Clinical Data Classification With Noisy Intermediate Scale Quantum Computers

Sasan Moradi

Medical University of Vienna

Christoph Brandner

Medical University of Vienna

Clemens Spielvogel

Medical University of Vienna

Denis Krajnc

Medical University of Vienna

Stefan Hillmich

Johannes Kepler University Linz

Robert Wille

Johannes Kepler University Linz

Wolfgang Drexler

Medical University of Vienna

Laszlo Papp (✉ laszlo.papp@meduniwien.ac.at)

Medical University of Vienna

Research Article

Keywords: Quantum computing, quantum machine learning, classic data encoding, clinical prediction models

Posted Date: October 5th, 2021

DOI: <https://doi.org/10.21203/rs.3.rs-942360/v1>

License: This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Version of Record: A version of this preprint was published at Scientific Reports on February 3rd, 2022. See the published version at <https://doi.org/10.1038/s41598-022-05971-9>.

Clinical Data Classification With Noisy Intermediate Scale Quantum Computers

S. Moradi¹, C. Brandner¹, C. Spielvogel², D. Krajnc¹, S. Hillmich³, R. Wille^{3,4}, W. Drexler¹, L. Papp^{1*}

¹ Center for Medical Physics and Biomedical Engineering, Medical University of Vienna, Vienna, Austria

² Division of Nuclear Medicine, Medical University of Vienna, Vienna, Austria

³ Institute for Integrated Circuits, Johannes Kepler University Linz, Linz, Austria

⁴ Software Competence Center Hagenberg GmbH, Hagenberg, Austria

*Corresponding author: L. Papp

Center for Medical Physics and Biomedical Engineering, Medical University of Vienna, Vienna, Austria.

Address: Währinger Gürtel 18-20, 1090 Vienna, Austria

E-mail: Laszlo.papp@meduniwien.ac.at

Tel: +4340400 72350

Keywords: Quantum computing, quantum machine learning, classic data encoding, clinical prediction models

ABSTRACT

Quantum machine learning has experienced a significant progress in both software and hardware development in the recent years and has emerged as an applicable area of near-term quantum computers. In this work, we investigate the feasibility of utilizing quantum machine learning (QML) on real clinical datasets. We propose two QML algorithms for data classification on IBM quantum hardware: a quantum distance classifier (qDS) and a simplified quantum-kernel support vector machine (sqKSVM). We utilize these different methods using the linear time quantum data encoding technique ($\log_2 N$) for embedding classical data into quantum states and estimating the inner product on 15-qubit IBMQ Melbourne quantum computer. We match the predictive performance of our QML approaches with prior QML methods and with their classical counterpart algorithms for three open-access clinical datasets. Our results imply that the qDS in small sample and feature count datasets outperforms kernel-based methods. In contrast, quantum kernel approaches outperform qDS in high sample and feature count datasets. We demonstrate that the $\log_2 N$ encoding increases predictive performance with up to +2% area under the receiver operator characteristics curve across all quantum machine learning approaches, thus, making it ideal for machine learning tasks executed in Noisy Intermediate Scale Quantum computers.

INTRODUCTION

Quantum technologies promise to revolutionize the future of information and computation using quantum devices to process massive amount of data. To date, considerable progress has been made from both software and hardware points of view. Many researches are underway to simplify quantum algorithms¹⁻⁸ in order to implement them on existing, so-called Noisy Intermediate Scale Quantum (NISQ) computers⁹. As a result, small quantum devices based on photons, superconductors, or trapped ions are capable of efficiently running scalable quantum algorithms^{6,7,10}. Quantum Machine Learning (QML) is a particularly interesting approach, as it is suited for existing NISQ architectures¹¹⁻¹⁵. While conventional machine learning is generally applied to process large amounts of data, many research fields cannot provide such large datasets. One example is medical research, where collecting cohorts that represent certain characteristics of diseases routinely results in small datasets¹⁶. NISQ devices can efficiently execute algorithms with shallow depth and a low number of qubits⁹. Therefore, it appears logical to exploit the potential of QML executed on NISQ devices incorporating clinical datasets.

However, the execution of QML algorithms in the form of practical quantum gate operations is non-trivial. First, the classical data needs to be encoded into quantum states. For this purpose,

prior QML algorithms assume that a quantum random access memory (QRAM) device for storing the data is present ¹⁷. Nevertheless, to date, such practical device is not available. Second, since the output of quantum algorithms are obviously quantum states, the efficient classical bits of information must be extracted through quantum measurements. To date, various classical data encoding approaches have been proposed ^{6,7,18–21}. In particular, encoding classical numerical features into quantum states has the advantage to utilize $\log_2 N$ number of qubits (a.k.a. linear time encoding) in relation to N number of input features ^{18–21}. This approach allows to utilize NISQ devices with a small numbers of qubits and to minimize quantum noise, while at the same time maintaining quantum speedup ¹⁴. In contrast, to date, this approach in combination with quantum machine learning appears to be underrepresented.

