

Singapore Management University

## Institutional Knowledge at Singapore Management University

---

Research Collection School Of Computing and Information Systems

School of Computing and Information Systems

---

7-2022

### Review of some existing QML frameworks and novel hybrid classical-quantum neural networks realising binary classification for the noisy datasets

N. Schetakis

D. Aghamalyan

Paul Robert GRIFFIN

Singapore Management University, paulgriffin@smu.edu.sg

M. Boguslavsky

Follow this and additional works at: [https://ink.library.smu.edu.sg/sis\\_research](https://ink.library.smu.edu.sg/sis_research)



Part of the [Artificial Intelligence and Robotics Commons](#), and the [Theory and Algorithms Commons](#)

---

#### Citation

Schetakis, N.; Aghamalyan, D.; GRIFFIN, Paul Robert; and Boguslavsky, M.. Review of some existing QML frameworks and novel hybrid classical-quantum neural networks realising binary classification for the noisy datasets. (2022). *Scientific Reports*. 12, (1), 1-12.

Available at: [https://ink.library.smu.edu.sg/sis\\_research/7214](https://ink.library.smu.edu.sg/sis_research/7214)

This Journal Article is brought to you for free and open access by the School of Computing and Information Systems at Institutional Knowledge at Singapore Management University. It has been accepted for inclusion in Research Collection School Of Computing and Information Systems by an authorized administrator of Institutional Knowledge at Singapore Management University. For more information, please email [cherylds@smu.edu.sg](mailto:cherylds@smu.edu.sg).



OPEN

## Review of some existing QML frameworks and novel hybrid classical–quantum neural networks realising binary classification for the noisy datasets

N. Schetak<sup>1,2</sup>, D. Aghamalyan<sup>3,4,5,7</sup>, P. Griffin<sup>3,7</sup> & M. Boguslavsky<sup>6,7</sup>

One of the most promising areas of research to obtain practical advantage is Quantum Machine Learning which was born as a result of cross-fertilisation of ideas between Quantum Computing and Classical Machine Learning. In this paper, we apply Quantum Machine Learning (QML) frameworks to improve binary classification models for noisy datasets which are prevalent in financial datasets. The metric we use for assessing the performance of our quantum classifiers is the area under the receiver operating characteristic curve AUC–ROC. By combining such approaches as hybrid-neural networks, parametric circuits, and data re-uploading we create QML inspired architectures and utilise them for the classification of non-convex 2 and 3-dimensional figures. An extensive benchmarking of our new FULL HYBRID classifiers against existing quantum and classical classifier models, reveals that our novel models exhibit better learning characteristics to asymmetrical Gaussian noise in the dataset compared to known quantum classifiers and performs equally well for existing classical classifiers, with a slight improvement over classical results in the region of the high noise.

Noisy Intermediate-Scale Quantum (NISQ)<sup>1–3</sup> devices hold a promise to deliver a practical quantum advantage by harnessing the complexity of quantum systems. Despite being several years away from having fault-tolerant quantum computing<sup>4–6</sup>, researchers have been hopeful to achieve this task. Perhaps one of the most exciting breakthroughs in this direction was a demonstration of “quantum supremacy” by Google researchers<sup>7</sup>, using their programmable superconducting Sycamore chip with 53 qubits, in which single-qubit gate fidelities of 99.85% and two-qubit gate fidelities of 99.64% were obtained on average. Here the task of sampling the output of a pseudo-random quantum circuit was successfully achieved. Quantum Supremacy would imply that a universal quantum computer has the ability to perform certain tasks exponentially faster than a classical computer<sup>8</sup>. However, it has been argued later that Google’s achievement amounted to a demonstration of a quantum advantage but not a practical advantage, in other words, the performed task was not useful for any real-life applications. Another quantum advantage breakthrough experiment has been implemented<sup>9</sup> utilising a Jiuzhang photonic quantum computer and performing Gaussian boson sampling (GBS) with 50 indistinguishable single-mode squeezed states. Here, quantum advantage has been elucidated in the sampling time complexity of a Torontonian matrix, which has exponential scaling with output photon clicks. However, this experiment demonstrates quantum advantage but fails to demonstrate quantum supremacy as this photonic quantum computer is not programmable. One of the most promising areas of research to obtain practical advantage is Quantum Machine Learning<sup>10–12</sup> which was born as a result of cross-fertilisation of ideas between Quantum Computing<sup>13,14</sup> and Classical Machine Learning<sup>15,16</sup>. QML in its spirit is similar to classical machine learning but with the main difference being that instead of classical neurons in the layers of a deep neural network, now we have qubits and quantum gates acting on qubits combined with quantum measurements playing the role of the activation function. The elegant field of

<sup>1</sup>Quantum Innovation P.c., 73100 Chania, Greece. <sup>2</sup>Alma Sistemi Srl, 00012 Guidonia, Rome, Italy. <sup>3</sup>School of Computing and Information Systems, Singapore Management University, 81 Victoria Street, Singapore 188065, Singapore. <sup>4</sup>Institute of High Performance Computing, Agency for Science, Technology, and Research (A\*STAR), 1 Fusionopolis Way, #16-16 Connexis, Singapore 138632, Singapore. <sup>5</sup>Centre for Quantum Technologies, National University of Singapore, Singapore 117543, Singapore. <sup>6</sup>Tradetec Ltd, London, UK. <sup>7</sup>These authors contributed equally: D. Aghamalyan, P. Griffin and M. Boguslavsky. ✉email: nsxetakis@yahoo.gr; davagham@gmail.com

QML has been providing a new platform for devising algorithms that exhibit quantum speedups. For instance, it has been demonstrated that such basic linear algebra subroutines as solving certain types of linear equations (the quantum version is known in the community as HHL), finding eigenvectors and eigenvalues, principal component analysis (PCA) exhibit exponential speedups compared to their classical counterparts<sup>17–21</sup>. However, in the recent findings Ref.<sup>22</sup> demonstrated that in case of PCA suggested Lloyd, Mohseni, and Rebentrost's quantum algorithm attaining the exponential speedup was simply an artifact of state preparation assumptions. Since we are dealing with a quantum system, one can utilise such quantum resources as coherence, entanglement, negativity, contextuality to leverage towards achieving practical advantage. However, it is still not completely understood what the role of different types of resources is in harnessing practical advantage from available 50 to 100 qubit noisy devices<sup>3</sup>. The three main building blocks of any QML algorithm are data encoding, unitary evolution of the system followed by the state readout performed through the measurement<sup>12</sup>. Uploading classical data in the quantum computer is not a trivial task and can account for most of the complexity of the algorithm, determining what kind of speed-ups are feasible. This procedure is called quantum embedding which can be achieved, for instance, with help of “quantum feature maps”<sup>23–27</sup> which take classical data and map it to the high-dimensional Hilbert space, where one hopes to achieve higher separation between the data classes compared to the original coordinate system. Moreover, one can train the quantum embedding to achieve maximal separation between the data clusters in the Hilbert space (this approach has been coined as “quantum metric learning”)<sup>26,27</sup>, paving the way towards constructing faithful quantum classifiers.

Binary classification is a ubiquitous task in machine learning. Perhaps the most prominent example is the cat recognition algorithm, which gives a flavour of the power brought by utilising such basic tools as logistic regression combined with deep neural network architectures<sup>15</sup>. Quantum classifiers hold a promise to bring feasible speedups compared to their classical counterparts. Several theoretical proposals combined with actual experimental runs on commercially available backends have been put forward for realising faithful quantum classifiers<sup>23,24,28–38</sup>. For instance, approaches in Refs.<sup>36,37</sup> are inspired by kernel methods used in classical machine learning. Refs.<sup>23,28,29</sup> are combining certain types of quantum embeddings to achieve quantum hybrid neural networks, which are promising candidates for building a faithful classifier. Ref.<sup>30</sup> suggests using hypergraph-states<sup>39</sup>, where the assumption is that such states can lower the circuit depth of the classifier. Refs.<sup>32,33</sup> are based on quantum Grover's search algorithm.

