Kernel methods are the basis of most classical machine learning algorithms such as Gaussian Process (GP) and Support Vector Machine (SVM). Computing kernels using noisy intermediate scale quantum (NISQ) devices has attracted considerable attention due to recent progress in the design of NISQ devices. However noise and errors on current NISQ devices can negatively affect the predicted kernels. In this paper we utilize two quantum kernel machine learning (ML) algorithms to predict the labels of a Breast Cancer dataset on two different NISQ devices: quantum kernel Gaussian Process (qkGP) and quantum kernel Support Vector Machine (qkSVM). We estimate the quantum kernels on the 11 qubit IonQ and the 5 qubit IBMQ Belem quantum devices. Our results demonstrate that the predictive performances of the error mitigated quantum kernel machine learning algorithms improve significantly compared to their non-error mitigated counterparts. On both NISQ devices the predictive performances became comparable to those of noiseless quantum simulators and their classical counterparts.

1 Introduction

In recent years, there has been significant progress in quantum hardware development. This has led to interest in the design and simplification of quantum algorithms for implementation on existing noisy intermediate scale quantum (NISQ) devices [1] [2–4]. Quantum Machine Learning (QML) is an emerging discipline combining data science with quantum computing, and it is one of the key applications of NISQ devices [5] for various reasons. First, QML algorithms do not need highly accurate gate operations and computations such as for those used in other potential applications of quantum computing such as factoring and search problems. Second, the execution of quantum circuits for estimation of kernels can be done using shallow quantum circuits which can be implemented using current NISQ devices. Because of this NISQ devices are able to execute quantum machine learning algorithms for small datasets [6].

The execution of QML algorithms on today’s NISQ devices is done in the form of quantum gate operations with two main steps. First, classical data must be encoded into quantum states using quantum gate operations. For this purpose various data encoding strategies have been proposed [3, 4, 7–11]. Second, since the output results of the quantum gate operations are always quantum states, a quantum measurement must be used to extract the classical bits of information from the quantum states. However the output results of QML algorithms on NISQ devices are inevitably deviated owing to noise originating from qubit initialization, errors from quantum gate operations (gate errors) and quantum measurements (read-out errors). Furthermore the execution times on NISQ devices are limited due to T1 relaxation and T2 dephasing times respectively. Therefore error mitigation and correction techniques at the algorithmic level are vital. For example quantum error correction [12, 13] is considered as a technique to reduce the effects of noise and
errors. The main challenge of quantum error correction is that they require a large number of physical qubits and a lower gate error rate which to date are not yet implementable on NISQ devices [5]. Another technique is quantum error mitigation which aims to reduce the effects of noise in NISQ devices [14–17].

Previously proposed techniques for error mitigation consider a combination of different methods to correct the sources of errors separately. For example correction of readout errors have been performed using unfolding methods [18, 19], while randomize compiling [20, 21] and zero noise extrapolation [22] are used for the correction of gate errors. In contrast in this paper we utilize two error mitigation techniques to correct all sources of errors simultaneously through estimation of the probabilistic error rate which depends on the NISQ device as well as the width and the depth of the quantum circuit.

2 Method

2.1 Data

In order to train the qkGP and qkSVM machine learning methods we used the Wisconsin Breast Cancer Dataset. This contains 32 features and 569 samples that have been evaluated in various contexts [23]. Corresponding information about the original paper and usages of Breast Cancer dataset can be found in the UCI Machine Learning Repository.

2.2 Quantum kernel Gaussian Process

Consider a binary classification problem with a training dataset which we call D such that

\[ D = (\tilde{X}, y) = \{\tilde{x}_i, y_i\}_{i=0\ldots M-1}, \]

where \( \tilde{x}_i \) is an N-dimensional vector and labels \( y_i \in \{0, 1\} \). The task is to model a function which can generate the labels within a dataset from given input vectors such that

\[ y = f(\tilde{X}) + \epsilon_{\text{noise}}, \]

where \( \epsilon_{\text{noise}} \sim \eta(0, \sigma^2) \) represents a distributed Gaussian noise with zero mean and variance \( \sigma^2 \). The function of the Gaussian process algorithm is to predict an unknown label \( y_* \) for a given test data vector \( \tilde{x}_* \). A scalar Gaussian Process (GP) is defined as the multivariate normal distribution \( GP(m(\cdot), k(\cdot, \cdot)) \), where \( m(\cdot) \) is a mean function and \( k(\cdot, \cdot) \) is a covariance function. The joint distribution of \( y \) as the label of the train data and \( y_* \) as the unknown label of \( N \)-dimensional vector test \( \tilde{x}_* \) is defined

\[ y_*(\tilde{X}, \tilde{x}_*, \epsilon_{\text{noise}}, y) \sim GP(m, k), \]

where \( m = K^T_x (K + \Sigma)^{-1} y, \quad k = K_{xx} + \Sigma - K^T_x (K + \Sigma)^{-1} K_x \), and \( \Sigma = \sigma^2 I \) is a \( M \times M \) diagonal matrix. \( K \) is the train kernel matrix, \( K_x \) is the train-test kernel matrix, \( K^T \) is the test-train kernel matrix, and \( K_{xx} \) is the test kernel matrix. In the binary classification, the joint probability for \( y_* \) to be 1 at \( \tilde{x}_* \) given train dataset \( \tilde{X} \) and label \( y \) can be approximated by [24]

\[ p(y_* = 1 | y) \approx S(\kappa m), \]

where \( S(x) = \frac{1}{1 + e^{-x}} \), a sigmoid function, and \( \kappa = \frac{1}{\sqrt{1 + \frac{\sigma^2}{\epsilon^2}}} \). Tresholding the value \( S(\kappa m) \) yields the binary output as following

\[ y_* = \begin{cases} 1 & \text{if } S(\kappa m) \geq 0.5 \\ 0 & \text{else} \end{cases} \]

For the qkGP algorithm, the NISQ device is used three times to estimate \( K, K_x, \) and \( K_{xx} \) (see Figure 10 in Appendix A).

