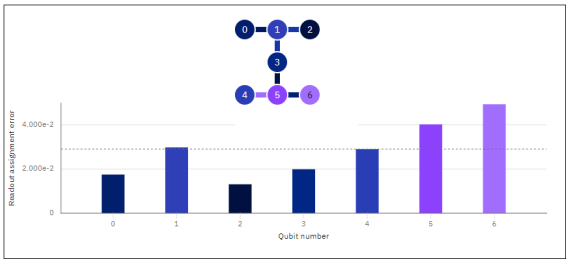

Figure 9: Readout assignment errors in a quantum processor refer to inaccuracies or imperfections in the measurement process

That occur at the end of a quantum computation. These errors can occur when mapping the quantum state of qubits to classical bits, which is necessary to obtain the measurement outcomes.

When a qubit is in a particular quantum state (e.g.,$|0\rangle$ or $|1\rangle$), but the measurement outcome (*readout*) does not accurately reflect this state, then there is a discrepancy between what the qubit's quantum state should be and what is observed as the measurement result [21]. Assignment errors are related to the mapping of qubit states to classical bits. When assignment errors occur, the classical bit recorded from the qubit's measurement does not correctly correspond to the qubit's quantum state. Quantum processors often have built-in calibration procedures to reduce readout errors. These calibrations involve adjusting the readout assignments for each qubit to align measurement outcomes with the actual qubit states. Falcon r5.11H Processor (*quantum hardware*) and statevector simulator (*quantum simulator*) involve fundamental differences related to hardware constraints, noise-error handling, execution

time, algorithm design, and other factors. We often use simulators during algorithm development and debugging before transitioning to quantum hardware for real-world experiments.

a. Runtime Service

The runtime_service object represents the connection to IBM Quantum's cloud-based quantum computing resources. The jobs () function is used to retrieve a list of quantum jobs that have been submitted to the specified quantum backend; backend_name="*ibmq_jakarta*" is the specific quantum processor we use for our quantum computations.

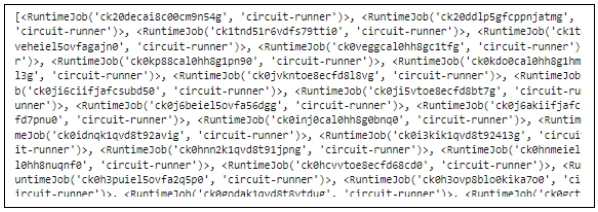

Figure 10: List that represent a collection of quantum jobs submitted to the 'I' quantum backend, a real-world 7-qubit Falcon r5.11H quantum processor. Each <Runtime Job> entry corresponds to a specific quantum computation task.

This list of jobs is critical because it provides insight into the status, progress, and results of quantum computations performed on real quantum hardware.

a. Iterating Through Jobs List and Retrieve Results

Each set of quantum circuits we submit to a quantum computer is assigned a unique Job ID. The Job ID is a way to track and identify a specific computation on the quantum computer. In the subsequent output, we show multiple Job IDs, each corresponding to a different set of quantum circuits.

Figure 11: List that represent a collection of quantum jobs submitted to the '*ibmq_jakarta*' quantum backend, a real-world 7-qubit Falcon r5.11H quantum processor. Each <RuntimeJob> entry corresponds to a specific quantum computation task.

The above figure provides detailed information about each quantum computation job, including the backend used, qobj_id (Quantum Object ID), job_id (Job ID), success status, and the actual results of the computation.

Backend Name: The "*backend_name*" field tells which IBM quantum computer was used for the computation.

Backend Version: This field specifies the version of the quantum computer backend.

Qobj ID: The "*qobj_id*" is a unique identifier for the quantum object used in the computation.

Job ID: This field repeats the Job ID mentioned earlier.

Success: The "*success*" field indicates whether the quantum computation was successful (*True*) or encountered errors (*False*).

Results: This section provides the actual measurement results obtained from running the quantum circuits. It includes information about the measured qubits, the number of shots (*measurements*) taken, and the measurement outcomes (*counts*) for each possible outcome.

Experiment Header: This section contains information about the quantum circuits that were executed in the job.