In light of the above proceedings, we hypothesize that clinically-relevant quantum prediction models can be built on NISQ devices employing $\log_2 N$ encoding, having prediction performances comparable to classic ML approaches.

In our work, we propose two quantum machine learning approaches that rely on the $\log_2 N$ encoding approach. First, we demonstrate a simple and efficient quantum distance classifier (qDC) executable on existing NISQ devices. Second, we present a simplified quantum-kernel SVM (sqKSVM) approach using quantum kernels which can be executed once without optimization instead of twice with optimization as in case of the quantum-kernel SVM (qKSVM) approach ^{6,7}.

In order to test our hypothesis, we demonstrate the performance of the qDC and the sqKSVM approaches using real clinical data and compare their performances to qKSVM, as well as to classic computing counterparts such as k-nearest neighbors ²² and classic support vector machines ²³.

RESULTS

Dataset

This study incorporated three open-access clinical datasets that have been presented and evaluated in various contexts ^{24–26}. Each dataset underwent redundancy reduction by correlation matrix analysis ²⁷ followed by a 10-fold cross-validation split with a training-validation ratio of 80%-20% ¹⁶. Training sets of the folds were subjects of feature ranking analysis ²⁸ and the highest-ranking eight as well as 16 (if available) features were selected for further analysis. The resulted dataset configurations were analyzed by class imbalance ratios and the quantum advantage

score (a.k.a. difference geometry) ²⁰ for quantum kernel methods. Table 1 demonstrates the characteristics of the data configurations as well as the results of the imbalance ratio and the quantum advantage scores (for estimation of the quantum advantage scores (g_{CQ}), see Appendix E of the supplementary material).

Dataset	#Samples	Imbalance Ratio	#Features	g_{CQ}	Reference
Pediatric Bone Marrow Transplant 2-year survival	134	0.33	8	0.40	24
			16	0.60	
Wisconsin Breast Cancer Malign-vs-benign	569	0.37	8	1.30	25
			16	3.50	
Heart Failure Mortality	300	0.5	8	0.42	26

Table 1. Clinical datasets utilized for the study with their sample and selected feature count as well as their imbalance ratios and quantum advantage scores (g_{CQ}). Given a two-class dataset, the imbalance ratio (IR) is $IR = x/y$, where x is the number of minority class and y is the total number of samples. Furthermore, g_{CQ} measures the similarities of quantum kernel and linear classical kernel functions of the same dataset.

Encoding strategies

This study relies on the data encoding strategy which uses sequences of Pauli-Y gate rotations (R_y) and $CNOT$ gates (see Appendix A of the supplementary material) to result in a number of $\log_2 N$ encoding qubits ^{18,19,21}. R_y puts each qubit q in a superposition state $R_y(2\theta)|q\rangle = \cos(\theta)|0\rangle \pm \sin(\theta)|1\rangle$ and $CNOT$ s entangle qubits. The data encoding feature map with the application of R_y and $CNOT$ is given by ²⁹

$$\varphi: \vec{x} \rightarrow |\varphi(\vec{\theta})\rangle\langle\varphi(\vec{\theta})| \quad (1)$$

where $|\varphi(\vec{\theta})\rangle = U_{\varphi(\vec{\theta})}|0\rangle^{\otimes \log_2 N}$. In equation (1) $\varphi(\vec{\theta})$ is the encoding map from the Euclidean space to the Hilbert space and $U_{\varphi(\vec{\theta})}$ is the model circuit for data encoding, which maps $|0\rangle^{\otimes \log_2 N}$ to another ket vector of input data $|\varphi(\vec{\theta})\rangle$. To find a relationship between the input data and θ , see Appendix A of the supplementary material.

In contrast, previously proposed quantum ML-specific encoding utilizes a block of the Hadamard gates followed by a block of Pauli-Z gate rotations (R_z) are applied to each qubit ⁷. To entangle the qubits, nearest neighbor $CNOT$ s are also applied. The features of data samples are considered as angles of R_z rotations and the required number of qubits for data encoding are

equal to the number of features. The data encoding feature map with the application of the Hadamard, R_z and $CNOT$ is given by ⁷

$$\varphi: \vec{x} \rightarrow |\varphi(\vec{x})\rangle\langle\varphi(\vec{x})| \quad (2)$$

where $|\varphi(\vec{x})\rangle = U_{\varphi(\vec{x})} H^{\otimes N} |0\rangle^{\otimes N}$. $U_{\varphi(\vec{x})}$ is the model circuit for N features data encoding.