2.3 Quantum kernel Support Vector Machine

For the quantum kernel SVM (qkSVM) [25], the standard form of the kernelized binary classifier is

\[ y_* = \text{sgn}(\sum_{i=1}^{M} y_i \alpha^*_i K(\tilde{x}_i, \tilde{x}_*)), \]

where \( y_* \) is the unknown label of the test dataset \( \tilde{x}_* \), \( y_i \) is the label of the \( i^{th} \) train sample, \( \alpha^*_i \) is the \( i^{th} \) component of the support vector \( \alpha^* = (\alpha^*_1, \alpha^*_2, ..., \alpha^*_M) \), \( M \) is the number of train data, and \( K(\tilde{x}_i, \tilde{x}_*) \) is the kernel of the train-test pairs. For a given dataset

\[ D = (\tilde{X}, y) = \{\tilde{x}_i, y_i\}_{i=0\ldots M-1}, \]

where \( \tilde{x}_i \) is an \( N \)-dimensional vector and labels \( y_i \in \{-1, 1\} \), one option to calculate the support vector \( \alpha^* \) without optimization of complex function as presented in [25] is to set uniform weight \( \alpha^*_i = 1 \), in case of balanced dataset, \( IR = 0.5 \).
where $IR = \frac{x}{y}$ with $x$ being the number of minority class and $y$ being the total number of samples. Otherwise, $\alpha^*_y = IR$ for the majority class and $\alpha^*_y = 1 - IR$ for the minority class applies. Thresholding the value $y_i\alpha^*_y K(x_i, x^*)$ yields the binary output

$$y_i = \begin{cases} 1 & \sum_{i=1}^M y_i\alpha^*_y K(x_i, x^*) \geq 0 \\ -1 & \text{else} \end{cases}.$$ (8)

The NISQ device is used only once to estimate the kernel matrix of train-test pairs (see Figure 11 in Appendix A).

### 2.4 Data encoding strategy

Given a classical normalized data vector $\vec{a} = (a_1, a_2, a_3, a_4)$, quantum circuit of Figure 1 is able to encode four dimensional vector $\vec{a}$ into amplitudes of a quantum state $|\psi\rangle = \sum_{i=0}^3 a_i|\alpha\rangle$ using only one $CNOT$ gate [26], with

$$\theta_0 = \text{Arg}(a_1 + ia_2) - \text{Arg}(a_3 + ia_4)$$ (9)

$$\theta_1 = 2\cos^{-1}(\sqrt{a_1^2 + a_2^2})$$ (10)

$$\theta_2 = \text{Arg}(a_1 + ia_2) + \text{Arg}(a_3 + ia_4).$$ (11)

![Figure 1: Quantum circuit for mapping a four-dimensional state vectors to two qubit states by applying three single $R_y$ rotation gates with rotation angles of $\theta$ (Eqs. 9, 10, and 11), and two Hadamard gates with a $CNOT$ gate in between.](image)

The application of the single controlled swap gates on the state given in Eq. 12 generates an entangled state $\frac{1}{\sqrt{2}}(|0\rangle_a|u\rangle + |1\rangle_a|v\rangle|u\rangle)$. Then, another Hadamard gate is used to generate the product state of the state vectors for train and test such that

$$\frac{1}{2}(|0\rangle_a(|u\rangle|v\rangle + |v\rangle|u\rangle) + |1\rangle_a((|u\rangle|v\rangle - |v\rangle|u\rangle)).$$ (13)

The quantum state given in Eq. 13 is measured in the computational basis of the $|0\rangle_a$ state to yield the probability

$$Pr(|0\rangle_a) = \frac{1 + |\langle u|v\rangle|^2}{2},$$ (14)

where $Pr(|0\rangle_a)$ is the probability of measurement on the $|0\rangle_a$ state of Eq. 13.

![Figure 2: Quantum circuit to compute kernels. The models of quantum circuits $U$ and $V$ encode data into amplitudes of quantum states $|u\rangle = U|00\rangle_{12}$ and $|v\rangle = V|00\rangle_{34}$. The quantum state of Eq. 13 is measured on the $|0\rangle$ basis to estimate the value of $|\langle u|v\rangle|^2$ from Eq. 14.](image)

### 2.5 Estimation of the kernels with the Swap Test

Figure 2 shows the quantum circuit for estimating $|\langle u|v\rangle|^2$ with a Swap test. First the quantum circuits (see section 2.4) encode train and test data into quantum states $|u\rangle = U|00\rangle_{12}$ and $|v\rangle = V|00\rangle_{34}$, respectively. The Hadamard gate is applied on to the ancillary qubit $|0\rangle_a$ to create a superposition of $|u\rangle|v\rangle$, i.e.

$$\frac{1}{\sqrt{2}}(|0\rangle_a|u\rangle|v\rangle + |1\rangle_a|u\rangle|v\rangle))$$ (12)

where $Pr(|0\rangle_a)$ is the probability of measurement on the $|0\rangle_a$ state of Eq. 13.