Clbit_Labels: Labels for classical bits used in measurement outcomes. creg_sizes: Sizes of classical registers.

Global Phase: The global phase applied to the quantum circuit.

Memory Slots: The number of classical memory slots used.

Metadata: Additional metadata associated with the experiment.

n_qubits: The total number of qubits used in the quantum circuit.

Name: The name of the quantum circuit or experiment. *qreg_sizes*: Sizes of quantum registers.

Qubit_Labels: Labels for individual qubits.

b. Classification Accuracy

The classification accuracy achieved in this experiment, utilizing real quantum hardware ('*ibmq_jakarta*') and employing only Pauli *ZY* feature maps, was determined to be 0.63. This section elaborates on the factors influencing this accuracy and underscores potential sources of bias that may have contributed to this result. Pauli *ZY* feature maps were chosen due to their effectiveness, as they had previously yielded an accuracy score of 0.94 in the statevector simulator, justifying their selection for quantum hardware experimentation. Quantum computations are inherently probabilistic and sensitive to noise [22]. The quantum kernel computation, which forms the core of the machine learning model, can yield different results due to quantum variability. This could contribute to the distribution of data points across TP, FP, FN, and TN categories.

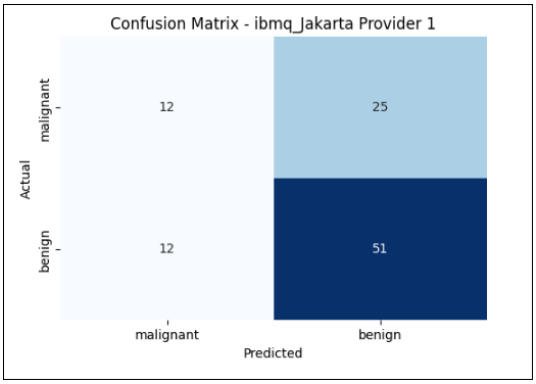

Figure 12: Confusion matrix revealed an intriguing pattern

12 True Positives (*TP*), 25 False Positives (*FP*), 12 False Negatives (*FN*), and 51 True Negatives (*TN*). This seemingly unusual result can be attributed to several factors and hints at potential sources of bias within the experiment.

Algorithmic bias can persist even in the context of quantum algorithms. The selection of quantum kernel construction methods, classical SVM settings, and hyperparameter optimization can wield substantial influence over classification outcomes. Prudent consideration and exhaustive optimization of these aspects remain pivotal in order to try to mitigate it.

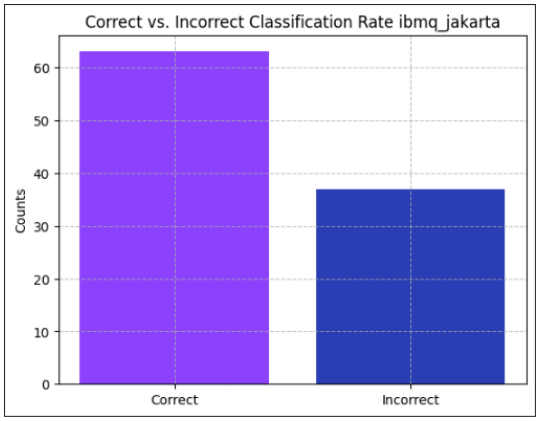

Figure 13: binary classification model achieves an accuracy of 0.63, marking a step forward in our explore for quantum computing advancements.

While the state vector simulator offers noise-free and idealized measurements, it is essential to recognize that real quantum hardware experiments involve probabilistic measurements affected by quantum noise. Therefore, results obtained from the simulator may not fully reflect the challenges posed by measurement bias in practical quantum computing scenarios.

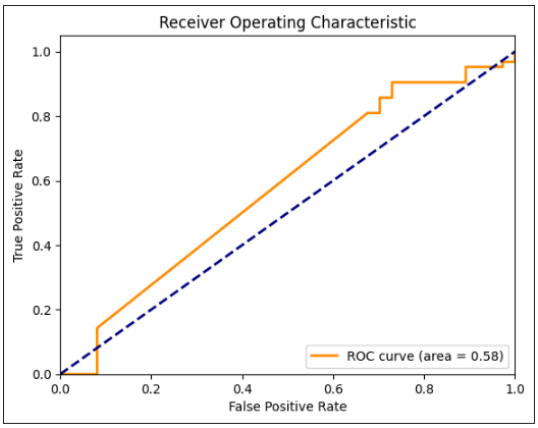

Figure 14: This *ROC* curve, shows an interesting pattern, it contains important clues, there are several possible factors that contribute to the almost diagonal shape and reveal signs of bias within of the experiment.