In order to compare the predictive performance of the above two data encoding strategies, the qDC, the sqKSVM and the qKSVM (see Appendix C of the supplementary material) approaches were compared utilizing a number of $N = 8$ features. This analysis was executed using the PennyLane simulator environment ³⁰, while the sqKSVM was also evaluated on the IBMQ Melbourne machine (see Methods). Table 2 demonstrates the cross-validation AUC performance values of the quantum ML algorithms in relation to the $\log_2 N$ and N encoding qubit strategies.

Dataset	qDC	qKSVM	sqKSVM	sqKSVM*	qubits
Pediatric Bone Marrow Transplant 2YS	0.62	0.63	0.62	0.61	$\log_2 N$
	0.61	0.63	0.61	0.59	N
Wisconsin Breast Cancer Malign-vs-benign	0.92	0.92	0.88	0.87	$\log_2 N$
	0.90	0.91	0.87	0.85	N
Heart failure Mortality	0.62	0.51	0.51	0.50	$\log_2 N$
	0.60	0.51	0.51	0.50	N

Table 2. Comparison of the cross-validation AUC performance for different data encodings. The qDC, qKSVM, and sqKSVM run on PennyLane simulator for $N = 8$. For $\log_2 N$ encoding, N features are encoded into $\log_2 N$ qubits with sequences of Pauli-Y gate rotations (R_y) and $CNOT$ s. In another strategy, N features are encoded into N qubits with sequences of the Hadamard gates, Pauli-Z gate (R_z) rotations followed by nearest neighbor $CNOT$ s.

*The sqKSVM was also executed on the IBMQ Melbourne machine for reference comparison.

Quantum and classic machine learning predictive performance evaluation

The quantum distance classifier (qDC) first calculates the distance between the state vector of a test sample and each state vector of the training sample in set P and set Q and, then, assigns a label of the test sample to the label of the closest set. In the qDC, we divide the training set, with M number of samples, based on their labels $\{a, b \in R\}$ into two subset $\{P\}$ and $\{Q\}$, where $\{P\}$ contains only label a with the number of samples MP and $\{Q\}$ contains only label b with the number of samples MQ with $MP + MQ = M$. The task is to determine the label of the given test sample $\{y_k\}$, if $y_k = a$ or $y_k = b$. Mathematically, if $|v\rangle$ is the state vector of the test sample as

well as $|u\rangle \in P$ and $|w\rangle \in Q$, then the label of $|v\rangle$ is determined by $y_k = a$, if $\min(|\langle u| - |v\rangle|) \leq \min(|\langle w| - |v\rangle|)$, otherwise $y_k = b$. The distance between the vectors is given by ⁸ i.e.

$$|\langle u| - |v\rangle| = |\langle u|||v\rangle| - \langle u|v\rangle| \quad (3)$$

where $|\cdot|$ is the norm l_2 of a vector. Therefore, the task is to calculate the inner product $\langle u|v\rangle$ with a quantum computer.

The two different approaches to estimate $\langle u|v\rangle$ with quantum computers are Hadamard Test ³¹ and the Swap Test ³². For the simplified quantum kernel SVM (sqKSVM), we first need to note that the standard form of the quantum kernelized binary classifiers is

$$\tilde{y} = \text{sgn}\left(\sum_{i=1}^M y_i \alpha_i^* K(\vec{x}_i, \vec{\tilde{x}})\right) \quad (4)$$

where $\tilde{y}$ is the unknown label, y_i is the label of the i^{th} training sample, α_i^* is the i^{th} component of the support vector $\vec{\alpha}^* = (\alpha_1^*, \alpha_2^*, \dots, \alpha_M^*)$, M is the number of training data, and $K(\vec{x}_i, \vec{\tilde{x}})$ is the kernel matrix of all the training-test pairs.

For a given dataset $D = \{(\vec{x}_i, y_i): \vec{x}_i \in R^M, y_i \in \{-1, 1\}\}_{i=1, \dots, M}$, one option to bypass the drawbacks of the qKSVM algorithm (see Appendix C of the supplementary material) as presented in ^{6,7} is to set uniform weight $\alpha_i^* = 1$, in case of $IR = 0.5$ (balanced dataset). Otherwise, $\alpha_i^* = IR$ for the majority class and $\alpha_j^* = 1 - IR$ for the minority class. Thresholding the value $\sum_{i=1}^M y_i \alpha_i^* K(\vec{x}_i, \vec{\tilde{x}})$ yields the binary output as following

$$\tilde{y} = \begin{cases} 1 & \sum_{i=1}^M y_i \alpha_i^* K(\vec{x}_i, \vec{\tilde{x}}) \geq 0 \\ -1 & \text{else} \end{cases} \quad (5)$$

In equation (5), $K(\vec{x}_i, \vec{\tilde{x}})$ is defined as (see Appendix F of the supplementary material)

$$K(\vec{x}_i, \vec{\tilde{x}}) = |\langle u|v\rangle|^2 \quad (6)$$

The dataset configurations were utilized to estimate the performance of quantum and classic machine learning algorithms incorporated in this study. Performance estimation was done by confusion matrix analytics ³³. Prediction models were built based on the given training subset, followed by evaluating the respective validation subsets of each fold. Average area under the

receiver operator characteristics curve (AUC) was calculated across validation cases for each predictive model. To build predictive models, quantum ML approaches included the qDC, the sqK SVM and the qK SVM (see Appendix C of supplementary material) were utilized. Classic machine learning approaches were k-nearest neighbors (ckNN)²² and support vector machines (cSVM)²³. See Table 3 for the comparison of cross-validation AUC performances of quantum and classic computing algorithm within the dataset configurations.