### 2.6 Noise model of NISQ devices and error mitigation

Noise and errors are significant issues for NISQ devices. Three important types of noise and errors in NISQ devices are qubit initialization, gate errors and readout errors. Qubits will lose their quantum properties due to interaction with their environment resulting in a mixed initial quantum state. The gate errors mostly result from miscalibration or imperfection in the control hardware and their interaction with the qubits. The readout errors concern the measuring of incorrect qubit values e.g. reading zero while the qubit is in the one state and vice versa. While there are
many distinct noise models [27], in this study, we consider the depolarizing noise as a model to explain the noise and errors in NISQ devices. For an n-qubits pure quantum state |φ⟩, the depolarizing noise channel is given by

$$\epsilon_\lambda(\rho) = (1 - \lambda)\rho + \lambda \frac{I}{2^n},$$

(15)

where $\epsilon_\lambda$ denotes the noise channel operator, $\rho = |\phi\rangle\langle\phi|$ is the density matrix, $\lambda$ is the probabilistic error rate, and $I$ is the $(2^n \times 2^n)$ identity matrix [28]. Therefore the expectation value of observable $O$ for a state represented by a density matrix $\rho$ is

$$\langle O \rangle = \text{tr}[\epsilon_\lambda(\rho)O] = (1 - \lambda)\langle O \rangle + \lambda \frac{\text{tr}(O)}{2^n},$$

(16)

where $\langle O \rangle$ is the noisy expectation value and $\langle O \rangle$ is the noiseless expectation value. Using Eq. 15, the noisy kernel matrix from the noiseless kernel matrix is computed [28]

$$\tilde{K}(x_i, x_j) = (1 - \lambda_i\lambda_j)K(x_i, x_j) + \frac{\lambda_i\lambda_j}{2^n}.$$  

(17)

Since all diagonal entries of the noiseless train kernel matrix and test kernel matrix must be equal to 1, therefore the probabilistic error rate $\lambda_1$ is obtained from the $i^{th}$ entry of kernel matrix $\tilde{K}(x_i, x_i)$ estimated by NISQ device, i.e.

$$\lambda_i = \sqrt{\frac{1 - K(x_i, x_i)}{1 - \frac{1}{2^n}}}.$$  

(18)

The error mitigated entry $K(x_i, x_j)$ is calculated from the noisy entry $\tilde{K}(x_i, x_j)$, i.e.

$$K(x_i, x_j) = \frac{\tilde{K}(x_i, x_j) - \frac{1}{2^n}\lambda_i\lambda_j}{1 - \lambda_i\lambda_j}.$$  

(19)

2.7 Estimation of the probabilistic error rate ($\lambda$) for error mitigation

The predictive performance of qkGP and qkSVM are influenced by noise and errors in NISQ devices. We corrected the train kernel matrix $K$, and the test kernel matrix $K_\sigma$, estimated by a NISQ device for qkGP (see Appendix A), first, by estimating the probabilistic error rate $\lambda$ from Eq. 18 and then using Eq. 19 (see Appendix D for error mitigation of a test kernel matrix). Alternatively, the train-test kernel matrix $K_\ast$ is corrected for qkGP and qkSVM, first, by computing the values of $\lambda$ with a noise-estimation circuit and then using Eq. 16.

To derive the noise estimation circuit, first, the quantum circuit for the Swap Test is transpiled into single-qubit and two-qubit gates and then all single-qubit gates are removed from the transpiled circuit. The main assumption of this approach is that CNOT gates are the dominant source of gate errors for NISQ devices. Since the Swap Test quantum circuit and the noise-estimation circuit have the same structure, they are affected by nearly similar $\lambda$s.

Figure 3 shows the noise-estimation circuit which can be implemented directly on the IonQ quantum machine according to coupling map (Figure 10 in Appendix B). The noise-estimation circuit was executed 324 (in this study size the train-test matrix is 54×6) times with 500 and 8192 measurement shots for each execution on the IonQ and the IBMQ Belem quantum machines, respectively, and then the expectation value of $\sigma_z$ was measured. From Eq. 16, the expectation value of $\sigma_z$ is 1 for noiseless quantum simulator and $1 - \lambda$ for the NISQ device, because the act of ideal (noiseless) CNOT gates on initial state of a quantum system (see Figure 3) does not transform it at all. Therefore the corrected train-test kernel matrix $K_\ast$ is obtained from the noisy $\bar{K}_\ast$ using Eq. 16, i.e.

$$K_\ast = \frac{\bar{K}_\ast}{1 - \lambda}.$$  

(20)

2.8 Software and Hardware

For classical machine learning algorithms we use scikit-learn [29]. Qiskit was used for simulating and experimenting with quantum circuits. For the estimation of the kernel matrices with the Swap Test on NISQ device, we choose the 11-qubit IonQ quantum machine. The 5-qubit IBMQ Belem quantum machine is also used for comparison. Both systems are NISQ (See Figures 10 and 11 in Appendix B), but with different connectivity of qubits, number of qubits, and different technologies.