The result employs a hybrid approach, combining the quantum kernel with a classical SVM classifier, allowing for quantum feature utilization and classical classification techniques. Accuracy of approximately 0.63 suggests suboptimal performance on the Breast Cancer dataset. The confusion matrix reveals a relatively high number of false positives (25), indicating a propensity for incorrect positive predictions.

Several factors may contribute to this limited accuracy, including quantum noise, a small dataset size (400 training samples, 100 test samples), the choice of quantum hardware, among others; further optimization of hyper parameters, such as quantum shots, PCA components, SVM's hyper parameter, and seek another perspectives, is needed for improvement.

6. Discussion

The results presented in this study raise intriguing questions about the application of quantum machine learning in binary classification. The exploration of different Pauli configurations, the transition from quantum simulators to real-world quantum hardware, and the detection of bias in the models all provide valuable insights into the challenges and opportunities in this emerging field.

a. Unconventional Pauli Configurations

Using unconventional Pauli configurations, such as
['*Z', 'YY', 'ZXZ*'], challenges traditional quantum feature mapping approaches. While this configuration achieved a remarkably high recall probably biased, it came at the cost of accuracy and F1-score.

This raises a fundamental question:
Can unconventional Pauli configurations open new avenues for capturing complex data correlations that classical methods might overlook?

Quantum computing is a realm of boundless possibilities, and exploring non-standard Pauli operators hints at the untapped potential for uncovering hidden relationships within data. However, it also appears to have a propensity for false positives. This intriguing trade-off warrants further investigation into how these configurations may be harnessed in unique ways to address specific challenges in healthcare and beyond.

b. Quantum Simulators: The Quantum Sandbox

The reliance on quantum simulators as a preliminary testing ground showcases their indispensable role in quantum research. These simulators serve as a quantum sandbox, allowing to experiment with quantum algorithms in a controlled and resource efficient environment. The high-fidelity results attained in simulators provide a tantalizing glimpse into the quantum realm's potential.

Simulators enable quantum algorithm development and debugging without the complexities and uncertainties of real hardware. They set a benchmark for what is theoretically achievable, laying the foundation for ambitious quantum experiments. As we push the boundaries of quantum computing, the role of simulators as a bridge between theory and practice cannot be overstated.

c. Quantum Hardware: Navigating the Quantum Frontier

Transitioning from simulators to real-world quantum hardware is akin to navigating a quantum frontier. The drop in accuracy from the simulator (0.94) to the quantum processor (0.63) underscores the harsh realities of working with real quantum systems. Quantum computations are inherently probabilistic, and quantum noise introduces variability that can impact results. This quantum noise is not a limitation but a characteristic of quantum systems. Understanding and harnessing this noise for specific tasks will be a pivotal challenge.

d. Addressing Bias and the Quantum Edge

Bias, a persistent challenge in machine learning, also finds its way into the quantum realm. The bias detected in certain experiments suggests that careful consideration is needed in selecting quantum configurations and mitigating unwanted correlations. While bias is

often seen as a problem, it can also be a tool for fine-tuning quantum algorithms.

Quantum computing's edge lies in its ability to explore unconventional paths and challenge classical paradigms. This includes not only the algorithms but also the way we perceive and address bias. Embracing the quantum advantage may require us to rethink how we interpret and utilize bias in quantum machine learning. As we venture further into the quantum frontier, it is a journey that promises not just solutions to complex problems but also a deeper understanding of the quantum universe itself.