Dataset	#Features	sqK SVM	qK SVM	qDC	cSVM	ckNN
Pediatric Bone Marrow Transplant 2YS	8	0.61	0.63	0.60	0.64	0.61
	16	0.66	0.69	0.64	0.71	0.64
Wisconsin Breast Cancer Malign-vs-benign	8	0.87	0.92	0.91	0.89	0.90
	16	0.88	0.93	0.90	0.89	0.93
Heart Failure Mortality*	8	0.50	0.51	0.60	0.53	0.58

Table 3. Comparison of the cross-validation AUC performance with QML and ML algorithms. For all QML algorithms, N features are encoded into $\log_2 N$ qubits with sequences of Pauli-Y gate rotations (R_y) and $CNOT$ s. All QML algorithms were executed on the IBMQ Melbourne machine.

*Heart failure has no 16-feature variant, since the maximum number of features are 13.

DISCUSSION

In this study, we aimed to investigate the effect of two encoding strategies in various quantum machine learning-built clinical prediction models. Next to prior quantum machine learning approaches, we also proposed two methods specifically designed for the $\log_2 N$ encoding approach.

Our results demonstrate that the $\log_2 N$ encoding in combination with low-complexity quantum machine learning approaches provides comparable or better results than the N encoding approach with previously-proposed quantum machine learning methods. This advantage was demonstrated not only in a simulator environment, but also utilizing NISQ devices. The low algorithmic quantum complexity also aims towards building prediction models that may be easier to interpret in the future, especially in light of the high complexity of classic machine learning approaches³⁴. In contrast, it is important to emphasize, that the proposed quantum machine learning processes are also applicable in big data, given, that calculating the inner product of quantum states in NISQ devices can be done efficiently with the $\log_2 N$ encoding approach^{21,31}. The $\log_2 N$ data encoding we used is also quite robust against noise, since they use a small part of the entire Hilbert space of the quantum device to estimate the inner product of quantum states

When feature count increases, g_{CQ} increases as well, because quantum state vectors of input features become closer due to the high dimensionality property of the Hilbert space. Higher feature count significantly influences performance in a positive way if g_{CQ} is < 1 (e.g. +5-6% AUC in the Pediatric bone marrow dataset). It has been shown that classical ML models are competitive or outperform quantum ML approaches when g_{CQ} is small ²⁰. Nevertheless, we demonstrated that when $g_{CQ} > 1$, higher feature count does not contribute much to the performance increase (e.g. 1% difference in the Wisconsin breast cancer dataset). It is important to point out that a high g_{CQ} (> 1) alone does not mean that the dataset is not ideal for kernel-based quantum machine learning. Specifically, the highest AUC of 0.93 was achieved in the 16 feature counts Wisconsin breast cancer dataset, while it also demonstrated the highest g_{CQ} , which also confirms prior findings ²⁰. In contrast, the same dataset in the classic SVM resulted in 0.89 AUC. We hypothesize that this phenomenon is due to the high sample count of the Wisconsin breast cancer dataset ($M=569$). In general, the imbalance ratio of the datasets did not appear to be correlated with predictive performance. The $\log_2 N$ increased AUC with up to 2% compared to the N encoding when comparing the execution of the quantum machine learning approaches using simulation environment. This behavior was also identifiable with executions in NISQ devices, in case of kernel methods and the qDC. We hypothesize that lower AUC performance for the N encoding method in the simulator environment and NISQ device is due to higher number of qubits which likely lead to lower value of inner products. This is in line with the findings in ²⁰.

In general, the qKSVM demonstrated 2-5% higher AUC compared to the sqKSVM. The relative performance increase of the qKSVM was in relation to sample count and feature count. Specifically, the qKSVM showed an average 2% higher AUC with small sample count (Heart failure and Pediatric bone datasets), while it had 5% higher AUC in the Wisconsin breast cancer dataset. Nevertheless, both the qKSVM and the sqKSVM increased its AUC with double feature counts in the small Pediatric bone marrow dataset. This level of performance increase was not identifiable in the larger Wisconsin breast cancer dataset. Classic SVM demonstrated similar properties in relation to higher feature counts in small datasets ²⁰, while it was outperformed by the qKSVM in the large Wisconsin breast cancer dataset.