The gate-based IonQ and IBMQ Belem quantum machines were available for this study as cloud computing services, hosted by Microsoft Azure [30] and IBM quantum [31], respectively.
The elementary gates used on the IonQ quantum machine are single qubit gates and maximally entangling two qubit Mølmer–Sørensen gate (see Figure 16 in Appendix B for decomposition of $CNOT$ to Mølmer–Sørensen($R_{xx}$)).

For the IonQ quantum machine, physical qubits are implemented as rare earth Ytterbium-171 ions ($^{171}\text{Yb}^+$) trapped by electric fields and manipulated with a mode-locked 355nm laser which drives gate operations. The initial state of each $^{171}\text{Yb}^+$ qubit has a coherence time of $T_1 \approx 10^7 \mu s$, $T_2 > 200,000 \mu s$.

The estimation of the kernel matrices with the Swap Test was done by measuring the state of an ancilla qubit. To do this with the IonQ quantum machine two species of ions are chosen: one for measuring the ancilla qubit and one for the other qubits that are not measured [32]. Reading the ion as an ancilla qubit is done by shining a resonant laser with a wavelength of 369.5nm such that the photon emitted by the ancilla qubit will not excite the other qubits [32].

For the IBMQ Belem quantum machine physical qubits are implemented as Josephson junctions with a coherence time of $T_1 \approx 80 \mu s$, $T_2 \approx 110 \mu s$. These qubits are connected together to form a 2D lattice topology. Because these qubits utilize superconducting circuitry the device has to be cooled to very low temperatures [33]. The elementary gates used on the IBMQ Belem quantum machine are $I, R_z, S_X, X$ as well as two-qubit $CNOT$ gates. Quantum gates on the device have a gate time operation of few hundred nanoseconds.

### 3 Results

#### 3.1 Mitigation experiments

To show the effect of noise and errors on the values of the train-test kernel matrix ($K_*$), we plotted a scatter diagram on the noiseless simulator as well as on the IonQ quantum machine and then fit optimal lines using least square regression as shown in Figure 4. The slope of fit line in Figure 4 shows the correlation between values of the kernel matrix entries obtained from the simulator and the NISQ device before error mitigation. Based on the slope of fit line, fidelity of execution of the Swap Test on the 11-qubits IonQ machine was estimated 0.89 (89%) for the Breast Cancer dataset with four features.

![Figure 4: Estimated values of the kernel matrix train-test from the IonQ quantum machine before mitigation vs. Simulator.](image)

After correcting the value of the train-test kernel matrix from the NISQ device with Eq. 20, a high correlation between values of the kernel matrix from the simulator and NISQ device can be
Estimated values of the kernel matrix train-test from the 5-qubits IBMQ Belem without error mitigation vs simulator with 8192 measurement shots and the same dataset is also reported on Figure 6. As can be seen from this plot, there is lower correlation between values of the kernel matrix resulted from the simulator and the IBMQ Belem quantum machine compared to the Figure 4. The main reason, besides much longer coherent time of trapped ion qubits and lower gate and readout error rates with respect to superconducting qubits, is the connectivity between qubits which enable two-qubit gate operations. For the IonQ quantum machine, the qubits are fully connected which enables the mapping of two-qubit gates to physical qubits directly [34], while, for the IBMQ Belem, some two-qubit gate operations must be decomposed into executable two-qubit gates (see Appendix C for additional information). These decomposition steps increase number of noisy two-qubit gate operations and the depth of quantum circuit drastically.

The quality of Error Mitigation (qEM) was also calculated as a success rate for error mitigation of the kernel matrix. The qEM is division of the mitigated error, which is the difference between ideal values of kernel and corrected values of kernel, respect to the unmitigated error, which is the difference between ideal values of kernel and noisy values of kernel. From Figures 4, and 5, qEM was achieved values between 5 and 11, while qEM was between 4 and 10 based on Figures 6, and 7.

3.2 Predictive performance evaluation

Our experimental demonstrations are, first, performed on the 11-qubit IonQ quantum machine to obtain the predictive performance of quantum kernel ML algorithms. The experiments have been conducted on the open access Breast Cancer dataset with 54 train samples, 6 test samples, and four features. In Table 1 we demonstrate, the label of test dataset and the labels assigned to them by the qkGP and classical RBF GP algorithms for the case of two classes 0 and 1 in the 4-dimensional Breast Cancer dataset. It can be seen that the qkGP with error mitigation provides the correct labels for 6 out of 6 for 4-dimensional points, or an accuracy of 100% while with the same qkGP algorithm without er-
ror mitigation 4 correct labels out of 6 for four features are assigned with an accuracy of 66.67%. It can also be seen from Table 1 that the qkGP using quantum simulator and the GP with classical RBF kernel can predict both the 6 corrected value labels of 6.

| Labels | Predicted Labels (qkGP using Simulator) | Predicted Labels (Non-Error Mitigated) | Predicted Labels (Error Mitigated) | Predicted Labels using GP algorithm |
|--------|----------------------------------------|----------------------------------------|------------------------------------|------------------------------------|
| 0 0 1 1 1 0 | 0 0 1 1 1 0 | 0 1 1 1 0 0 | 0 0 1 1 1 0 | 0 0 1 1 1 0 |

Table 1: Comparison of labels assigned by the qkGP algorithm on the IonQ quantum machine. The first line shows the labels for the test dataset. The second line shows the prediction results of qkGP with simulator. The third and fourth lines show the labels assigned by the quantum kernel GP algorithm without and with error mitigation, respectively. The fifth line shows the labels assigned by GP algorithm with the classical RBF kernel. Values highlighted in red denote incorrectly predicted labels.

