Working Paper: July 2022

SINGAPORE MANAGEMENT UNIVERSITY

Quantum Machine Learning for Credit Scoring

Nikolaos Schetakis
ALMA Sistemi Srl

Davit Aghamalyan
A*STAR

Michael Boguslavsky
Tradeteq

Agnieszka Rees
Tradeteq

Marc Rakotomalala
Sim Kee Boon Institute

Paul Robert Griffin
Singapore Management University

SIM KEE BOON INSTITUTE FOR FINANCIAL ECONOMICS

LEE KONG CHIAN SCHOOL OF BUSINESS

Quantum Machine Learning for Credit Scoring

N. Schetakis[3,6], D. Aghamalyan[1,5], M. Boguslavsky[2], A. Rees[2], M. Rakotomalala[4], P. Griffin[1]

1. School of Computing and Information Systems, Singapore Management University, 81 Victoria Street, Singapore 188065
2. Tradeteq Ltd, London, UK
3. ALMA Sistemi Srl, Guidonia (Rome), 00012, Italy
4. Sim Kee Boon Institute for Financial Economics, Singapore Management University, 50 Stamford Road, Singapore 178899
5. Agency for Science, Technology and Research (A*STAR), Materials Science and Chemistry, Institute of High Performance Computing, 1 Fusionopolis Way, 16-16 Connexis, Singapore 138632
6. Quantum innovation, PC, Chania, Greece

Keywords: Quantum computing, machine learning, credit scoring, finance, QML, ML, fintech

ABSTRACT

In this paper we explore the use of quantum machine learning (QML) applied to credit scoring for small and medium size businesses (SMEs). A quantum/classical hybrid approach has been used for two years of experimentation with several models, activation functions, epochs, other parameters. Results are shown from the best model, using two quantum classifiers and a classical neural network, applied to data for companies in Singapore. We observe significantly more efficient training for the quantum models over the classical models for comparable prediction performance. Practical issues are also explored including a quadratic computational slow down with the number of qubits and a linear slow down with the number of blocks of classifiers using classical simulators. Running the models on real quantum computers is discussed including the number of times a circuit has to be executed. Surprisingly, a degradation in the accuracy was observed as the number of qubits was increased beyond 12 qubits and also with the addition of extra classifier blocks in the quantum model. Overall, we see great promise in this first in-depth exploration of the use of hybrid QML in credit scoring.

1. INTRODUCTION

The main concern in credit scoring is the risk of defaults, i.e. the risk that a party does not meet the obligations in the terms of a contract or an agreement. Over the past 50 years, credit risk measurement has constantly evolved from lenders relying on bankers' analysis to assess the creditworthiness of potential borrowers to moving towards more objective credit scoring systems based on "hard" accounting data, for example, the Z-score model [1]. Recent models rely on capital market data to pick up fast-moving changes in borrower conditions [2] and nonlinear discriminant analysis and artificial neural networks, capable of fitting complex data samples. The extraordinary increase in affordable computing power, the creation of innovative algorithms, and the vast amount of data produced daily have spurred further development of sophisticated machine learning (ML) models. This, along with free open-source libraries such as TensorFlow, now allow complex pattern recognition problems in large datasets to be tackled. One industry leader in credit rating for SME's is Tradeteq, that uses sophisticated graph ML algorithms running on cloud GPU farms to calibrate and optimize these models.