In conclusion, quantum SVM approaches benefit from higher feature count in general, where the qKSVM – due to relying on optimization – has a particular benefit compared to the sqKSVM. In the large Wisconsin breast cancer dataset, the qDC demonstrated higher performance compared to the sqKSVM, especially in small feature counts (0.91 AUC vs 0.87 AUC in the qDC and the sqKSVM respectively in 8 features). The qDC resulted in the highest AUC of 0.60 across all other

quantum (0.50-0.51 AUC) and classic machine learning (0.53-0.58 AUC) approaches in the Heart failure dataset. We hypothesize that this is due to the distribution characteristics of the samples belonging to the two subclasses in the feature space, which challenges classification with kernel methods. Generally, the performance of the executed quantum and classic machine learning approaches are comparable within the collected cohorts (Table 3).

According to our findings, quantum distance approaches can provide high performance with small feature and sample counts, which is particularly ideal for NISQ devices. In contrast, quantum kernel methods appear to provide high performance with high feature and sample counts. We demonstrated that the $\log_2 N$ encoding strategy allows to execute quantum ML algorithms for highly dimensional clinical datasets on low qubit count NISQ devices. In general, quantum machine learning benefits from utilizing the $\log_2 N$ encoding strategy, as it increases predictive performance and reduces execution time in NISQ devices, while keeping model complexity lower. We consider these findings of high importance in relation to building future quantum ML prediction models for clinically-relevant cohorts.

METHODS

All experiments of this study were performed in accordance with the respective guidelines and regulations of the open-access data sources this study relied on. For details, see section “Access”.

Estimation of the inner product $\langle u|v \rangle$ and $|\langle u|v \rangle|^2$

Fig. 1 shows the quantum circuit for estimation of the real part of $\langle u|v \rangle$ with the Hadamard Test.

Figure 1. Quantum circuit computes the real part of the inner product $\langle u|v \rangle$. The Hadamard gate puts the ancilla qubit (q_0) into uniform superposition. A single-controlled unitary gate entangles the excited state of the ancilla qubit with the training data state vector ($|u\rangle = U|q_1q_2q_3\rangle$). The X gate flips the ancilla qubit. Another single unitary controlled gate entangles the state vector of the test data ($|v\rangle = V|q_1q_2q_3\rangle$) with the excited state of the ancilla qubit. A second X gate flips the ancilla qubit. The Hadamard gate on the ancilla qubit interferes train and test data state vectors. The ancilla

qubit is measured using a Pauli-Z gate. The real value of $\langle u|v \rangle$ is estimated from equation (9). The measurement gate is done by a Pauli-Z gate and $Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$.

To estimate the real part of $\langle u|v \rangle$ on the quantum computer with the Hadamard Test, the training and test data needs to be prepared in a quantum state as

$$\frac{1}{\sqrt{2}}(|0\rangle_a |v\rangle + |1\rangle_a |u\rangle) \quad (7)$$

where $|u\rangle$ and $|v\rangle$ are the quantum states for the train and test datasets, respectively.

Then the Hadamard gate on the ancilla qubit interferences the training vector $|u\rangle$ with the test vector $|v\rangle$

$$\frac{1}{2}(|0\rangle_a (|v\rangle + |u\rangle) + |1\rangle_a (|v\rangle - |u\rangle)) \quad (8)$$

Finally, the measuring quantum state given in equation (8) in the computational basis $|0\rangle_a$ gives probability as

$$\Pr(|0\rangle_a) = \frac{(1 + \text{Re}(\langle u|v \rangle))}{2} \quad (9)$$

where $\Pr$ is the value of the probability of measurement on the $|0\rangle_a$ state of equation (8) and $\langle u|u \rangle = \langle v|v \rangle = 1$. Since our datasets are real values $\text{Re}(\langle u|v \rangle) = \langle u|v \rangle$.

The inner product $\langle u|v \rangle$ can also be estimated on a quantum computer with the Swap Test (see Fig. 2). The Hadamard gate is applied on the ancilla qubit to create a superposition of $|u\rangle|v\rangle$, i.e.

$$\frac{1}{\sqrt{2}}(|0\rangle_a |u\rangle|v\rangle + |1\rangle_a |u\rangle|v\rangle) \quad (10)$$

The application of the single-controlled swap gates on the state given in equation (10) entangles the ancilla qubit with $|u\rangle|v\rangle$. The resulted entangled quantum state is $\frac{1}{\sqrt{2}}(|0\rangle_a |u\rangle|v\rangle + |1\rangle_a |v\rangle|u\rangle)$. Then, another Hadamard gate interferences the product state of the state vectors of the training and the test i.e.

$$\frac{1}{2}(|0\rangle_a (|u\rangle|v\rangle + |v\rangle|u\rangle) + |1\rangle_a (|u\rangle|v\rangle - |v\rangle|u\rangle)) \quad (11)$$

Measuring the quantum state given in equation (11) in the computational basis $|0\rangle_a$ yields the probability

$$\Pr(|0\rangle_a) = \frac{(1 + |\langle u|v\rangle|^2)}{2} \quad (12)$$

where $\Pr$ is the value of the probability of measurement on the $|0\rangle_a$ state of equation (11).