In this manuscript, we take a rather pragmatic approach and try to benefit from a plethora of available QML software packages<sup>40–44</sup>, which grant access to run the quantum circuit in the quantum simulator or an actual hardware (such as IBM Quantum Experience, Amazon Braket, Rigetti Computing, Strawberry Fields). By utilising these tools we provide new software that is particularly well suited for targeting classification problems in the unbalanced and noisy datasets which are prevalent in the financial industry<sup>45</sup>.

In this paper at first we briefly outline and review three different necessary building block QML architectures for our software package: hybrid-neural networks<sup>23,28,29</sup>, parametric quantum circuits<sup>2,46–48</sup> and data-reuploading<sup>24,25</sup>.

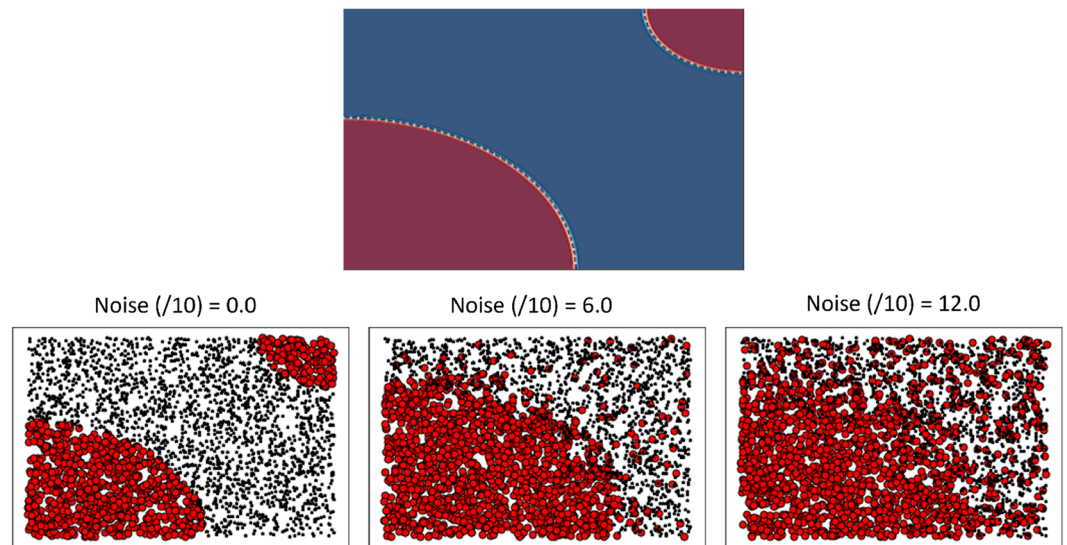
The metric we use for assessing the performance of our quantum classifiers is the area under the receiver operating characteristic curve AUC–ROC. ROC is a probability curve and AUC represents the degree of separability. In general a good model has AUC close to 1. We test our FULL HYBRID models and benchmark them against existing QML classifiers and also to the best known classical machine learning counterparts by running simulations on quantum simulators for three different 2-dimensional non-convex surfaces. It is believed that non convex boundaries represent more difficult classification problems as linear regression is bound to fail in this tasks. Then by introducing asymmetrical Gaussian noise we study the resilience of our different approaches to the noise. This kind of study sheds light on learning properties for the amount of noise in the dataset. We also perform systematic hyperparameter tuning by studying how AUC–ROC curve changes with the number of repeating units in the data-re-uploading approach, number of qubits, batch size, number of epochs and number of strongly entangling units. We remark, that our binary classifiers can be extended to multi-class classification problems using a one-versus-all approach.

## Results

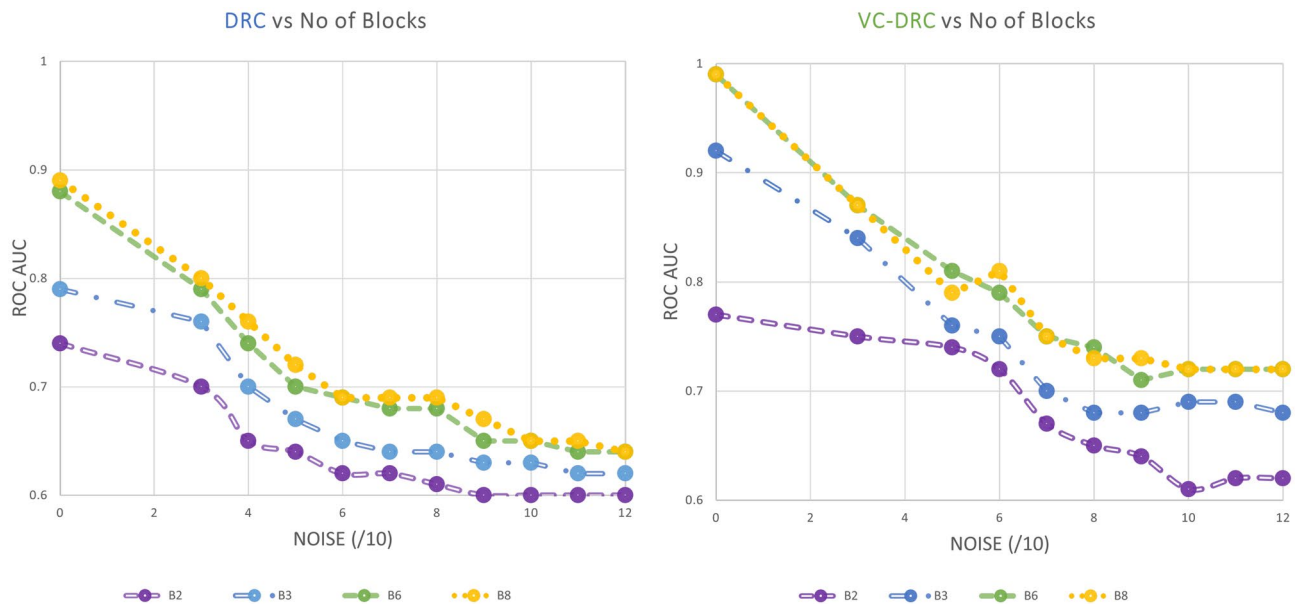
**Problem setting.** We consider a non-trivial classification problem and will train single and multi-qubit variational quantum circuits to achieve this goal. The data is generated as a set of random points in a plane  $x_1, x_2$  and labelled as 1 (blue) or 0 (red) depending on whether they lie inside or outside of a given 2-dimensional non-convex figure. The goal is to train a quantum circuit to predict the label (red or blue) given an input point's coordinate.

**Comparative study of different quantum and classical classifiers.** Here we test several models (including our proposed models) and benchmark them against each other as well as to the best-known classical machine learning counterparts by running on the simulator backends (such as Aer in qiskit) for 2-dimensional and 3-dimensional non-convex datasets. Then we will study the resilience of our different approaches to the noise by introducing asymmetrical Gaussian noise by studying the prediction grids and AUC–ROC characteristics.

This kind of study sheds light on the learning properties as a function of the amount of existing noise in the dataset. These results have been obtained by systematic hyperparameter tuning, by observing how the AUC–ROC curve changes with: the number of repeating units in the data-re-uploading approach, batch size, number of epochs and the number of strongly entangling units.