In Table 2, we demonstrate the label of test dataset and the label assigned to them by the qkSVM algorithm and classical RBF SVM for two classes 0 and 1 in the 4-dimensional Breast Cancer dataset. It can be seen that the qkSVM with simulator and with error mitigation provides the correct labels for 6 out of 6 for 4-dimensional data points, or an accuracy 100%, while with the same qkSVM algorithm without error mitigation 5 corrected labels out of 6 for 4-dimensional points are assigned with an accuracy of 83.33%.

| Labels | Predicted Labels (qkGP using Simulator) | Predicted Labels (Non-Error Mitigated) | Predicted Labels (Error Mitigated) |
|--------|----------------------------------------|----------------------------------------|------------------------------------|
| 0 0 1 1 1 0 | 0 0 1 1 1 0 | 0 1 0 1 1 1 | 0 1 1 1 1 0 |

Table 2: Comparison of labels assigned by qkSVM algorithm on the IonQ quantum machine. The first line shows the labels of test dataset. The second line shows the prediction results of qkSVM with simulator. The third and fourth lines show the labels assigned by the qkSVM algorithm without and with error mitigation, respectively. The fifth line shows the labels assigned by SVM algorithm with the classical RBF kernel. Values highlighted in red denote incorrectly predicted labels.

We estimated the quantum kernel on the IBMQ Belem quantum machine and, then, matched the predictive performance of the qkGP and the qkSVM algorithms for the same dataset without and with error mitigation techniques. As Tables 3 and 4 show, for four features dataset, we achieved 83.33% accuracy and 100% accuracy with error mitigation and 8192 shots for qkGP and qkSVM, respectively.

| Labels | Predicted Labels (Non-Error Mitigated) | Predicted Labels (Error Mitigated) |
|--------|----------------------------------------|------------------------------------|
| 0 0 1 1 1 0 | 1 0 1 1 0 1 | 0 1 1 1 1 0 |

Table 3: Comparison of labels assigned by qkGP algorithm on the IBMQ Belem quantum machine. The first line shows the labels for the test dataset. The second and fourth lines show the labels assigned by the qkGP algorithm without and with error mitigation, respectively.

| Labels | Predicted Labels (Non-Error Mitigated) | Predicted Labels (Error Mitigated) |
|--------|----------------------------------------|------------------------------------|
| 0 0 1 1 1 0 | 1 0 1 1 0 1 | 0 1 1 1 1 0 |

Table 4: Comparison of labels assigned by qkSVM algorithm on the IBMQ Belem quantum machine. The first line shows the labels for the test dataset. The second and fourth lines show the labels assigned by the qkSVM algorithm without and with error mitigation, respectively.

4 Conclusion

In this study, we aimed to investigate the effects of noise and errors on the predictive performance
of two quantum kernel based machine learning algorithms executed on the 11-qubit IonQ and the 5-qubit IBMQ Belem quantum machines. Then we reduced the effects of errors and noise with error mitigation techniques.

Our results demonstrate that error mitigation techniques for kernel matrices improves the prediction performance of QML algorithms running on the IonQ and the IBMQ Belem quantum machines. However the error rates of the IBMQ Belem quantum machine are high compared to the IonQ quantum machine but the error mitigation techniques under the depolarizing noise model can correct values of the kernel matrices successfully. According to this we concluded that the depolarizing noise model is optimal for NISQ devices. This is in agreement with the findings in [35].

In the context of this study, we found that the values of kernel matrix resulted from the IonQ quantum machine in combination with error mitigation technique are closer to those expected from an ideal circuit [4] as Figure 5 represents. In general, the advantage of IonQ on IBMQ is particularly due to the long time coherence, low gate error rate, low readout error rate, and coupling map of qubits. [36].

Tables 1-4 show that the data classification resulted from the the IonQ and the IBMQ Belem quantum machines without error mitigation can shift the label assignment non-trivially due to noise and errors. In conclusion data classification with two quantum kernel based algorithms e.g. qkGP and qkSVM can be comparable to these algorithms with classical RBF kernel if one applies error mitigation techniques on the measurement results of quantum circuits from NISQ devices.

Our experiments pointed out that the error mitigation techniques are vital for QML algorithms. We consider our findings of essential in relation to building quantum ML algorithms in future experiments with much higher fidelity gates of NISQ devices.

References

[1] Sergey Bravyi, David Gosset, Robert König, and Marco Tomamichel. “Quantum advantage with noisy shallow circuits”. In: Nature Physics 16.10 (2020), pp. 1040-1045. DOI: 10.1038/s41567-020-0948-z.

[2] Maria Schuld and Nathan Killoran. “Quantum Machine Learning in Feature Hilbert Spaces”. In: Phys. Rev. Lett. 122 (2019), p. 040504. DOI: 10.1103/PhysRevLett.122.040504.