Figure 2. Quantum Circuit to compute $|\langle u|v\rangle|^2$. The model circuits encode train and test data into quantum states $|u\rangle = U|q_1q_2q_3\rangle$ and $|v\rangle = V|q_1q_2q_3\rangle$. The Hadamard gate on the ancilla qubit (q_0) generates a superposition of the quantum state including the train and test datasets. The application of the single-controlled swap gates with the ancilla qubit as the control results in an entangled state of equation (10). Another Hadamard gate on the ancilla qubit interferes $|u\rangle|v\rangle$ and $|v\rangle|u\rangle$. The ancilla qubit on the $|0\rangle$ state is measured with Pauli-Z gate. Therefore, the value of $|\langle u|v\rangle|^2$ can be obtained from equation (12).

Simplified Quantum Kernel Support Vector Machine

The quantum Support Vector Machine algorithm is proposed in ³⁵ for big data classification. They show exponential speedup for their algorithm via quantum mechanically access to data. Nevertheless, this approach is not ideal for NISQ devices ⁹. To date, two separate qKSVM approaches are proposed for data classification via classical access to data ^{6,7}. In these approaches, the quantum circuits must run twice on the quantum computer and a cost function needs to be optimized on the classical computer to compute the support vector ⁷. We propose a simplified version qKSVM called sqKSVM as shown in Fig. 3.

Figure 3. Schematic of the sqKSVM for data classification algorithm. First, the training data vector $\vec{x}$ and test $\tilde{x}$ are prepared on a classical computer. Next, the original training data and test data are encoded into quantum states followed by computing the kernel matrix of all pairs of the training-test data $K(\vec{x}_i, \tilde{x})$ with a NISQ computer. If $\alpha^* = (\alpha_1^*, \alpha_2^*, \dots, \alpha_M^*)$ are considered to be a solution of the support vector, the binary classifier can be constructed based on equation (5).

Software and Hardware

For classical machine learning algorithms, we use classical machine learning (CML) libraries of scikit-learn³⁶. PennyLane-Qiskit³⁰ is used for quantum circuit simulation and quantum experiment for designing quantum computing programs. PennyLane-Qiskit 0.13.0 plugin integrates the Qiskit quantum computing framework to the PennyLane simulator.

For executing quantum algorithms on existing quantum computers, this study relied on IBM's remote quantum machines (<https://quantum-computing.ibm.com/>) that can run quantum programs with noisy qubits. Since IBM quantum computers only support single qubit gate and two-qubit *CNOT* gate operations, complex gate operations must be decomposed into elementary supported gates before mapping the quantum circuit on noisy hardware. Owing to the specific architecture of IBM quantum computers, the coupling map must satisfy all two qubit *CNOT* gate operations³⁷, 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 . In case of running the QML algorithms on the quantum computer, we choose the 15-qubits IBMQ Melbourne machine with the standard gates I, U_1, U_2, U_3 , and *CNOT*, where I is identity single qubit gate, U_1, U_2, U_3 are single qubit arbitrary rotation gates with *CNOT* as two-qubit gate. Fig. 4 shows the coupling map of IBMQ Melbourne with its gate error rates.

Figure 4. Topology and coupling map of IBMQ Melbourne (<https://quantum-computing.ibm.com/services>). Singlequbit error rate is the error induced by applying the single qubit gates. *CNOT* error is the error of the only two-qubit *CNOT* gates. Each circle represents a physical superconducting qubit and each $\leftrightarrow$ shows coupling between nearest neighbor qubits.