**Figure 1.** (First row) Pattern of Dataset. (Second row) Pattern of dataset for different noise levels.

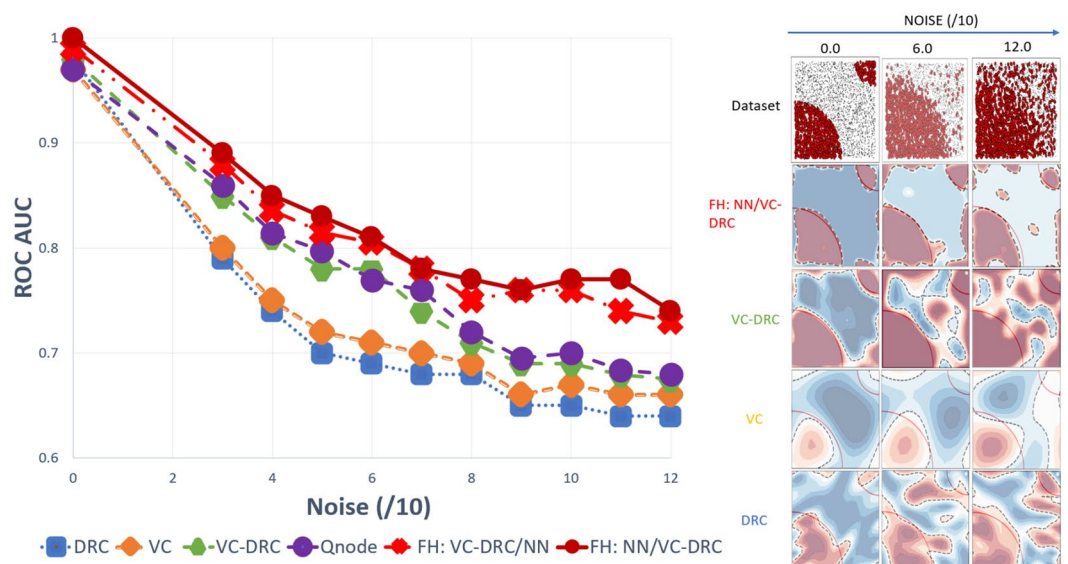


**Figure 2.** AUC–ROC as a function of number of repeating blocks of data re-uploading for the DRC classifier (left) and VC–DRC classifier (right).

To produce datasets with noise, we introduce asymmetrical (here noise is only applied to one class) Gaussian noise ( $N$ ). In bottom of Fig. 1 we plot the case of  $N = 0.0$ ,  $N = 0.6$  and  $N = 1.2$ . Each dataset has 6000 data points and is further equally split into training and testing datasets.

Here we would like to refer readers to the respective subsections of the “Methods” section, for a detailed description of different types of quantum classifiers referred as DRC (data-reuploading classifier), VC (variational classifier “Variational Quantum Algorithms (VQA)” section), VC–DRC (variational classifier combined with data-reuploading one), QNODE (quantum node, see “QNode” section) and our newly designed FULL HYBRID circuit architectures referred as FH: VC–DRC/NN and FH: NN/VC–DRC.

To demonstrate the power of the data-reuploading technique combined with the variational classifier in the VC–DRC model, we plot the AUC–ROC curve versus noise for different number of blocks. The results are shown in Fig. 2. It is apparent from Fig. 2 (left) that with an increasing number of repeating blocks, we get better AUC–ROC curve for every noise level for the DRC classifier. On the right of Fig. 2 we show results for the VC–DRC where compared to DRC we get even higher AUC–ROC curve. We remark that no major improvements are seen for a Block number greater than six. From now on, in all codes of this section, we will set the number of blocks equal to six ( $B = 6$ ). In what follows we specify number of blocks and layers for each classifier: 1) The single qubit DRC ( $B = 6$ ) 2) 2 qubit VC (with 6 layers,  $L = 6$ ) 3) VC–DRC ( $B = 6$ ,  $L = 1$ ) 4) QNode ( $B = 6$ ,  $L = 1$ ) 5)



**Figure 3.** (Left) AUC–ROC curve for DRC, VC and VC–DRC classifiers for different noise levels. (Right) Prediction grids for respective classifiers for different noise levels.

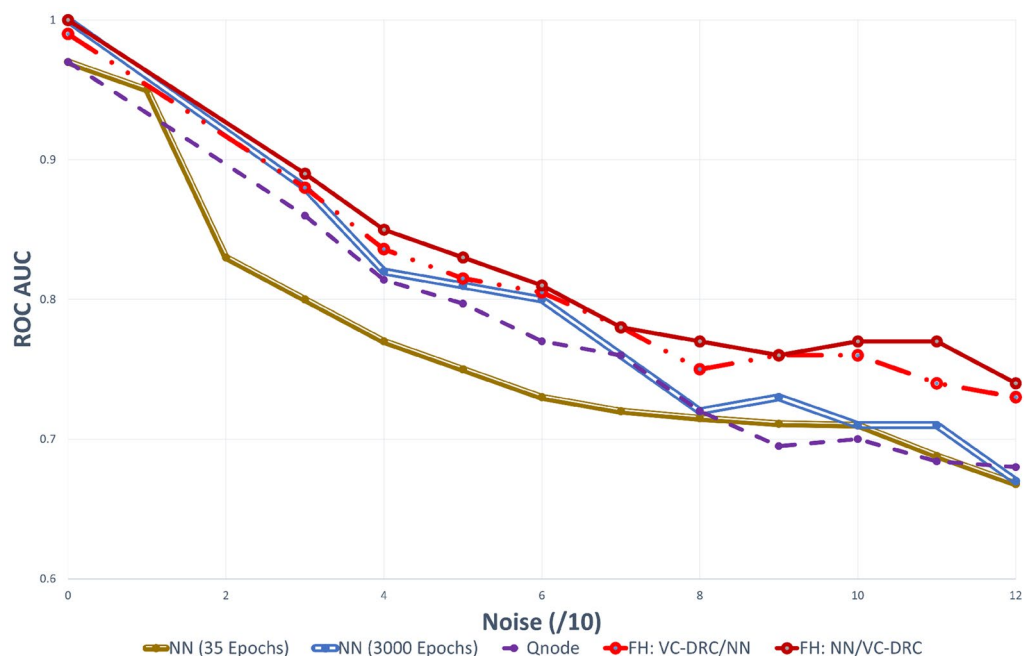
FH: VC–DRC/NN ( $B = 6, L = 1$ ) 6 FH: NN/VC–DRC ( $B = 6, L = 1$ ). All models have been trained for maximum 35 epochs, using the same optimizer and learning rate. The best result during the training process is shown. On the left Fig. 3 we compare all the previously mentioned classifiers. As we can see from on the left Fig. 3 VC–DRC outperforms both VC and DRC. VC–DRC and Qnode have almost identical performance. The FH:NN/VC–DRC outperforms all classifiers whilst FH:VC–DRC/NN has slightly worse behavior. In the right of Fig. 3 we can see the prediction grids for all classifiers at different noise levels. For low noise levels (Noise/10 = 0), DRC and VC struggle to capture the prediction grid pattern while VC–DRC and FH almost capture it. For medium noise levels (Noise/10 = 6), DRC tends to capture the noise (overfitting) while VC looks more stable. VC–DRC still captures the main pattern but also shows signs of overfitting. FH performs very well thanks to the classical preprocessing and utilising the power brought by VC–DRC. For high noise levels (Noise/10 = 12) FH captures the pattern and shows robustness to the noise while the rest of the classifiers are capturing the noise. In order to demonstrate that FULL HYBRID does not perform well only because of the strong classical NN attached to the quantum circuit, we benchmark FH versus just the classical part (NN) and versus just the Quantum part (QNode). From Fig. 4 onwards we show results for two NN's one with 35 epochs training (same training epochs as in the FH) and 3000 epochs to see what is the best outcome this NN can produce. We conclude that the FH outperforms both it's components (NN and QNode) which shows that FH is more powerful classifier than it's isolated parts.

To test even further the FH classifier, we benchmark its performance against a great number of classical counterparts, which are specified in the inset of the Fig. 5. Interestingly, this figure shows that in the high noise region, the quantum classifier appears to outperforms some classical ones, at least performing equally well in all noise regions. We also see that compared to the other classical approaches (QDA, Decision tree, KNN and Random forest) that are well suited for non-convex classification problems and showing good performance in all noise regimes. In Fig. 6 we are showing results for a more complicated non-convex classification problem versus noise. In the table on the right we summarize the highest AUC–ROC curve scores for the respective classifiers. In the left figure we show prediction grids for the respective quantum classifiers. As in previous case, VC is more stable to noise and DRC tends to overfit and explores richer prediction grids. That is why VC–DRC, which combines both features, and the more complex approach like FH, is giving great results as apparent from row number 6. Surprisingly, for this particular dataset FH: NN/VC–DRC fails to capture the pattern of the dataset while FH: VC–DRC/NN captures the pattern and has the highest AUC–ROC score. It should be noted that the FH models outperforms again both it's components (NN and QNode).