[3] Vojtěch Havlíček et al. “Supervised learning with quantum-enhanced feature spaces”. In: Nature 567 (2019), pp. 209–212. DOI: 10.1038/s41586-019-0980-2.

[4] Sonika Johri et al. “Nearest centroid classification on a trapped ion quantum computer”. In: npj Quantum Information 7 (2021), p. 122. DOI: 10.1038/s41534-021-00456-5.

[5] John Preskill. “Quantum Computing in the NISQ era and beyond”. In: Quantum 2 (2018), p. 79. DOI: 10.22331/q-2018-08-06-79. URL: https://doi.org/10.22331/q-2018-08-06-79.

[6] Martin J. Willemink et al. “Preparing Medical Imaging Data for Machine Learning.” In: Radiology (2020), p. 192224. DOI: 10.1148/radiol.2020192224.

[7] Maria Schuld and Francesco Petruccione. Supervised Learning with Quantum Computers. 1st. Springer Publishing Company, Incorporated, 2018.

[8] Mikko Möttönen, Juha J. Vartiainen, Ville Bergholm, and Martti M. Salomaa. “Quantum Circuits for General Multi-qubit Gates”. In: Phys. Rev. Lett. 93 (2004), p. 130502. DOI: 10.1103/PhysRevLett.93.130502.

[9] Hsin-Yuan Huang et al. “Power of data in quantum machine learning”. In: Nature Communications 12 (2021), p. 2631. DOI: 10.1038/s41467-021-22539-9.

[10] Maria Schuld, Alex Bocharov, Krysta M. Svore, and Nathan Wiebe. “Circuit-centric quantum classifiers”. In: Phys. Rev. A 101 (3 2020), p. 032308. DOI: 10.1103/PhysRevA.101.032308.

[11] M. Schuld, M. Fingerhuth, and F. Petruccione. “Implementing a distance-based classifier with a quantum interference circuit”. In: EPL (Europhysics Letters) 119 (2017), p. 60002. DOI: 10.1209/0295-5075/119/60002.
[12] Peter W. Shor. “Scheme for reducing decoherence in quantum computer memory”. In: *Phys. Rev. A* 52 (1995), R2493–R2496. DOI: 10.1103/PhysRevA.52.R2493.

[13] A. M. Steane. “Error Correcting Codes in Quantum Theory”. In: *Phys. Rev. Lett.* 77 (5 1996), pp. 793–797. DOI: 10.1103/PhysRevLett.77.793.

[14] Ying Li and Simon C. Benjamin. “Efficient Variational Quantum Simulator Incorporating Active Error Minimization”. In: *Phys. Rev. X* 7 (2017), p. 021050. DOI: 10.1103/PhysRevX.7.021050.

[15] Kristan Temme, Sergey Bravyi, and Jay M. Gambetta. “Error Mitigation for Short-Depth Quantum Circuits”. In: *Phys. Rev. Lett.* 119 (18 2017), p. 180509. DOI: 10.1103/PhysRevLett.119.180509.

[16] Suguru Endo, Simon C. Benjamin, and Ying Li. “Practical Quantum Error Mitigation for Near-Future Applications”. In: *Phys. Rev. X* 8 (2018), p. 031027. DOI: 10.1103/PhysRevX.8.031027.

[17] Abhinav Kandala et al. “Error mitigation extends the computational reach of a noisy quantum processor”. In: *Nature* 567 (2019), pp. 491–495. DOI: 10.1038/s41586-019-1040-7.

[18] Benjamin Nachman, Miroslav Urbanek, Wibe A. de Jong, and Christian W. Bauer. “Unfolding quantum computer readout noise”. In: *npj Quantum Information* 6 (2020), p. 84. DOI: 10.1038/s41534-020-00309-7.

[19] Miroslav Urbanek, Benjamin Nachman, and Wibe A. de Jong. “Error detection on quantum computers improving the accuracy of chemical calculations”. In: *Phys. Rev. A* 102 (2020), p. 022427. DOI: 10.1103/PhysRevA.102.022427.

[20] Joel J. Wallman and Joseph Emerson. “Noise tailoring for scalable quantum computation via randomized compiling”. In: *Phys. Rev. A* 94 (2016), p. 052325. DOI: 10.1103/PhysRevA.94.052325.

[21] Zhenyu Cai and Simon C. Benjamin. “Constructing Smaller Pauli Twirling Sets for Arbitrary Error Channels”. In: *Scientific Reports* 9 (2019), p. 11281. DOI: 10.1038/s41598-019-46722-7.

[22] Andre He, Benjamin Nachman, Wibe A. de Jong, and Christian W. Bauer. “Zero-noise extrapolation for quantum-gate error mitigation with identity insertions”. In: *Phys. Rev. A* 102 (2020), p. 012426. DOI: 10.1103/PhysRevA.102.012426.

[23] Dheeru Dua and Casey Graff. *UCI Machine Learning Repository*. 2017. URL: http://archive.ics.uci.edu/ml.

[24] Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*. Vol. 4. 4. Springer, 2006.

[25] Sasan Moradi et al. “Clinical data classification with noisy intermediate scale quantum computers”. In: *Scientific Reports* 12 (2022). DOI: https://doi.org/10.1038/s41598-022-05971-9.