Implementation of the Hadamard Test on IBMQ Melbourne machine

The Hadamard Test approach is used to estimate the inner product of two state vectors. To map the quantum circuit of the Hadamard Test (Fig. 1) on IBMQ Melbourne machine, first, we design the circuit in the PennyLane-Qiskit 0.13.0 environment. As can be seen from Fig. 5, there are Toffoli gates (*CCNOT*), and single control R_y gates which are needed to be decomposed before mapping the circuit of Fig. 5 on the IBMQ Melbourne machine architecture. To this end, we decompose each Toffoli gate in Fig. 5 into Hadamard (H), T , $T^\dagger$, and *CNOT* gates and each single control R_y gates into single qubit rotation gates and two *CNOT*s (see Appendix B of the supplementary material). To design the Hadamard Test dataset with 16 features, *CNOT*(q_0, q_4) cannot be implemented on IBMQ Melbourne machine directly due to coupling constraints³⁷. There are two approaches to map *CNOT*(q_0, q_4) on IBMQ Melbourne. Let's first consider the physical qubits path Q_4 (root node), Q_{10} , Q_9 , Q_8 , and Q_7 (goal node) from Fig. 4 as simulator qubits q_0 , q_1 , q_2 , q_3 , and q_4 , respectively. The first option is to use a SWAP gate between Q_4 and Q_{10} , then, a *CNOT* between Q_4 and Q_7 . To complete the effect, a final SWAP between Q_{10} and Q_4 is applied^{37,38}. Since each SWAP gate is implemented using three *CNOT* gates, seven *CNOT*s are required to reproduce the effect of just one *CNOT*(q_0, q_4). Another more efficient way is to only use sequences of *CNOT* gates as following³⁸

$$CNOT(Q_4, Q_7) = CNOT(Q_4, Q_{10})CNOT(Q_{10}, Q_7)CNOT(Q_4, Q_{10})CNOT(Q_{10}, Q_7), \quad (13)$$

where *CNOT*(q_0, q_4) = *CNOT*(Q_4, Q_7).

For the implementation of SWAP test on IBMQ Melbourne machine see Appendix D in the supplementary material.

Figure 5. The Hadamard Test circuit to estimate the inner product of state vectors of the training and the test datasets with 8 features (3+1-qubit). The ancilla qubit (q_0) is measured to estimate $\langle u|v \rangle$. The blue boxes are the part of quantum circuit that encodes the train data into the quantum state and entangles the ancilla qubit with the quantum state vector of the train data. The quantum circuit in the green boxes encode the test data and also entangle quantum state vector of the test data with the ancilla qubit.

ACCESS

The source code of the implemented quantum algorithms can be accessed by the link:

<https://github.com/sassan72/Quantum-Machine-learning>

For classical machine learning executions, this study relied on the scikit-learn library ³⁶:

<https://scikit-learn.org/stable/>

All included open-access datasets are accessible through the following links:

Pediatric bone marrow transplant dataset ²⁴:

<https://archive.ics.uci.edu/ml/datasets/Bone+marrow+transplant%3A+children>

Wisconsin breast cancer dataset ²⁵:

[https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+\(Diagnostic\)](https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic))

Heart failure dataset ²⁶:

<https://www.kaggle.com/andrewmvd/heart-failure-clinical-data>

REFERENCES

1. Suzuki, Y. *et al.* Amplitude estimation without phase estimation. *Quantum Inf. Process.* **19**, 75 (2020).
2. Tanaka, T. *et al.* Amplitude estimation via maximum likelihood on noisy quantum computer. *Quantum Inf. Process.* **20**, 293 (2021).
3. Grinko, D., Gacon, J., Zoufal, C. & Woerner, S. Iterative quantum amplitude estimation. *npj Quantum Inf.* **7**, 52 (2021).
4. Aaronson, S. & Rall, P. Quantum Approximate Counting, Simplified. (2019) doi:10.1137/1.9781611976014.5.
5. Bouland, A., van Dam, W., Joorati, H., Kerenidis, I. & Prakash, A. Prospects and challenges of quantum finance. (2020).
6. Schuld, M. & Killoran, N. Quantum Machine Learning in Feature Hilbert Spaces. *Phys. Rev. Lett.* **122**, 040504 (2019).
7. Havlíček, V. *et al.* Supervised learning with quantum-enhanced feature spaces. *Nature* **567**, 209–212 (2019).
8. Wiebe, N., Kapoor, A. & Svore, K. Quantum Algorithms for Nearest-Neighbor Methods for Supervised and Unsupervised Learning. (2014).
9. Preskill, J. Quantum Computing in the NISQ era and beyond. *Quantum* **2**, 79 (2018).
10. Johri, S. *et al.* Nearest centroid classification on a trapped ion quantum computer. *npj Quantum Inf.* **7**, 122 (2021).
11. Saggio, V. *et al.* Experimental quantum speed-up in reinforcement learning agents. *Nature* **591**, 229–233 (2021).
12. Steinbrecher, G. R., Olson, J. P., Englund, D. & Carolan, J. Quantum optical neural networks. *npj Quantum Inf.* **5**, 60 (2019).
13. Peters, E. *et al.* Machine learning of high dimensional data on a noisy quantum processor. (2021).
14. Liu, Y., Arunachalam, S. & Temme, K. A rigorous and robust quantum speed-up in