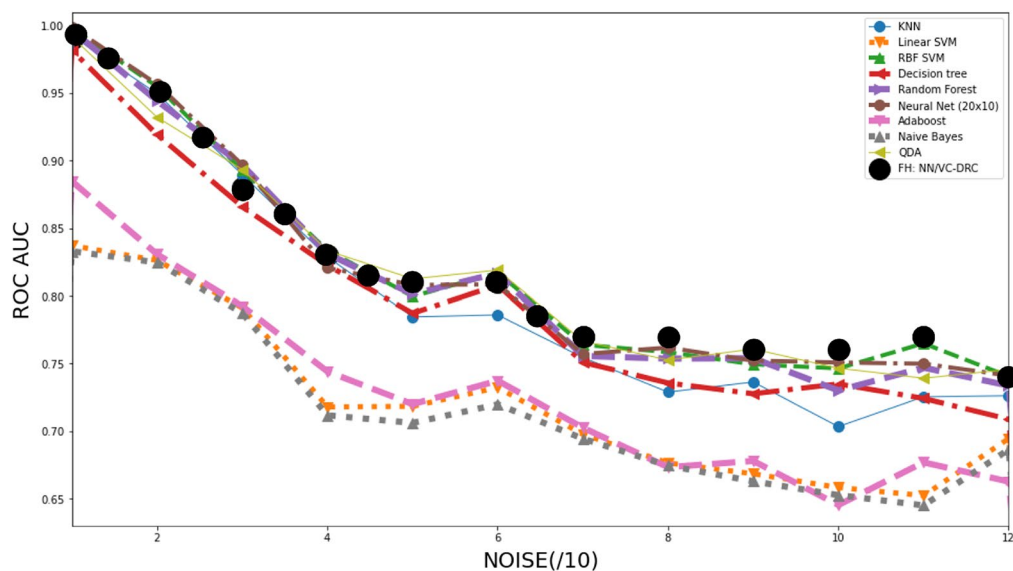
## Discussion

In this paper, we applied Quantum Machine Learning frameworks to improve binary classification models for noisy datasets which are prevalent in financial markets. The metric used for assessing the performance of our quantum classifiers is the area under the receiver operating characteristic curve AUC–ROC curve. By combining such approaches as hybrid-neural networks, parametric circuits, and data re-uploading we created a new approach called Full Hybrid (FH). We tested our models for the classification of 2 and 3-dimensional non-convex datasets and benchmarked them against each other as well as to the best known classical machine learning counterpart by running simulations on quantum simulators. Then, by introducing asymmetrical Gaussian noise in the input datasets, we studied the resilience of our different approaches to noise. This kind of study sheds light on the learning efficacy to the amount of noise in the dataset. In the scope of the manuscript we also performed systematic hyperparameter tuning by studying how AUC–ROC curve changes with the number of





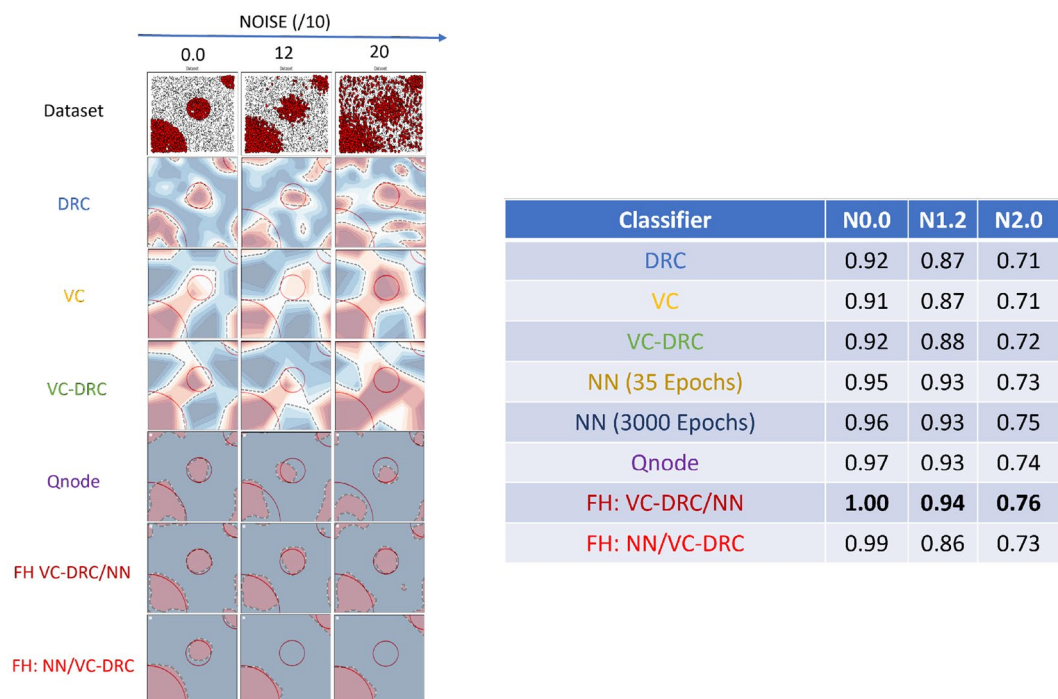
**Figure 4.** AUC–ROC curve for increasing level of noise for the classification of 2d dataset. Here we benchmark FH versus just the classical part (NN) and versus just the quantum part (Qnode).



**Figure 5.** AUC–ROC curve for increasing level of noise for the classification of 2d dataset. Here we benchmark a great number of classical classifiers against our proposed FULL HYBRID classifier.

repeating units in the data-re-uploading approach, number of qubits, batch size, number of epochs and number of strongly entangling units.

An extensive benchmarking of our new QML approach against existing quantum and classical classifier models reveals that our novel (FH) models exhibits better learning properties with asymmetric Gaussian noise in the dataset compared to known quantum classifiers, and performs equally well or possibly better for existing classical counterparts. Yet more understanding of the merits of the (FH) classifier has been gained by a detailed analysis and comparison of the prediction grids for the VC, DRC, VC–DRC, QNode binary classifiers. We observed that for low noise levels, DRC and VC struggle to capture the prediction grid pattern while VC–DRC and FH almost fully capture it. For medium noise levels, DRC tends to capture the noise (overfitting) while VC looks more stable. VC–DRC still captures the main pattern but also shows signs of overfitting. FH performs very well thanks



**Figure 6.** (Right) AUC–ROC curve for classifiers for different noise levels. (Left) Prediction grids for respective classifiers for different noise levels.

to the classical preprocessing and utilising the power brought by VC–DRC. For high noise levels, (FH) captures the pattern and shows robustness in noise while the rest of the classifiers are capturing the noise in the dataset.

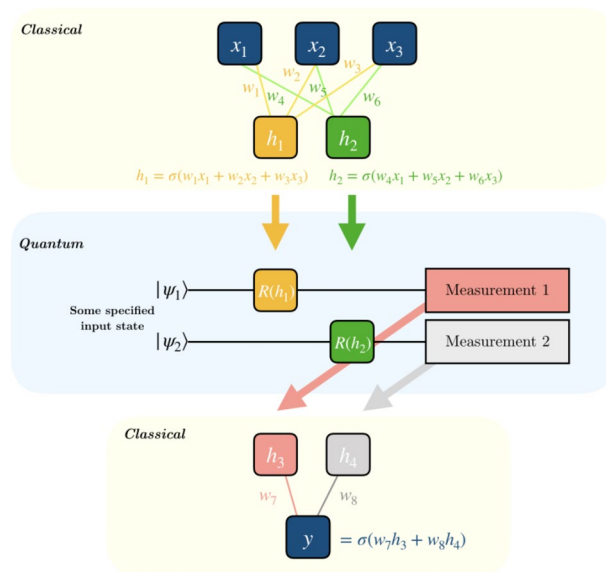
It is a well conceived fact that one of the bottlenecks for VQAs is the phenomenon called “barren plateau”<sup>49</sup>. As it has been demonstrated in Ref.<sup>49</sup>, a given spin-spin interacting Hamiltonians cost function may exhibit a barren plateau, associated with exponentially vanishing variance in its first derivative, when one increases the number of qubits. Moreover, the VQE based algorithms perform a classical–quantum feedback loop to update the parameters of the parametric quantum circuits. For future studies, it would be interesting to implement non-VQA algorithms for building more efficient quantum classifiers. By the time a classical computer calculates its output, the classical–quantum feedback loop limits the efficiency of the quantum device, slowing the algorithm execution on current cloud computing frameworks. Most of the obstacles faced by VQE, such as the barren plateau issue<sup>49</sup> as well as lacking a systematic method to select the ansatz and the innate necessity of having controlled unitaries, have been recently tackled by suggesting a quantum assisted simulator (QAS)<sup>50,51</sup>. Remarkably, The QAS algorithm does not require any classical–quantum feedback loop, can be parallelized, alleviates the barren plateau problem by prescribing a systematic approach to constructing the ansatz, and is not based on the usage of complicated unitaries.