[26] Oscar Perdomo. “Canonical representation of three-qubit states with real amplitudes”. In: *Journal of Physics A: Mathematical and Theoretical* 54.46 (2021), p. 465301. DOI: 10.1088/1751-8121/ac2e27.

[27] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2010. DOI: 10.1017/CBO9780511976667.

[28] Thomas Hubregtsen et al. *Training Quantum Embedding Kernels on Near-Term Quantum Computers*. 2021. DOI: 10.48550/ARXIV.2105.02276. URL: https://arxiv.org/abs/2105.02276.

[29] Fabian Pedregosa et al. “Scikit-learn: Machine Learning in Python”. In: *CoRR* abs/1201.0490 (2012). arXiv: 1201.0490. URL: http://arxiv.org/abs/1201.0490.

[30] Microsoft Quantum Team. *Microsoft Azure*. URL: https://microsoft.com/azure/quantum.

[31] IBM Quantum Team. *IBM Quantum Experience*. URL: https://quantum-computing.ibm.com.
Stephen Crain et al. “High-speed low-crosstalk detection of a 171Yb+ qubit using superconducting nanowire single photon detectors”. In: Communications Physics 2 (2019), pp. 1–6. DOI: https://doi.org/10.1038/s42005-019-0195-8.

Morten Kjaergaard et al. “Superconducting Qubits: Current State of Play”. In: Annual Review of Condensed Matter Physics 11.1 (2020), pp. 369–395. DOI: 10.1146/annurev-conmatphys-031119-050605.

Andrii Maksymov, Pradeep Niroula, and Yunseong Nam. “Optimal calibration of gates in trapped-ion quantum computers”. In: Quantum Science and Technology 6.3 (2021), p. 034009. DOI: 10.1088/2058-9565/abf718. URL: https://doi.org/10.1088/2058-9565/abf718.

Miroslav Urbanek et al. “Mitigating Depolarizing Noise on Quantum Computers with Noise-Estimation Circuits”. In: Phys. Rev. Lett. 127 (2021), p. 270502. DOI: 10.1103/PhysRevLett.127.270502.

Nicolas Schwaller, Valeria Vento, and Christophe Galland. “Experimental QND measurements of complementarity on two-qubit states with IonQ and IBM Q quantum computers”. In: Quantum Information Processing 21 (2022), pp. 1–24. DOI: https://doi.org/10.1007/s11128-021-03354-z.

Kenneth Wright et al. “Benchmarking an 11-qubit quantum computer”. In: Nature Communications 10 (2019). DOI: https://doi.org/10.1038/s41467-019-13534-2.

Robert Wille, Stefan Hillmich, and Lukas Burgholzer. “Efficient and Correct Compilation of Quantum Circuits”. In: 2020 IEEE International Symposium on Circuits and Systems (ISCAS). 2020, pp. 1–5. DOI: 10.1109/ISCAS45731.2020.9180791.
A Quantum kernel machine learning algorithms

Figure 8 shows a flowchart of the quantum kernel Gaussian Process (qkGP) algorithm. After the kernel matrix has been estimated the error mitigation technique can be implemented in order to reduce the effects of noise on predictive performance. As can be seen from Figure 8, NISQ device estimates the kernel matrices of the train, test, and train-test data.

Figure 8: Schematic of the Gaussian Process algorithm for binary classification. NISQ device is used three times to estimate $K(x_i, x_j)$, $K(x_i^*, x_j^*)$, and $K(x_i, x_j^*)$. After estimation of kernel matrix in each step of the flowchart, error mitigation technique proposed in Section 2.6 is used to degrade the effects of noise and errors.

Figure 9 represents a flowchart of the quantum kernel SVM (qkSVM) algorithm. NISQ device is used only once to estimate the kernel matrix of the train-test pairs.
Figure 9: Schematic of the qkSVM for binary data classification. First, the train data vector $\vec{x}_i$ and the test data vector $\vec{x}^*$ are prepared on a classical computer. Next, the train data and the test data data are encoded into quantum states $|u\rangle$ and $|v\rangle$ followed by computing the kernel matrix for all pairs of the train-test data $K(\vec{x}, \vec{x}^*)$ with a NISQ device. Then the error mitigation technique is utilized to reduce the effects of noise and errors on the NISQ device. If $\vec{\alpha}^* = (\alpha^*_1, \alpha^*_2, \ldots, \alpha^*_M)$ are considered to be a solution of the support vector, the binary classifier can be constructed based on Eq. 8.

B NISQ devices

Figures 10 shows the coupling maps of the 11-qubit IonQ quantum machine. Each circle represents a trapped ion qubit. The lines in-between the qubits represent physical connections for implementation of two qubit gate. The single-qubit and two-qubit gate error rates can also be seen for the IonQ quantum machine and also via https://IonQ.com/technology.

Figure 10: Topology graph and coupling map of the 11-qubits IonQ quantum machine [37].

Figures 11 shows the coupling maps of the 5-qubit IBMQ Belem quantum machine. Each circle
represents a physical superconducting transmon qubit. The lines in-between qubits represent physical connections via superconducting transmission lines. The gate error rates and readout errors can also be seen for the IBMQ Belem.