7. Conclusions

This study provides valuable insights into the application of quantum computing algorithms associated with machine learning for breast cancer classification. It highlights the importance of selecting appropriate Pauli configurations, leveraging quantum simulators for development and benchmarking, and addressing bias in quantum models. While the high recall achieved with certain Pauli settings may be biased results, this indicates the potential of quantum computing to identify true cancer cases, there is room for improvement in terms of accuracy and reducing false positives. Further research and optimization efforts are needed to harness the power of quantum computing for healthcare applications effectively. The study also underscores the significance of quantum simulators as a crucial component of the quantum research pipeline, enabling researchers to iteratively develop and validate quantum algorithms before running experiments on real quantum hardware. Ultimately, the combination of quantum computing and classical machine learning techniques holds promise for addressing complex healthcare challenges. Nevertheless, it requires a thorough understanding of quantum algorithms, careful experimentation, and continuous optimization to realize its full potential in real-world applications.

References

1. Zwitter M, Soklic M. (1988). Breast Cancer UCI Machine Learning Repository.
2. Yala, A., Lehman, C., Schuster, T., Portnoi, T., & Barzilay, R. (2019). A deep learning mammography-based model for improved breast cancer risk prediction. Radiology, 292(1), 60-66.
3. Matsuura, A. Y., & Mattson, T. G. (2022). Introducing the Quantum Research Kernels: Lessons from Classical Parallel Computing.
4. Bishop, C. M., & Nasrabadi, N. M. (2006). Pattern recognition and machine learning (Vol. 4, No. 4, p. 738). New York: springer.
5. Hastie, T., Tibshirani, R., Friedman, J. H., & Friedman, J. H. (2009). The elements of statistical learning: data mining, inference, and prediction (Vol. 2, pp. 1-758). New York: springer.
6. Landauer, R. (1991). Information is physical. Physics Today, 44(5), 23-29.
7. Anshu, A., & Arunachalam, S. (2023). A survey on the complexity of learning quantum states.
8. Montanaro, A. (2016). Quantum algorithms: an overview. npj Quantum Information, 2(1), 1-8.
9. Adesso, G., Bromley, T. R., & Cianciaruso, M. (2016). Measures and applications of quantum correlations. Journal of Physics A: Mathematical and Theoretical, 49(47), 473001.
10. Gelß, P., Klus, S., Schuster, I., & Schütte, C. (2021). Feature space approximation for kernel-based supervised learning. Knowledge-Based Systems, 221, 106935.
11. Häner, T., Kliuchnikov, V., Roetteler, M., Soeken, M., & Vaschillo, A. (2022). QParallel: Explicit Parallelism for Programming Quantum Computers.
12. Gönen, M., & Alpaydın, E. (2011). Multiple kernel learning algorithms. The Journal of Machine Learning Research, 12, 2211-2268.
13. Mancilla, J., & Pere, C. (2022). A Preprocessing Perspective for Quantum Machine Learning Classification Advantage in Finance Using NISQ Algorithms. Entropy, 24(11), 1656.
14. Li, G., Ye, R., Zhao, X., & Wang, X. (2022). Concentration of data encoding in parameterized quantum circuits. Advances in Neural Information Processing Systems, 35, 19456-19469.
15. Chen, B. S., & Chern, J. L. (2022). Generating quantum feature maps for SVM classifier.
16. Caro, M. C., Gil-Fuster, E., Meyer, J. J., Eisert, J., & Sweke, R. (2021). Encoding-dependent generalization bounds for parametrized quantum circuits. Quantum, 5, 582.
17. 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.
18. Shor, P. W. (1995). Scheme for reducing decoherence in quantum computer memory. Physical review A, 52(4), R2493.
19. Georgescu IM, Ashhab S, Nori F. (2014). Quantum simulation. Reviews of Modern Physics. 2014 mar;86(1):153-85.
20. Maurya, S., Mude, C. N., Oliver, W. D., Lienhard, B., & Tannu, S. (2023, June). Scaling Qubit Readout with Hardware Efficient Machine Learning Architectures. In Proceedings of the 50th Annual International Symposium on Computer Architecture (pp. 1-13).
21. Schuld, M., & Petruccione, F. (2018). Supervised learning with quantum computers (Vol. 17). Berlin: Springer.
22. Kübler, J., Buchholz, S., & Schölkopf, B. (2021). The inductive bias of quantum kernels. Advances in Neural Information Processing Systems, 34, 12661-12673.