Of course, for the future studies, one has to keep in mind that sensitivity to errors and noise in qubits and quantum gates are the two most prominent obstacles towards scalable universal quantum computers. Given that, it would be nice to study how our results are affected if one implements noise models for realistic quantum backends. In general, a noisy quantum system is described by the open system model and systems dynamics within the Born–Markov approximation is governed by the Lindblad master equation for the system’s density matrix<sup>52</sup>. Another approach to describe the different noise channels is based on Kraus operators which are the most general physical operations acting on density matrices<sup>13</sup>.

It has been elucidated that sensitivity to input errors such as adversarial robustness is a severe problem in quantum classifiers Refs.<sup>53,54</sup>. Robustness of our FULL Hybrid architecture is a topic of future investigation, however as it has been demonstrated in Ref.<sup>54</sup> practical quantum classification tasks classify a subset of encoded states with some commonly used qubit encoding scheme (Which is indeed the case as in the current article we have used angle embedding for the data encoding). For such tasks, the authors have shown that one can use the concentration of measure phenomenon to derive the robustness of any quantum classifiers in situations where the distribution of states to be classified can be smoothly generated from a Gaussian latent space.

Since most of our codes were based on PennyLane, it is instructive to mention that PennyLane has 3 different ways for implementing noise in quantum circuits: classical parametric randomness, PennyLane’s built-in default, mixed device, and plugins for other platforms. Of course, Quantum circuits may be run on a variety of backends, some of which have their own associated programming languages and simulators. PennyLane interfaces to these other languages via plugins such as for Cirq and Qiskit.

Finally, it is also worth mentioning that we plan to test our classifiers on real world financial data. Here we hope to demonstrate that our proposed classifiers have the potential to improve credit scoring accuracy. Credit



**Figure 7.** Quantum circuit implementing hybrid classical–quantum classifier, each block corresponds to the layer of classical neural network. Taken from the online source Ref.<sup>57</sup>.

scoring provides lenders and counterparties better transparency of the credit risk they are taking when dealing with a counterparty. For large companies, this transparency is provided by public credit ratings. Small and medium enterprise companies (SMEs) are not covered by rating agencies and are suffering from reduced availability of credit. These datasets, along with the best classical neural networks, will be provided by the company called Tradeteq (Tradeteq is a value-added service provider to the Networked Trading Platform (NTP) of Singapore).

In summary, we have demonstrated that the FH architecture outperforms several previously known quantum classifiers along with some of the best known classical counterparts. Interestingly, in the FH: VC–DRC/NN case, the power of the approach is given by the fact that, the VC–DRC part is acting as quantum embedding.

## Methods

**Review of existing QML frameworks.** In this section we briefly review three different necessary building block QML architectures for our software package: hybrid-neural networks<sup>23,28,29</sup>, variational circuits<sup>2,47</sup> and data-reuplodging<sup>24,25</sup>.

**Hybrid classical–quantum classifier (Hybrid).** Recent findings of the Ref.<sup>55,56</sup> on applying Hybrid quantum based memory-centric and heterogeneous multiprocessing architecture, have revealed the practical advantage of hybrid algorithms compared to standard classical algorithms in both the computational speed and quality of the solution. These findings encapsulate a strong motivation for studying hybrid classical–quantum architectures for obtaining practical advantage.

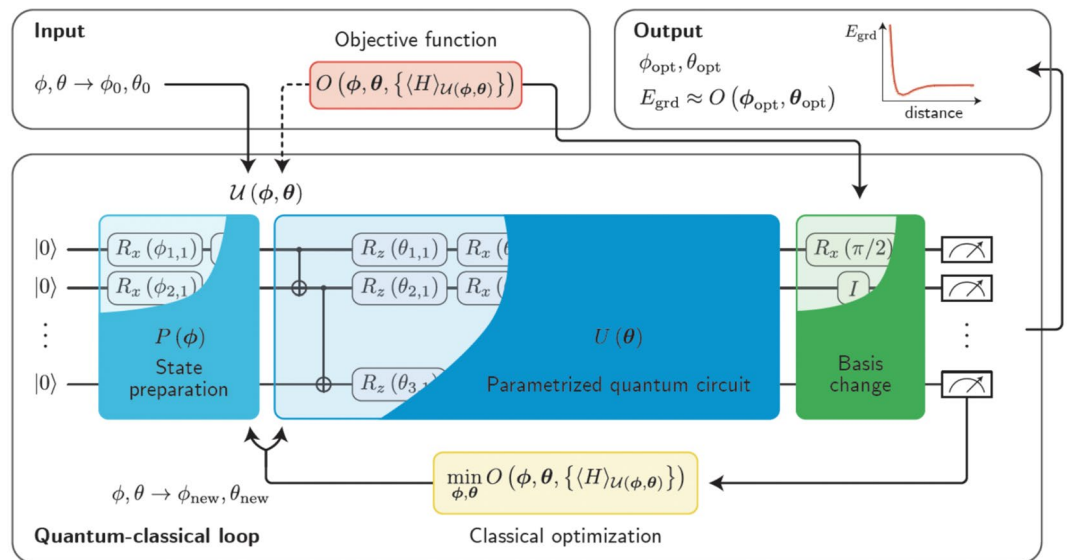
Hybrid neural networks are formed by concatenating classical and quantum neural networks and can bring a great advantage by having a number of features in the initial classical layers that exceeds the number of qubits in the quantum layer. Normally we assume that in each layer we have one qubit for each feature and a sequence of one and two-qubit gates acting on it.

To create a quantum-classical neural network, a hidden layer is normally implemented utilising a parameterized quantum circuit (Fig. 7). By “parameterized quantum circuit”, we mean a quantum circuit where, for instance, the rotation angles for each gate are trainable parameters, specified by the components of a classical input vector. The outputs from our neural network’s previous layer will be collected and used as the inputs for our parameterized circuit. Normally measurement statistics at the end of the quantum circuit would be fed into the subsequent classical neural network layer. Notice that this kind of approach establishes a link between the classical and quantum neural networks. An important point to note is that a single qubit classifier generates no entanglement, and can therefore be simulated classically. If one hopes to achieve a quantum advantage using hybrid neural networks, one needs to introduce several qubits and consequently entangle them, harnessing that quantum resource.

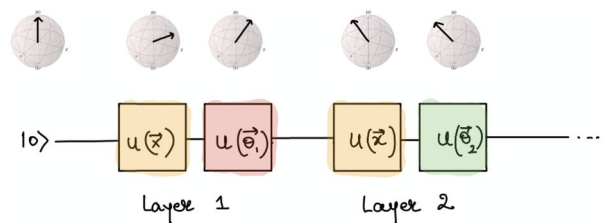
**Variational Quantum Algorithms (VQA).** Variational circuits are quantum circuits that have learning parameters that are optimised through classical learning subroutines, in spirit, this kind of approach is reminiscent of a Variational Quantum Eigensolver (VQE)<sup>2,47</sup>.