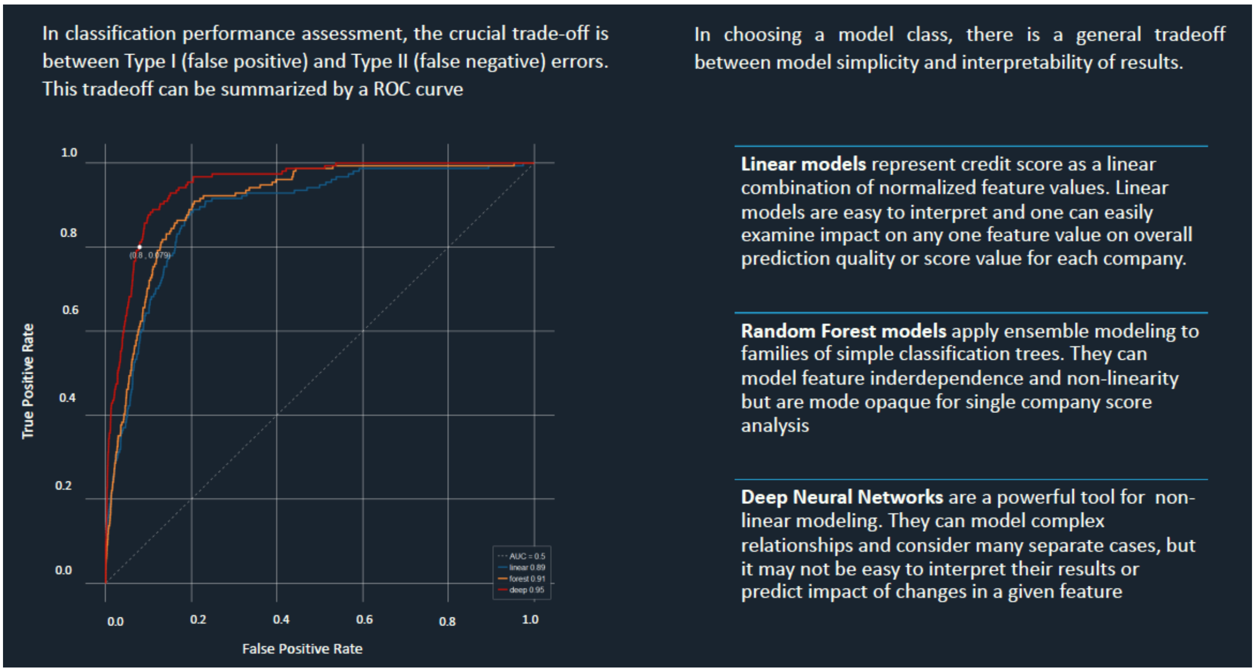
Quantum computers are now expected to disrupt numerous industry sectors, and finance in particular. A classical computer encodes and processes information in either of two states (bits), 0 and 1s. However, a quantum computer, based on quantum mechanics, exploits the fact that physical quantum particles can be in any combination of two states - quantum bits called qubits. This so-called superposition enables much larger computational capabilities[1]. Another critical property of quantum states is that qubits can become entangled — actions on one qubit impacting the other, even if they are separated in space [3]. Superposition and entanglement are the main quantum phenomena responsible for the theoretical computational advantage associated with quantum computing — without superposition, a qubit would behave like a bit; and without entanglement, useful calculations would not take place. However, quantum computing is still in its infancy. The technology is complex, and the hardware has to engineer quantum particles such as atomic nuclei, electrons, or photons. Hardware and software needed to handle the large and complex problems does not yet exist and the focus is to develop new algorithms to make current limited quantum computers applicable to real world problems. However, quantum computing hardware and software is rapidly improving and the major quantum hardware providers have published roadmaps, such as from IBM where they expect to increase the number of qubits from the current 127 to over 4,158 by 2025 whilst also improving software such as by introducing serverless capabilities [4].

This paper is organized as follows. After this introduction we explain the theory behind the classical models used for credit scoring, describe the quantum models that have been used and make some remarks on benchmarking. Following that section we describe the dataset, the experimental setup and the results. Finally, there is a discussion, conclusion, and future work.

2. THEORY

In this section we review current classical and quantum models used for credit scoring and set the groundwork for our own novel hybrid classical/quantum model. Firstly, to assess and benchmark the models in credit scoring, the crucial trade-off is between Type I (false positive) and Type II (false negative) errors. This trade-off can be summarized by a ROC (Receiver Operating Characteristics) curve (Fig. 1). Where a false positive is rejecting a company for a loan that it would pay back, and true positive is rejecting a loan where the company would have defaulted. Ideally, a model would not reject a good company (0.0 on the x axis on the graph in fig. 1) and would reject all companies that would default (1.0 on the y axis on the graph in fig. 1).

[1] N qubits can express the same amount of information as 2^N classical bits — a system of 16 qubits can be in the same amount of states as 65,536 bits; a system of 300 qubits has more states than the 10^{82} atoms in the observable universe, an amount of information out of reach from classical computers.



In classification performance assessment, the crucial trade-off is between Type I (false positive) and Type II (false negative) errors. This tradeoff can be summarized by a ROC curve

In choosing a model class, there is a general tradeoff between model simplicity and interpretability of results.

Linear models represent credit score as a linear combination of normalized feature values. Linear models are easy to interpret and one can easily examine impact on any one feature value on overall prediction quality or score value for each company.

Random Forest models apply ensemble modeling to families of simple classification trees. They can model feature inderdependence and non-linearity but are mode opaque for single company score analysis

Deep Neural Networks are a powerful tool for non-linear modeling. They can model complex relationships and consider many separate cases, but it may not be easy to interpret their results or predict impact of changes in a given feature

Figure 1. The ROC curves of three models used for credit scoring. (Source: Tradeteq Ltd.)

The AUC is calculated by summing the area under each ROC curve in Figure 1 and normalising the areas to 1.0. This is a useful criterion to compare models and is the performance metric used throughout the paper.

2.1 Best classical models for credit scoring

Here we briefly outline classical ML attempts towards credit scoring where we focus on the methods used and on the accuracy of prediction of a company defaulting on a loan. As a machine learning problem, credit scoring is typically formulated as a binary classification problem with highly imbalanced data (most companies do not default). The data used has a number of features for a set of companies observed at a specific time, T0, and an outcome observed at a later time, T1. The outcome is usually treated as binary highlighting, class “1” if a company had some kind of adverse event between T0 and T1, such as companies in bankruptcy proceedings or in administration, and class “0” for all other companies in the dataset. As most companies do not suffer adverse events in most periods, the data is strongly imbalanced to class “0”.