C Gate Decomposition

In order to execute a quantum circuit on different NISQ architectures, two-qubit gates must satisfy the coupling constraints of the architecture [38]. Since all NISQ devices only support single-qubit gates and two-qubit gate operations, complex gate operations must be decomposed into supported gates before mapping on noisy hardware. Owing to the specific architectures of different NISQ devices, all two-qubit gate operations must satisfy the constraints imposed by the coupling map of physical qubits [38], i.e., if $q_i$ is the control qubit and $q_j$ is the target qubit, $CNOT(q_i, q_j)$ can only be applied if there is coupling between $q_i$ and $q_j$. Otherwise $CNOT(q_i, q_j)$ must be decomposed into executable $CNOT$ gate operations. The compilation of quantum circuit for Swap Test on NISQ devices is computationally expensive due to existence three-qubits controlled swap gates (See Figure 2).

Figure 12 shows the efficient decomposition of the $C\text{SWAP}(q_0, q_1, q_3)$ (Figure 12) into single-qubit rotation $U_3$ and $CNOT$ gates. The largest $CNOT$s in Figure 12 are $CNOT(q_0, q_3)$ and $CNOT(q_0, q_4)$ that can be implemented directly on the IonQ machine due to coupling map (see Figure 10), while $CNOT(q_0, q_3)$ and and $CNOT(q_0, q_4)$ must be decomposed to shorter $CNOT$s before execution on the IBMQ Belem quantum machine. There are two approaches to map $CNOT(q_0, q_3)$ on the IBMQ belem [25]. The first option is to apply a SWAP gate between $q_0$ and $q_1$ and, then, a $CNOT$ between $q_1$ and $q_4$. To complete the effect, another SWAP gate is applied between $q_0$ and $q_1$. This approach is not efficient, since action of each SWAP gate is equal to three $CNOT$ gates. Another approach is to use sequence of $CNOT$ gates as shown in Figure 13.

The largest $CNOT$s in Figure 14 are $CNOT(q_0, q_4)$ that must be decomposed to shorter $CNOT$s before execution the quantum circuit on the IBMQ Belem quantum machine. For $CNOT(q_0, q_4)$, decomposition to shorter $CNOT$ is used twice. First, $CNOT(q_0, q_4)$ is decomposed into $CNOT(q_0, q_1)$ and $CNOT(q_1, q_3)$. Then, $CNOT(q_1, q_4)$ is decomposed to executable $CNOT(q_2, q_4)$. Figure 15 shows the decomposition step for $CNOT(q_0, q_4)$.
Figure 12: The decomposition of $CSWAP(q_0, q_1, q_3)$ to general single-qubit rotation gates and two-qubit $CNOT$ gates.

Figure 13: Decomposition of $CNOT(q_0, q_3)$ to shorter $CNOT$s.

Figure 14: The decomposition of $CSWAP(q_0, q_2, q_4)$ to general single-qubit rotation gates and two-qubit $CNOT$ gates.

Figure 15: Decomposition of $CNOT(q_0, q_4)$ to shorter $CNOT$s. $CNOT(q_1, q_4)$ must be decomposed to shorter $CNOT$s with the same procedure as shown in Figure 13.

Figure 16: Decomposition of $CNOT$ to two qubit Mølmer–Sørensen gate.

D Error mitigation of the test kernel matrix

To show an example about how mitigated entries can be obtained from noisy entries using Eq. 19, we, first, estimated the noisy kernel matrix for one fold of four features Breast cancer dataset on the IonQ
quantum machine as

\[
K = \begin{bmatrix}
0.861 & 0.778 & 0.757 & 0.649 & 0.82 & 0.766 \\
0.769 & 0.859 & 0.788 & 0.514 & 0.783 & 0.514 \\
0.766 & 0.809 & 0.85 & 0.739 & 0.843 & 0.577 \\
0.642 & 0.519 & 0.756 & 0.878 & 0.75 & 0.612 \\
0.842 & 0.793 & 0.832 & 0.751 & 0.854 & 0.695 \\
0.76 & 0.528 & 0.586 & 0.595 & 0.711 & 0.858
\end{bmatrix}
\]

and then the error mitigated kernel matrix

\[
K = \begin{bmatrix}
1 & 0.904 & 0.884 & 0.745 & 0.956 & 0.891 \\
0.893 & 1 & 0.922 & 0.589 & 0.914 & 0.597 \\
0.895 & 0.946 & 1 & 0.855 & 0.989 & 0.674 \\
0.737 & 0.596 & 0.875 & 1 & 0.866 & 0.703 \\
0.982 & 0.925 & 0.977 & 0.867 & 1 & 0.811 \\
0.884 & 0.612 & 0.684 & 0.684 & 0.83 & 1
\end{bmatrix}
\]

. The kernel matrix with quantum simulator was estimated

\[
K_{\text{sim}} = \begin{bmatrix}
1 & 0.898 & 0.895 & 0.765 & 0.963 & 0.903 \\
0.903 & 1 & 0.919 & 0.608 & 0.912 & 0.606 \\
0.899 & 0.927 & 1 & 0.855 & 0.979 & 0.679 \\
0.761 & 0.621 & 0.871 & 1 & 0.878 & 0.714 \\
0.963 & 0.909 & 0.978 & 0.888 & 1 & 0.802 \\
0.896 & 0.626 & 0.679 & 0.699 & 0.824 & 1
\end{bmatrix}
\]