As schematically shown in Fig. 8, the first step towards developing a VQA is to define a cost or loss function  $C$  which encompasses the solution to the problem. After that, an ansatz is introduced through the quantum





**Figure 8.** Schematic diagram of a Variational Quantum Algorithm (VQA). Figure is taken from the Ref. <sup>2</sup>.



**Figure 9.** Quantum circuit implementing data re-uploading, each block corresponds to the layer of classical neural network. Image is taken from the online source Ref. <sup>58</sup>.

operation depending on a set of continuous or discrete parameters that can be optimized. This ansatz is then trained in a hybrid quantum-classical loop to solve the optimization task at hand

$$\theta^* = \arg \min_{\theta} C(\theta). \tag{1}$$

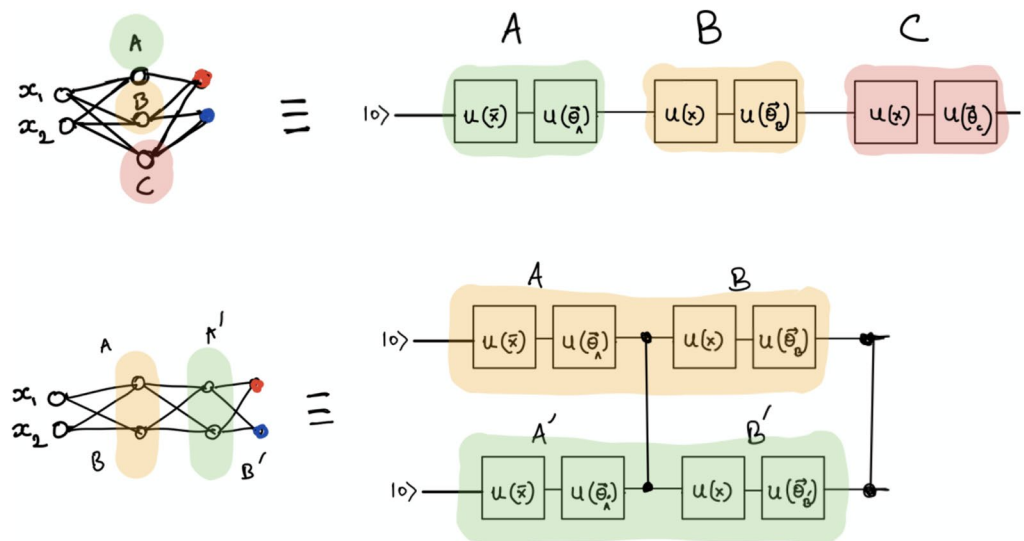
The trademark of VQAs is that a quantum computer is utilised to estimate the cost function  $C(\theta)$  while harnessing the power of classical optimizers for training the quantum parameters. A rather crucial assumption here is that one cannot efficiently compute the cost function on the classical computer, as this would imply an absence of quantum advantage in the VQA framework.

**DRC: Data-reuploading classifier.** Data re-uploading is a subclass of quantum embedding which is realised by catenating repeating units in a row. Single-qubit rotations applied several times along the circuit generate the necessary non-linearity for engineering a functional neural network. Moreover, it has been demonstrated that a single qubit can realise both being a universal quantum classifier<sup>24</sup> and being a universal approximant<sup>25</sup>.

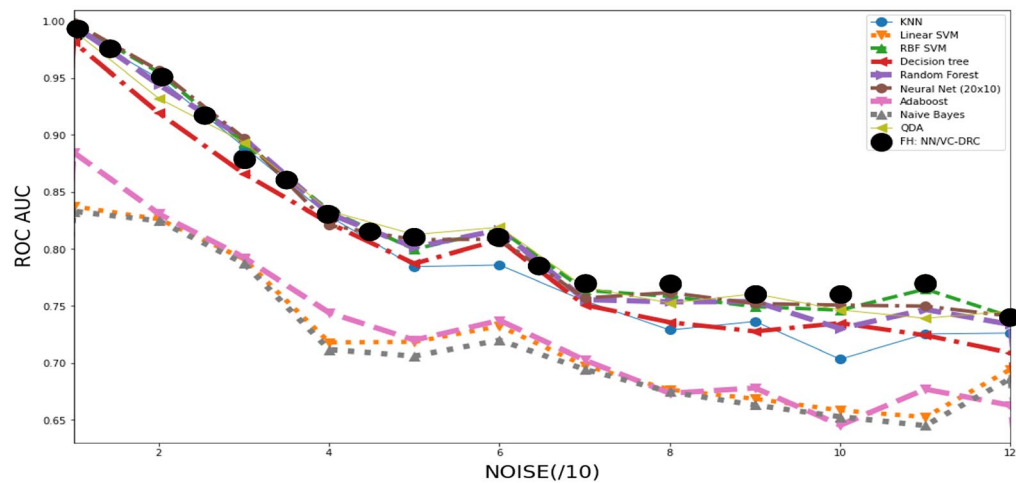
To load  $[x_1, x_2]$  into the qubit, we just start from some initial state vector,  $|0\rangle$ , apply the unitary operation  $U(x_1, x_2, 0)$  and end up at a new point on the Bloch sphere. Here we have padded 0 since our data is only 2-dimensional. Authors of Ref. <sup>24</sup> discuss how to load a higher dimensional data point  $[x_1, x_2, x_3, x_4, x_5, x_6]$  by breaking it down into sets of three parameters ( $U(x_1, x_2, x_3), U(x_4, x_5, x_6)$ ).

After the data loading stage, we want to have some trainable non-linear model analogous to a deep neural network with a non-linear activation function where one can learn the weights of the model. Fig. 9 are showing how data reuploading is implemented by the sequence of  $B$  repeating units which correspond to the layers of classical neural networks, consequently one expects that with increasing  $B$  one gets a deeper neural network and consequently better learning can be obtained. Each unit is realised as a product of two unitaries  $U(x_1, x_2, 0)$  and  $U(\theta_1, \theta_2, \theta_3)$ , where the second unitary contains the trainable parameters. This approach can be boosted by introducing strongly entangling layers through utilisation of CNOT gates as it is shown on Fig. 10.

As it has been mentioned in the previous section, one can also speculate that multiple qubits with an entanglement between them could provide some quantum advantage over classical neural networks.



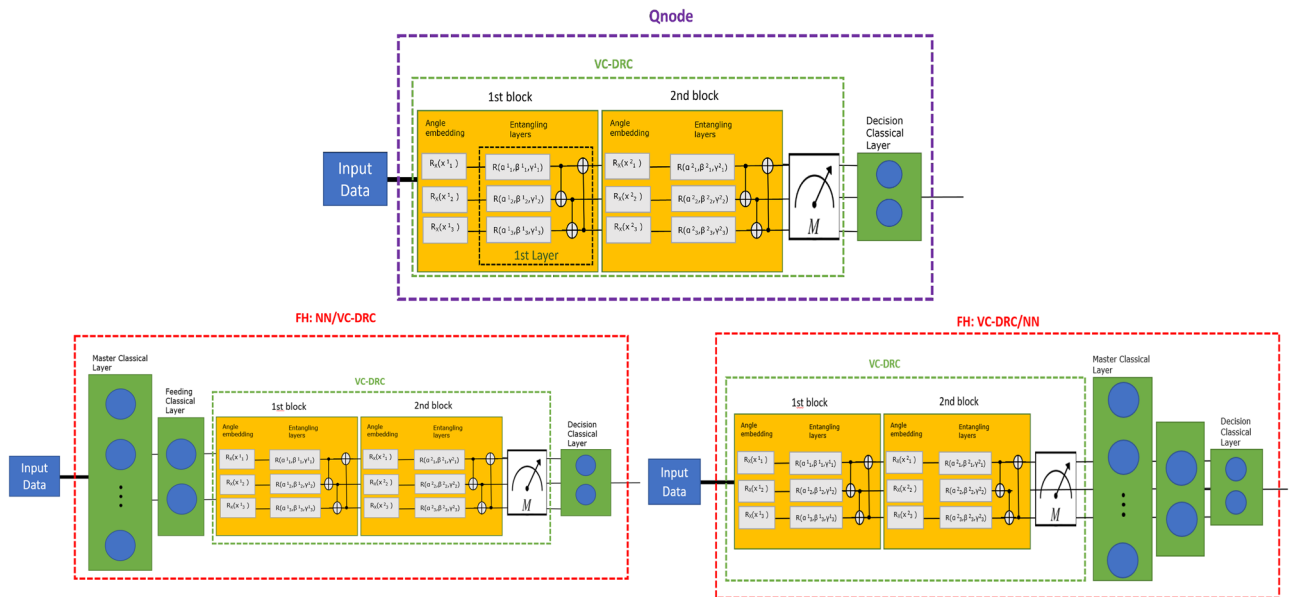
**Figure 10.** Quantum circuit implementing data-reuploading with strongly entangling layers, where entanglement between the blocks is introduced with controlled two-qubit gates. Figure is taken from the online source Ref.<sup>58</sup>.