A review of all the studies on machine learning for credit scoring is a subject large enough for a standalone paper. Indeed, we refer the reader to systematic academic reviews [5]. The first models used relatively small company sets (hundreds of companies) and just a few manually constructed accounting features; thus linear techniques such as linear discriminant analysis and support vector machines were used. Later, researchers obtained larger datasets with more companies and features, and applied linear regressions, decision trees, fuzzy logic, ensemble models, and neural networks to the problem. With large enough datasets, modern ensemble techniques, such as boosted trees, and neural networks perform broadly on par. However, boosted trees are often preferred in practice due to their better explainability and stability.

Choosing a classification model for credit scoring is a challenging task, and conflicts often arise when comparing performance. For example, linear discriminant models for predicting bad loans are found to perform better than neural networks for some data and opposite for other. These different outcomes are difficult to assess but possible explanations include differences in sample sizes, transformation functions applied to the data, model parameters, or network topology. Also, the chosen performance metric matters; traditional statistical methods seem to perform as well as neural networks if one considers the total percentage of correct identification but, if identifying bad loans is the main goal, then neural networks have been found to perform better[6]. The search for an effective "generic universal" model may be elusive but failure to incorporate situational data and consider local economic circumstances affects the ability to develop relevant powerful predictive models. In that regard, in this paper, we consider the best classical Singapore country model for classification of two groups of potential small and medium enterprise (SME) borrowers (called "best classical model" in the rest of the paper). The best classical model is described in the experiment section below.

The performance metric throughout the paper is the ROC/AUC (see benchmarking section below) and this model achieves a score of 0.73 (see results section). XGboost has been found to be the best model for this dataset but as we want to compare neural networks this is not used in this paper.

2.2 Quantum models

In the book chapter co-authored by M. Boguslavsky, P. Griffin et al. [7], the authors introduce a new framework for addressing business problems with quantum computing, assessing classes of problems which could benefit and show a use case for quantum machine learning (QML) algorithms. The authors outline two frameworks for quantum neural networks: (i) a 2-qubit perceptron inspired by the Entropica Labs algorithm for classification of cancerous cells and (ii) a hybrid neural networks where it is suggested to establish an interface between classical and quantum neural networks using PYTORCH and Qiskit.

In finance there are extensive overviews/reviews for quantum computing and QML applied to finance [8]. In all these overviews, credit scoring is mentioned as a problem which the current community is targeting to solve by making use of QML algorithms.

The Quantum Neural Network model used

Now we would like to describe our own novel architecture used for the experiments in this paper. It is a FULL HYBRID (FH) quantum neural network model consisting of three different approaches based on: hybrid-neural networks [9] , variational circuits (VC) [10] and data-reuploading classifier (DRC) [11].

First, classical data has to be encoded into quantum states. Angle embedding is commonly used to load the data [x1, x2] into a qubit. Starting from an initial state vector, typically $|0\rangle$, a unitary operation U(x1, x2, 0) is applied and a new quantum state is formed which can be described by a new point on a Bloch sphere. Padding with 0 is required when dealing with 2 or more dimensional data, for example, loading higher dimensional data [x1, x2, x3, x4, x5, x6] can be broken down into sets of three parameters: U(x1, x2, x3), U(x4, x5, x6). We use a Rx gate for angle embedding in our experiments.

Hybrid neural network classical-quantum classifiers are formed by connecting in series a number of classical and quantum neural networks. This architecture takes advantage of the specific capabilities of both types of neural networks and also benefits from being able to have the number of features in the initial classical layers exceed the number of qubits in the quantum layer, instead of being limited to one qubit for each feature.

To create our hybrid classical-quantum neural network a hidden layer is implemented utilising a variational quantum circuit (Fig. 2). A variational classifier (VC) is a quantum circuit (also called a "parameterized" circuit) consisting of the data embedding layer followed by parameterised gates such as rotation gates and entangling layers (CNOT gates that entangle each qubit with its neighbour). The quantum properties such as the rotation angles for the quantum gates are trainable parameters.

DRC is introduced by replicating the VC into more blocks (for example 2nd block in Fig. 2). To combine a VC circuit with the DRC technique we define a block (B) as a sequence of data embedding and entangling layers (L). By adding many blocks we re-introduce the input data into the model.

Our novel approach is to use the DRC technique combined with a VC in a single model in the quantum part of the hybrid classifier. This novel combination is expected to provide greater robustness to our results after making observations on 2d and 3d synthetic non-convex datasets [12]. The advantage brought by the VC approach for binary classification is in increased robustness against noise, however it struggles to capture the complicated patterns in the prediction grid diagrams. Conversely, the DRC approach has good abilities to capture complex grid structures but more sensitive to the noise in data. Consequently, it is expected that VC and DRC combined together will complement each other to get the best results. The power of the model lies in capturing the complex patterns in the data with robustness to data noise. We previously demonstrated [12] that, for synthetic datasets, FH architectures: (i) outperform several previously known quantum classifiers, (ii) perform equally well compared to classical counterparts and (iii) have an improvement over classical counterparts in regions of high noise in the dataset.