- supervised machine learning. *Nat. Phys.* **17**, 1013–1017 (2021).
15. Hubregtsen, T. *et al.* Training Quantum Embedding Kernels on Near-Term Quantum Computers. (2021).
 16. Papp, L., Spielvogel, C. P., Rausch, I., Hacker, M. & Beyer, T. Personalizing Medicine Through Hybrid Imaging and Medical Big Data Analysis. *Front. Phys.* **6**, (2018).
 17. Giovannetti, V., Lloyd, S. & Maccone, L. Architectures for a quantum random access memory. *Phys. Rev. A* **78**, 052310 (2008).
 18. Schuld, M. & Petruccione, F. *Supervised Learning with Quantum Computers*. (Springer International Publishing, 2018). doi:10.1007/978-3-319-96424-9.
 19. Möttönen, M., Vartiainen, J. J., Bergholm, V. & Salomaa, M. M. Quantum Circuits for General Multiqubit Gates. *Phys. Rev. Lett.* **93**, 130502 (2004).
 20. Huang, H.-Y. *et al.* Power of data in quantum machine learning. *Nat. Commun.* **12**, 2631 (2021).
 21. Schuld, M., Bocharov, A., Svore, K. M. & Wiebe, N. Circuit-centric quantum classifiers. *Phys. Rev. A* **101**, 032308 (2020).
 22. Zhang, Z. Introduction to machine learning: k-nearest neighbors. *Ann. Transl. Med.* **4**, 218–218 (2016).
 23. Chang, C.-C. & Lin, C.-J. LIBSVM. *ACM Trans. Intell. Syst. Technol.* **2**, 1–27 (2011).
 24. Sikora, M., Wróbel, Ł. & Gudyś, A. GuideR: A guided separate-and-conquer rule learning in classification, regression, and survival settings. *Knowledge-Based Syst.* **173**, 1–14 (2019).
 25. Dua, D. & Graff, C. {UCI} Machine Learning Repository. *University of California, Irvine, School of Information and Computer Sciences* <http://archive.ics.uci.edu/ml> (2017).
 26. Chicco, D. & Jurman, G. Machine learning can predict survival of patients with heart failure from serum creatinine and ejection fraction alone. *BMC Med. Inform. Decis. Mak.* **20**, 16 (2020).
 27. Krajnc, D. *et al.* Breast Tumor Characterization Using [18F]FDG-PET/CT Imaging Combined with Data Preprocessing and Radiomics. *Cancers (Basel)*. **13**, 1249 (2021).
 28. Papp, L. *et al.* Supervised machine learning enables non-invasive lesion characterization in primary prostate cancer with [68Ga]Ga-PSMA-11 PET/MRI. *Eur. J. Nucl. Med. Mol. Imaging* (2020) doi:10.1007/s00259-020-05140-y.
 29. Schuld, M. Supervised quantum machine learning models are kernel methods. (2021).
 30. Bergholm, V. *et al.* PennyLane: Automatic differentiation of hybrid quantum-classical computations. (2018).

31. Schuld, M., Fingerhuth, M. & Petruccione, F. Implementing a distance-based classifier with a quantum interference circuit. *EPL (Europhysics Lett.* **119**, 60002 (2017).
32. Blank, C., Park, D. K., Rhee, J.-K. K. & Petruccione, F. Quantum classifier with tailored quantum kernel. *npj Quantum Inf.* **6**, 41 (2020).
33. Luque, A., Carrasco, A., Martín, A. & de las Heras, A. The impact of class imbalance in classification performance metrics based on the binary confusion matrix. *Pattern Recognit.* **91**, 216–231 (2019).
34. Carvalho, D. V., Pereira, E. M. & Cardoso, J. S. Machine Learning Interpretability: A Survey on Methods and Metrics. *Electronics* **8**, 832 (2019).
35. Rebentrost, P., Mohseni, M. & Lloyd, S. Quantum Support Vector Machine for Big Data Classification. *Phys. Rev. Lett.* **113**, 130503 (2014).
36. Pedregosa, F. *et al.* Scikit-learn: Machine Learning in Python. (2012).
37. Wille, R., Hillmich, S. & Burgholzer, L. Efficient and Correct Compilation of Quantum Circuits. in *2020 IEEE International Symposium on Circuits and Systems (ISCAS)* 1–5 (IEEE, 2020). doi:10.1109/ISCAS45731.2020.9180791.
38. Bataille, M. Quantum circuits of CNOT gates. (2020).

AUTHOR CONTRIBUTIONS

All co-authors of this paper were involved in the conceptualization of the study, in writing, in the critical review as well as final approval to submit this paper. Specific author contributions are as follows:

S. Moradi: Quantum machine learning design, implementation and execution

C. Brandner: Development environment support, algorithm design

C. Spielvogel: Classic machine learning comparative analysis

D. Krajnc: Data preparation, cross-validation scheme, performance evaluation

S. Hillmich, R. Wille: Simulator environment establishment, support in algorithm optimization and execution

W. Drexler, L. Papp: Study design, supervision

COMPETING INTERESTS

The authors declare no competing interests.

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- [MoradietalNatSciRep2021Supplemntal.pdf](#)