**Figure 11.** VC–DRC example for two Blocks and one entangling layer per block ( $B = 2, L = 1$ ).

**FH:NN/VC–DRC and FH:VC–DRC/NN Full hybrid neural networks enriched with Variational and data-reuploading technics.** *VC–DRC.* A Variational classifier circuit (VC) consists of a data embedding layer which in turn loads the classical data into the qubits followed by the entangling layers (CNOT gates that entangle each qubit with its neighbour) and the measurement outcome is the expectation value of a Pauli observable for each qubit<sup>59</sup>. In our case we use an angle embedding  $R_x$ . In order to combine a VC circuit with DRC technique we define as one block (B) a sequence of data embedding and entangling layers (L). By adding many blocks we re-introduce the input data into the model. In Fig. 11 we illustrate such VC–DRC circuit for  $B = 2, L = 1$ . The trainable parameters are the Rotational gates  $R_x$  and  $R$  in the Angle embedding and Entangling layers of each block respectively.

*QNode.* PennyLane is an open-source software framework for differentiable programming of quantum computers. All our models are builded using this framework. In PennyLane an object QNode represents a quantum node in the hybrid computational graph. Here a quantum function is used to create a quantum node, or QNode object, encapsulating the quantum function (corresponding to a variational circuit) and the device used to execute the function.



**Figure 12.** (First row) Block Diagram of the specific QNode proposed in the current study. (Second row left) Block Diagram of the Full Hybrid (NN/VC–DRC) classifier where a VC–DRC circuit is placed after a classical neural network, (Second row right) Block Diagram of the Full Hybrid classifier where a fully quantum VC–DRC circuit is placed before classical neural network (VC–DRC/NN).

Here we would like to clarify what we call a QNode in the scope of the current manuscript. As depicted in the first row of the Fig. 12, a QNode is a specific circuit where input data are passed to the quantum Node which consists of a VC–DRC and a final classical decision layer.

The input classical data is passed into the quantum circuit as rotation angles  $R_x$  (“angle embedding”) on the Bloch sphere. After the computation on the quantum node is completed, measurement is performed and the outcome is passed to the classical decision layer which decides the final prediction label of the binary classifier.

*FH:NN/VC–DRC and FH:VC–DRC/NN.* In this section we propose two varieties of a new binary classifier architectures which are named under the common name Full Hybrid (FH). On a basic level, FH consists of a VC–DRC combined with classical layers. We came up with two novel architectures for the FH circuits depending on whether the VC–DRC circuit is at the end or at the beginning of the model (named FH: NN/VC–DRC and FH: VC–DRC/NN respectively). These architectures are extensively studied in the current manuscript as novel candidates for performing binary classification on noisy datasets (See second row of Fig. 12). Moreover, we demonstrate in great detail, in the next section, that FH architectures outperforms several previously known quantum classifiers and performs equally well compared to classical counterparts. We comment that, in the FH:VC–DRC/NN case the power of the approach is given by the fact that the VC–DRC part can be acting as a quantum embedding as evidenced by Refs. 26,27. In this case the goal is to derive the angle embedding for which the separation of the data labels is maximized in the Hilbert space. Mean absolute error (MAE) is the loss function we used. The loss is the mean overseen data of the absolute differences between true and predicted values. In what follows, we provide more technical details on the Full Hybrid architectures with an emphasis on explaining and giving more details on the Master, Feeding and Decision classical layers which are depicted in the second row of the Fig. 12.

For FH: NN/VD-DRC the first part is a classical neural network (NN), followed by the VC–DRC circuit and a final decision layer, which is just a single neuron layer with a sigmoid activation function. We use a classical NN that is not fine-tuned for this specific classification task. Moreover, as it can be seen on Fig. 12, the classical NN can contain an arbitrary number of layers, and each layer can contain an arbitrary number of neurons but the last layer (Feeding classical layer) should always have the same number of neurons as number of qubits. In our 2D case the classical NN, consists of a 2-neuron layer with ReLU as the activation function (Master classical layer), followed by a 2-neuron layer with a Leaky ReLU activation function (Feeding classical layer).

For FH: VC–DRC/NN the first part is a VC–DRC circuit, followed by the previously described NN network and the same final decision layer. We remark that we also tried Sigmoid, tanh and general geometric functions, and the best performing activation functions were selected.

### Data availability

All the codes used in the manuscript will be provided under the reasonable request. Moreover one of the co-authors (N. Schertakis) was invited by Pennylane to contribute in their community forum by writing a detailed tutorial which shares large chunk of our codes for FULL HYBRID architecture. To access the tutorial use the following link: <https://github.com/nsansen/Quantum-Machine-Learning/blob/main/Pennylane>.