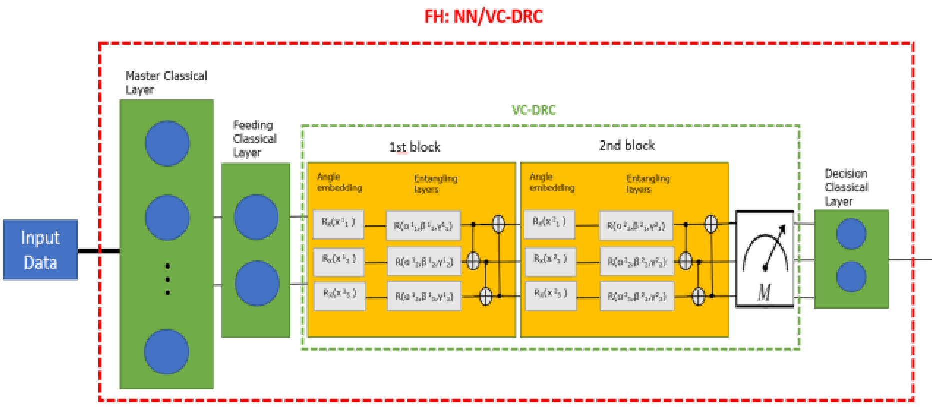


Figure 2. Block Diagram of the Full Hybrid (NN/VC-DRC) classifier where a VC-DRC circuit is placed after a classical neural network.

After computation on the quantum node is completed, measurement is performed. The measurement outcome is the expectation value of a Pauli observable for each qubit. The measurements are passed to the classical decision layer which makes the final prediction label of the binary classifier.

Another architecture explored for this problem area is to append the master classical layer to the quantum layer instead of having it first (i.e. FH:VC-DRC/NN). However, as the results for the dataset under investigation in this paper were found to be better with the FH:NN/VC-DRC model, we only focus on this model and going forward call this the FH model.

3. DATA DESCRIPTION

In this paper, we compare the performance of quantum machine learning classification algorithms with their classical counterparts, applied to a real-life credit scoring dataset. The dataset originates from various Singapore institutions — the Accounting & Corporate Regulatory Authority (ACRA), a statutory board under the Ministry of Finance of the Government of Singapore, the Singapore Land Authority, a statutory board under the Ministry of Law of the Government of Singapore, the Singapore Buildings database, Handshakes, a corporate data provider, and Tradeteq, a provider of data, technology, and software to the trade finance industry.

The dataset covers nearly 2,300 small and medium size enterprises (SMEs) firms incorporated over the 1940 – 2016 period, active and healthy in 2016, with 94% of the firms incorporated since 1990 and distributed mainly across seven industry sectors. The most important peculiarities of SME datasets are that firms are privately held and that there is limited information about the financial situation of borrowers — only accounting data is available, and no information from rating agencies nor financial markets prices. These data limitations restrict the modelling choices of an SME portfolio to binary default or no default models and panel data analysis, rather than time series analysis.

For each small business, the panel of 24 primary features, includes the year of incorporation, accounting and operating information, geo-sociological data, and an indicator of whether the company defaulted or not. The firm is statutorily deemed “Healthy” (class 0) on October 1, 2016 and its status, in case of default (e.g. compulsory winding down, receivership, or under judicial management), is statutorily changed to “Unhealthy” (class 1) over the following two years, between October 1, 2016 and October 1, 2018.

The dataset size is limited by the number of class 1 examples, i.e. the companies that defaulted. We sampled all 246 class 1 companies in the period and added to it a random subsample of 2000 class 0 companies. Therefore, the dataset is highly imbalanced. For the experiments, due to the limitations of the computing hardware, the maximum number of features used is 21.

4. EXPERIMENTS

In this section we discuss the experiments performed on the Singapore data over a period of two years of simulations trying out many different models, activation functions, epochs, and other parameters. Here we discuss the best quantum and classical counterpart (CC) models found, the hardware and software used, the hyperparameter configurations, model executions and the processing of the experiment outputs.

We used an FH:NN/VC-DRC model (Fig. 2) where the first part is a classical neural network, followed by the VC-DRC circuit and a final decision layer, a single neuron layer with a sigmoid activation function. The first classical layer «Master Classical layer» has 21 neurons, equal to the number of features used in our dataset. The second classical layer «Feeding Classical Layer» has the same number of neurons as the number of qubits. Moreover, the classical neural network can contain an arbitrary number of layers, and each layer can contain an arbitrary number of neurons but the last layer (Feeding classical layer) must have the same number of neurons as number of qubits. In our 2D case the classical NN, consists of a 2-neuron layer with a rectified linear activation function (ReLU) (Master classical layer), followed by a 2-neuron layer with a Leaky ReLU activation function (Feeding classical layer). Finally, a classical decision layer «Decision Classical layer» makes the prediction. All layers have a ReLU activation function except for the final decision layer with a sigmoid activation function. After each classical layer, to avoid overfitting, we added a dropout layer to both quantum and CC models with 10% dropout rate.

Before the data is ingested into the model, it is pre-processing using a standard pipeline along with some proprietary processing. This is the same processing that is used in the best XGboost classical models.

Data encoding into quantum states used angle embedding (see 2.THEORY). As simulators were used throughout, there is no quantum computer noise and no additional noise was introduced in these experiments. See the discussion section for considerations on how noise may affect the results on real quantum computers.

Pennylane, an open-source software framework for differentiable programming of quantum computers, was used to build the models. The computer used for these experiments was a PC with 64GB of RAM and an AMD Ryzen 7 processor.

We tested the quantum model performance versus the number of qubits and number of blocks and against the (CC) model. The training dataset has 1798 rows of data split into a validation set of 180 rows (10%) and a test set of 270 rows (15%). Throughout the simulations, the same training, validation, and testing dataset is used. For every configuration, we used the average outcome of ROC/AUC score of 5 simulations.

The hyperparameters used are summarised in table 1 and were kept consistent throughout the study except where explicitly mentioned.

Epochs	350	Number of complete passes through the training dataset

Dropout rate	0.1	Probability of training a given node in a layer, 1.0 = no dropout, 0.0 = no outputs from the layer
Learning rate	0.001	Step size at each iteration while moving toward a minimum of a loss function
Optimizer	SGD	Stochastic Gradient Descent
Batch size	16	Number of training samples to work through before the model's internal parameters are updated

Table 1. The study hyperparameters.

For a fair comparison, the number of epochs for quantum and classical models were kept identical at 350. This number was chosen after observing that, on average, after 350 epochs, overfitting was observed. However, the classical model improved if training was increased up to 5,000 epochs.

The number of DRC blocks was increased from 1 to 10 and the number of qubits was increased from 6 to 18. The batch size was reduced to 16 training examples per iteration due to memory restrictions. The optimizer we used is a Stochastic Gradient Descent (SGD) optimizer. The loss function is binary cross-entropy and the optimising parameter is ROC/AUC as described in 2. THEORY.

Throughout the simulations, we used the same training, validation, and testing dataset for consistency. For every configuration specified, we used the average outcome of the ROC/AUC score over 5 simulations. Our simulations were restricted to a maximum number of 18 qubits due to execution time constraints (see scaling results in 5. RESULTS and 6. DISCUSSION).

5. RESULTS

We now show the experimental results comparing the overall ROC/AUC of the FH model to the CC model and show the quantum model's behaviour as we scale up the number of qubits and the number of processing blocks.

Comparison of quantum to classical

The quantum model achieved an ROC/AUC of 0.75 (Fig. 3 black dots and Fig. 5 right column) whereas the CC model achieved an ROC/AUC of 0.73 (Fig. 3 orange line). However, letting the classical counterpart model train up to 5,000 epochs can lead to the highest score of 0.75 (Fig. 3 red line). This indicates that the FH model has the ability to achieve a higher score than its classical counterpart with fewer training epochs.

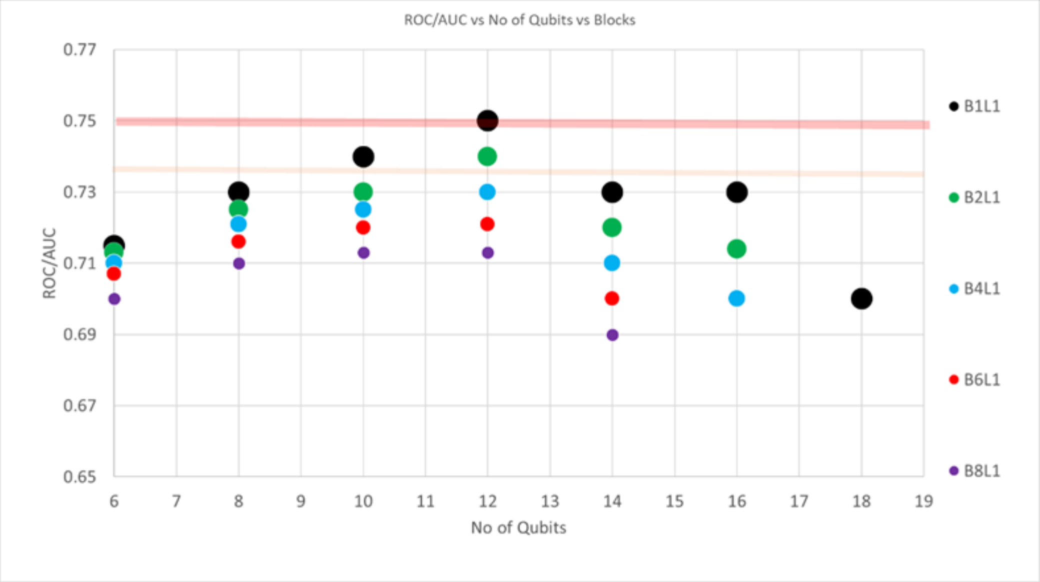


Figure 3: The FH model versus the number of qubits (x-axis) and number of blocks (coloured circles). The orange line denotes the best classical model ROC/AUC when trained for a maximum of 350 Epochs while the red line denotes the best classical model's ROC/AUC when trained for a maximum of 3,500 epochs.