Received: 11 March 2022; Accepted: 14 June 2022

Published online: 13 July 2022

## References

1. Preskill, J. Quantum computing in the NISQ era and beyond. *Quantum* **2**, 79 (2018).
2. Bharti, K. *et al.* Noisy intermediate-scale quantum (nisq) algorithms. arXiv preprint <http://arxiv.org/abs/2101.08448> (2021).
3. Deutsch, I. H. Harnessing the power of the second quantum revolution. *PRX Quantum* **1**, 020101 (2020).
4. Preskill, J. Fault-tolerant quantum computation. In *Introduction to Quantum Computation and Information*, 213–269 (World Scientific, 1998).
5. Gottesman, D. Theory of fault-tolerant quantum computation. *Phys. Rev. A* **57**, 127 (1998).
6. Shor, P. W. Fault-tolerant quantum computation. In *Proceedings of 37th Conference on Foundations of Computer Science*, 56–65 (IEEE, 1996).
7. Arute, F. *et al.* Quantum supremacy using a programmable superconducting processor. *Nature* **574**, 505–510 (2019).
8. Harrow, A. W. & Montanaro, A. Quantum computational supremacy. *Nature* **549**, 203–209 (2017).
9. Zhong, H.-S. *et al.* Quantum computational advantage using photons. *Science* **370**, 1460–1463 (2020).
10. Biamonte, J. *et al.* Quantum machine learning. *Nature* **549**, 195–202 (2017).
11. Wittek, P. *Quantum Machine Learning: What Quantum Computing Means to Data Mining* (Academic Press, 2014).
12. Schuld, M. *Supervised Learning with Quantum Computers* (Springer, 2018).
13. Nielsen, M. A. & Chuang, I. *Quantum Computation and Quantum Information* (2002).
14. Preskill, J. Lecture notes for physics 229: Quantum information and computation. *Calif. Inst. Technol.* **16**, 10 (1998).
15. Goodfellow, I., Bengio, Y. & Courville, A. Machine learning basics. *Deep Learn.* **1**, 98–164 (2016).
16. Jordan, M. I. & Mitchell, T. M. Machine learning: Trends, perspectives, and prospects. *Science* **349**, 255–260 (2015).
17. Harrow, A. W., Hassidim, A. & Lloyd, S. Quantum algorithm for linear systems of equations. *Phys. Rev. Lett.* **103**, 150502 (2009).
18. Huang, H.-Y., Bharti, K. & Rebentrost, P. Near-term quantum algorithms for linear systems of equations. arXiv preprint, <http://arxiv.org/abs/1909.07344> (2019).
19. Rebentrost, P., Steffens, A., Marvian, I. & Lloyd, S. Quantum singular-value decomposition of nonsparse low-rank matrices. *Phys. Rev.* **97**, 012327 (2018).
20. Lloyd, S., Mohseni, M. & Rebentrost, P. Quantum principal component analysis. *Nat. Phys.* **10**, 631–633 (2014).
21. Wiebe, N., Braun, D. & Lloyd, S. Quantum algorithm for data fitting. *Phys. Rev. Lett.* **109**, 050505 (2012).
22. Tang, E. Quantum principal component analysis only achieves an exponential speedup because of its state preparation assumptions. *Phys. Rev. Lett.* **127**, 060503 (2021).
23. Schuld, M. & Killoran, N. Quantum machine learning in feature Hilbert spaces. *Phys. Rev. Lett.* **122**, 040504 (2019).
24. Pérez-Salinas, A., Cervera-Lierta, A., Gil-Fuster, E. & Latorre, J. I. Data re-uploading for a universal quantum classifier. *Quantum* **4**, 226 (2020).
25. Pérez-Salinas, A., López-Núñez, D., García-Sáez, A., Forn-Díaz, P. & Latorre, J. I. One qubit as a universal approximant. arXiv preprint <http://arxiv.org/abs/2102.04032> (2021).
26. Lloyd, S., Schuld, M., Ijaz, A., Izaac, J. & Killoran, N. Quantum embeddings for machine learning. arXiv preprint, <http://arxiv.org/abs/2001.03622> (2020).
27. Mitarai, K., Negoro, M., Kitagawa, M. & Fujii, K. Quantum circuit learning. *Phys. Rev.* **98**, 032309 (2018).
28. Schuld, M., Bocharov, A., Svore, K. M. & Wiebe, N. Circuit-centric quantum classifiers. *Phys. Rev.* **101**, 032308 (2020).
29. Farhi, E. & Neven, H. Classification with quantum neural networks on near term processors. arXiv preprint, <http://arxiv.org/abs/1802.06002> (2018).
30. Tacchino, F., Macchiavello, C., Gerace, D. & Bajoni, D. An artificial neuron implemented on an actual quantum processor. *NPJ Quantum Inf.* **5**, 1–8 (2019).
31. Cappelletti, W., Erbanni, R. & Keller, J. Polyadic quantum classifier. In *2020 IEEE International Conference on Quantum Computing and Engineering (QCE)*, 22–29 (IEEE, 2020).
32. Wiebe, N., Kapoor, A. & Svore, K. M. Quantum perceptron models. arXiv preprint, <http://arxiv.org/abs/1602.04799> (2016).
33. Liao, Y., Ebler, D., Liu, F. & Dahlsten, O. Quantum advantage in training binary neural networks. arXiv preprint, <http://arxiv.org/abs/1810.12948> (2018).
34. Schuld, M., Fingerhuth, M. & Petruccione, F. Implementing a distance-based classifier with a quantum interference circuit. *EPL (Europhys. Lett.)* **119**, 60002 (2017).
35. Tiwari, P. & Melucci, M. Towards a quantum-inspired binary classifier. *IEEE Access* **7**, 42354–42372 (2019).
36. Blank, C., Park, D. K., Rhee, J.-K. & Petruccione, F. Quantum classifier with tailored quantum kernel. *NPJ Quantum Inf.* **6**, 1–7 (2020).
37. Park, D. K., Blank, C. & Petruccione, F. The theory of the quantum kernel-based binary classifier. *Phys. Lett. A* **384**, 126422 (2020).
38. Huggins, W., Patil, P., Mitchell, B., Whaley, K. B. & Stoudenmire, E. M. Towards quantum machine learning with tensor networks. *Quantum Sci. Technol.* **4**, 024001 (2019).
39. Rossi, M., Huber, M., Bruß, D. & Macchiavello, C. Quantum hypergraph states. *New J. Phys.* **15**, 113022 (2013).
40. Bergholm, V. *et al.* Pennylane: Automatic differentiation of hybrid quantum-classical computations. arXiv preprint, <http://arxiv.org/abs/1811.04968> (2018).
41. Killoran, N. *et al.* Strawberry fields: A software platform for photonic quantum computing. *Quantum* **3**, 129 (2019).
42. Broughton, M. *et al.* Tensorflow quantum: A software framework for quantum machine learning. arXiv preprint, <http://arxiv.org/abs/2003.02989> (2020).
43. Efthymiou, S. *et al.* Qibo: a framework for quantum simulation with hardware acceleration. arXiv preprint, <http://arxiv.org/abs/2009.01845> (2020).
44. Kottmann, J. *et al.* Tequila: A platform for rapid development of quantum algorithms. *Quantum Sci. Technol.* **6**(2), 024009 (2021).
45. Orus, R., Mugel, S. & Lizaso, E. Quantum computing for finance: Overview and prospects. *Rev. Phys.* **4**, 100028 (2019).
46. Benedetti, M., Lloyd, E., Sack, S. & Fiorentini, M. Parameterized quantum circuits as machine learning models. *Quantum Sci. Technol.* **4**, 043001 (2019).
47. Cerezo, M. *et al.* Variational quantum algorithms. arXiv preprint, <http://arxiv.org/abs/2012.09265> (2020).
48. Funcke, L., Hartung, T., Jansen, K., Kühn, S. & Stornati, P. Dimensional expressivity analysis of parametric quantum circuits. *Quantum* **5**, 422 (2021).
49. McClean, J. R., Boixo, S., Smelyanskiy, V. N., Babbush, R. & Neven, H. Barren plateaus in quantum neural network training landscapes. *Nat. Commun.* **9**, 1–6 (2018).
50. Haug, T. & Bharti, K. Generalized quantum assisted simulator. arXiv preprint, <http://arxiv.org/abs/2011.14737> (2020).
51. Bharti, K. Quantum assisted eigensolver. arXiv preprint, <http://arxiv.org/abs/2009.11001> (2020).
52. Carmichael, H. Master equations and sources i. *An Open Systems Approach to Quantum Optics: Lectures Presented at the Université Libre de Bruxelles October 28 to November 4, 1991* 5–21 (1993).
53. Liu, N. & Wittek, P. Vulnerability of quantum classification to adversarial perturbations. *Phys. Rev.* **101**, 062331 (2020).

54. Liao, H., Convy, I., Huggins, W. J. & Whaley, K. B. Robust in practice: Adversarial attacks on quantum machine learning. *Phys. Rev.* **103**, 042427 (2021).
55. Sagingalieva, A. *et al.* Hyperparameter optimization of hybrid quantum neural networks for car classification. arXiv preprint, <http://arxiv.org/abs/2205.04878> (2022).
56. Perelshtein, M. *et al.* Practical application-specific advantage through hybrid quantum computing. arXiv preprint, <http://arxiv.org/abs/2205.04858> (2022).
57. Hybrid quantum-classical neural networks with pytorch and qiskit. <https://qiskit.org/textbook/ch-machine-learning/machine-learning-qiskit-pytorch.html> (2020).
58. Ahmed, S. Data-reuploading classifier. <https://pennylane.ai/qml/demos/tutorial-data-reuploading-classifier.html> (2021).
59. Variational classifier. <https://pennylane.ai/qml/demos/tutorial/variational/classifier.html> (2021).

## Acknowledgements

All the codes used in the manuscript will be provided under the reasonable request. We would like to thank Tobias Haug for critical reading of our manuscript and for making several useful comments. We would also like to thank Kishor Bharthi for supporting our work from its early stages. The research project was supported by the Artificial Intelligence and Data Analytics (AIDA) scheme, which provides funding support to strengthen the AIDA ecosystem in the Singapore financial sector. The AIDA scheme is part of the Financial Sector Technology and Innovation (FSTI) scheme under the Financial Sector Development Fund administered by the Monetary Authority of Singapore (MAS). Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of MAS. D. A. would acknowledge the funding support from Agency for Science, Technology and Research (#21709).

## Author contributions

N.S. conducted the numerical experiment, D.A, P.G. and M.B. analysed the results. All authors reviewed the manuscript and discussed the results. D.A. drafted the manuscript.

## Competing interests

The authors declare no competing interests.

## Additional information

**Supplementary Information** The online version contains supplementary material available at <https://doi.org/10.1038/s41598-022-14876-6>.

**Correspondence** and requests for materials should be addressed to N.S. or D.A.

**Reprints and permissions information** is available at [www.nature.com/reprints](http://www.nature.com/reprints).

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2022