Furthermore, in Figure 3, we see how the ROC/AUC of FH model performance changes with the number of qubits and number of blocks. For blocks=1 (black dots) we observe that ROC/AUC increases up to 12 qubits and then decreases. The same behaviour can also be observed when one increases the number of blocks with the highest results being achieved with no data-reupload. See the discussion section below for possible explanations.

The training processes that produce the highest ROC/AUC scores on the testing dataset are depicted in Figure 4 for both CC model (left and middle columns) and the FH model (right column). The top row depicts the final ROC/AUC, the middle and bottom rows respectively show the loss and ROC/AUC training process for the training and validation dataset. The FH model achieved the highest score with fewer training epochs for all simulations shown in Figure 4.

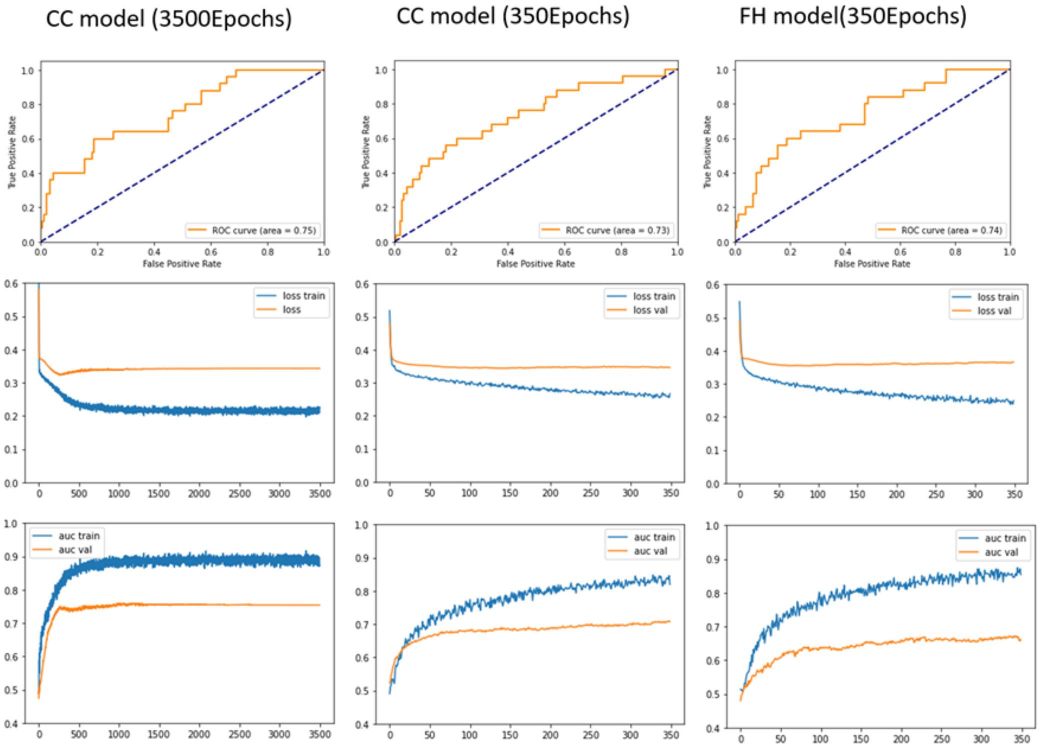


Figure 4: Results of both CC model (left and middle columns) and FH model (right column). Results for CC model are shown for different max number of training epochs 3,500 epochs (left column) and 350 epochs (middle column).

FH model execution times

Measuring execution times on the simulator allows estimating resource requirements for actual quantum computers. We observed that data embedding time doubles with each additional qubit, model execution time scales quadratically with the number of qubits and linearly with the number of blocks (Fig. 5). These observations are of importance for the use of real quantum computers and are discussed in the section below. We used up to 20 qubits for this test.

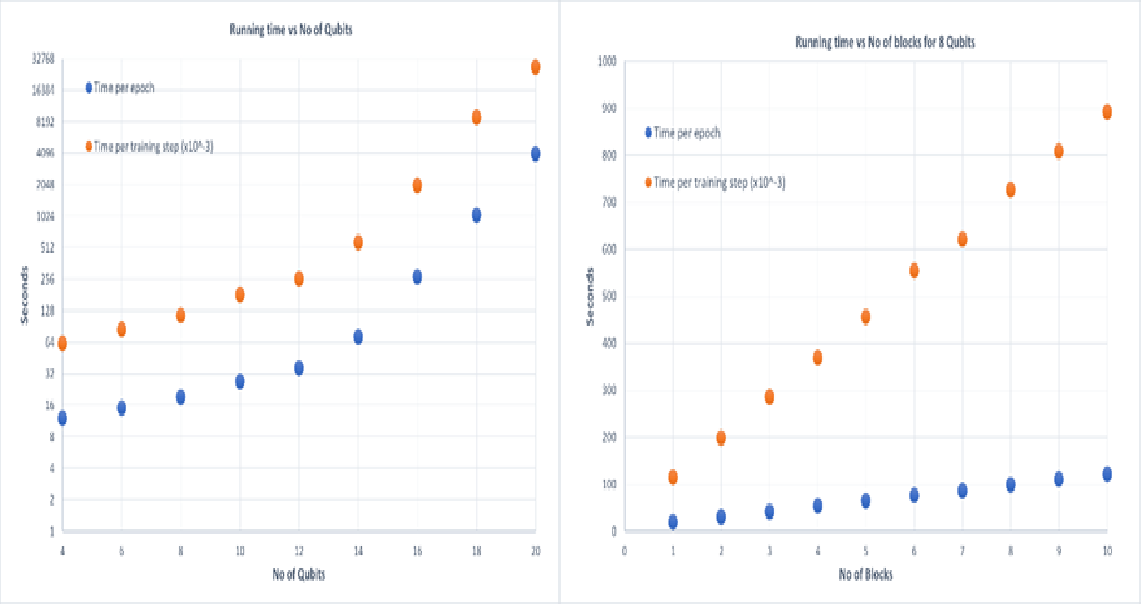


Figure 5: The execution time for the FH model on the simulator per epoch and per training step for an increasing number of total qubits (left), and for an increasing number of blocks with a constant number of 8 qubits (right).

6. DISCUSSION

Firstly, we would like to highlight that what we propose is a novel approach using a new architecture combining typical quantum variational circuits combined with a data re-uploading technique in a hybrid (classical-quantum) neural network (our new architecture is named under common name FULL HYBRID (FH) [12]).

We will now discuss the comparison of our FH and CC models as well as the execution times and limitations. We found the accuracy of the FH model to be equal to the CC model. Also, the FH model required significantly less epochs to train than the CC model by a factor if 2 or more (Fig. 5). Given that training time is a significant business driver this result is very interesting and grounds for optimism for practical quantum advantages in the future.

That the results are satisfactory, even when using a classical feeding layer to reduce the number of features fed into the quantum layer (limited by the number of qubits), is another positive finding. Currently, the number of qubits is limited in the simulator by memory requirements growing exponentially with the number of qubits. For real quantum computers the physical number of qubits is also limited by the available hardware and extra qubits needed for error correction. Consequently, this is the most practical FH architecture we can use for this specific dataset. In the future, increased computing power will enable us to benchmark with more qubits and produce more results.

The unexpected finding that increasing the number of qubits above 12 (Fig. 3) reduced the accuracy could be related to barren plateau issues but is to be investigated in future studies. The effect of adding more blocks, i.e. DRC, reducing the accuracy (Fig. 3) could be due to our dataset not having features that have a trigonometric (i.e. sine) structure thus re-entering the data to the model doesn't improve its performance [13]. It is not due to overfitting as the training is stopped before overfitting occurs.

We note that finding the best learning gradients in a non-convex landscape (for the problem structure) is an open question even in the classical machine learning community and no resolution has been discovered so far. To this end, one possible solution is to use Quantum Convolutional Neural networks and, moreover, the use of a data re-uploading circuit could possibly overcome this problem based on the prima facie argument that, since data is introduced many times to the network, the solution is forced from any local minima. Another promising method is to characterize the landscape by computing the Hessian of the loss function where, since the eigenvalues of the Hessian loss function quantify directly the local curvature of the loss function, we can adjust the learning rate of our model for faster convergence during the training process.

For simulators, we observe the exponential time complexity of increasing the number of qubits and blocks (Fig. 5), both of which would not be a problem for a real quantum computer as long as the circuit width (number of qubits) and circuit depth (number of gates) are within the specification of the quantum computer. However, another general issue for QML is the need to make many executions of the quantum circuit, for 350 epochs and a training set of 1798, this amounts to 629,300 circuit executions. This is not an issue for simulations run on a standalone computer. However, for a quantum computer accessed on a cloud platform, the overall time taken may not practical, for example it may take 7 days in total if each execution takes 1 second due to network overheads and queuing time on a shared quantum computer. Quantum computer providers such as IBM have very recently introduced mechanisms such as Qiskit runtime that enables the execution of the classical and quantum code to be run as one unit reducing the overall execution time by 120 times, potentially bringing 7 days down to 1.5 hours.

The dataset size is limited by the number of class 1 examples available. This has an impact on the most appropriate model but does not lead to overfitting. A much larger dataset such as the UK's with 1000+ defaults each year and over 4 million companies would be interesting to use with more complex models and finer grained risk periods. We also note that hyperparameters for the FH and CC models could potentially be tuned further.

6. CONCLUSION AND FUTURE WORK

In conclusion, the use of hybrid quantum/classical models is promising given the ease of obtaining comparable results to a purely classical counterpart and with much fewer epochs for training.

We have investigated the practical issues of using the models on simulators and on real quantum computers and expect that, with even modest improvements in hardware expected of over 4,000 qubits by 2025 [4], along with software improvements such as runtime environments [14], real advantages, at least in model training, will be achieved. It is also possible that improvements in accuracy may also be observed due to the resilience in the FH model.

However, the best classical model for Singapore data is still XGBoost and this work can be extended to other datasets, with more features using more qubits. Furthermore, this study shows that anyone in the machine learning community can relatively easily experiment with QML for their own problems.

The next step of moving to real quantum hardware may also prove interesting with the introduction of quantum noise possibly removing the need for dropout layers. The future for QML is very exciting.

Code is available on request.

Acknowledgements

References

[1] E. I. Altman, “Financial Ratios, Discriminant Analysis and the Prediction of Corporate Bankruptcy” *J. Finance*, vol. 23, no. 4, pp. 589–609, Sep. 1968, doi: 10.1111/j.1540-6261.1968.tb00843.x.

[2] J. Y. Campbell, J. Hilscher, and J. Szilagyi, “In Search of Distress Risk,” *J. Finance*, vol. 63, no. 6, pp. 2899–2939, 2008, doi: 10.1111/j.1540-6261.2008.01416.x.

[3] F. Wilczek, “Entanglement Made Simple,” *Quanta Magazine*, Apr. 28, 2016. https://www.quantamagazine.org/entanglement-made-simple-20160428/ (accessed May 20, 2022).

[4] “Expanding the IBM Quantum Roadmap to anticipate the future of quantum-centric supercomputing | IBM Research Blog.” https://research.ibm.com/blog/ibm-quantum-roadmap-2025 (accessed May 20, 2022).

[5] F. Louzada, A. Ara, and G. B. Fernandes, “Classification methods applied to credit scoring: Systematic review and overall comparison,” *Surv. Oper. Res. Manag. Sci.*, vol. 21, no. 2, pp. 117–134, Dec. 2016, doi: 10.1016/j.sorms.2016.10.001.

[6] V. S. Desai, J. N. Crook, and G. A. Overstreet, “A comparison of neural networks and linear scoring models in the credit union environment,” *Eur. J. Oper. Res.*, vol. 95, no. 1, pp. 24–37, 1996.

[7] Paul R Griffin, Michael Boguslavsky, Junye Huang, Robert J. Kauffman, and Brian R. Tan, “Quantum Computing: Computational excellence for Society 5.0,” in *Data Science and Innovations for Intelligent Systems Computational Excellence and Society 5.0*, 1st Edition., CRC Press, 2022, p. 392.

[8] D. J. Egger *et al.*, “Quantum Computing for Finance: State of the Art and Future Prospects,” *IEEE Trans. Quantum Eng.*, vol. 1, pp. 1–24, 2020, doi: 10.1109/TQE.2020.3030314.

[9] E. Farhi and H. Neven, “Classification with Quantum Neural Networks on Near Term Processors,” arXiv, arXiv:1802.06002, Aug. 2018. doi: 10.48550/arXiv.1802.06002.

[10] K. Bharti *et al.*, “Noisy intermediate-scale quantum (NISQ) algorithms,” *Rev. Mod. Phys.*, vol. 94, no. 1, p. 015004, Feb. 2022, doi: 10.1103/RevModPhys.94.015004.

[11] A. Pérez-Salinas, D. López-Núñez, A. García-Sáez, P. Forn-Díaz, and J. I. Latorre, “One qubit as a Universal Approximant,” *Phys. Rev. A*, vol. 104, no. 1, p. 012405, Jul. 2021, doi: 10.1103/PhysRevA.104.012405.

[12] N. Schetakis, D. Aghamalyan, M. Boguslavsky, and P. Griffin, “Binary classifiers for noisy datasets: a comparative study of existing quantum machine learning frameworks and some new approaches,” arXiv, arXiv:2111.03372, Nov. 2021. doi: 10.48550/arXiv.2111.03372.

[13] A. Pérez-Salinas, A. Cervera-Lierta, E. Gil-Fuster, and J. I. Latorre, “Data re-uploading for a universal quantum classifier,” *Quantum*, vol. 4, p. 226, Feb. 2020, doi: 10.22331/q-2020-02-06-226.

[14] “Qiskit Runtime,” *IBM Quantum*. https://quantum-computing.ibm.com/lab/docs/iql/runtime/ (accessed May 25, 2022).