qLogAnomaly: Applying Quantum Machine Learning for System Log Anomaly Detection

Vaibhav Ganatra
Department of Computer Science and Information Systems
BITS Pilani, K.K. Birla Goa Campus
Goa, India
f20190010@goa.bits-pilani.ac.in

Heng Li
Department of Computer and Software Engineering
Polytechnique Montreal
Montreal, Canada
heng.li@polymtl.ca

***Abstract*—Anomaly Detection in system logs is an essential in pointing out areas of abnormal behaviour within large complex software systems. The current methods of anomaly detection are limited by their high false-positive rate, and slow speed in processing large volumes of log data. In this work, we introduce the idea of applying quantum machine learning on the problem, and demonstrate the superior performance of quantum models in detecting anomalies in software logs. We establish a trend of improving performance of quantum models with increasing sizes of training data, while constantly outperforming classical models. This work opens up the domain of using quantum machine learning for problems in software engineering that are difficult to solve in a classical setting.**

***Index Terms*—anomaly detection, quantum machine learning, log analysis, qLogAnomaly**

I. Introduction

Recent years have witnessed a monumental growth in the scale of operation of software systems. Modern software has to serve millions of customers simultaneously with minimal downtime (ideally none) in order to ensure smooth operation. These are commonly hosted on large-scale distributed systems (such as Hadoop and Spark) or involve high-performance computing through supercomputers (such as Blue Gene/L) [1]. In order to ensure smooth performance, the mechanism of monitoring system health for such complex systems assumes vital importance. System logs are an essential indicator of the run-time performance of such systems. Identifying anomalies within the logs can help in pinpointing locations/components displaying abnormal behaviour and might help in speedy resolution in times of crashes, thereby ensuring minimal downtime.

Given the importance of anomaly detection in system logs, a wide variety of methods are prevalent for the task. For small scale systems, initially, the problem was tackled primarily through keyword searches and pattern matching within the logs. This involved a lot of manual inspection and careful crafting of anomaly-detecting patterns. Since this method did not scale well, Machine and Deep Learning based methods were adopted for anomaly detection. He et al. [2] provide a comprehensive review and an open-source implementation of the traditional ML methods used for solving the problem. This includes supervised methods such as Support Vector Machines (SVMs) [3] and Decision Trees [4], and unsupervised methods such as Log Clustering [5], Invariants Mining [6] and PCA [7]. Chen et al. [8] review and compare the performance of Deep Learning based techniques used in System Log Anomaly Detection. These techniques can also be classified into supervised - LogRobust [9] and CNN-based [10] and unsupervised techniques - DeepLog [11], LogAnomaly [12] and Logsy [13]. This distinction exists because logs are not usually labelled, and the process of labelling system logs is an expensive process that involves extensive manual effort.

While there exist multiple techniques for detecting anomalies in system logs, there are some challenges associated with them.

1) Existing methods have a high false-positive rate, *i.e.* they frequently label normal logs as anomalous thereby leading to false alarms of anomalous behaviour leading to unnecessary human effort [14].
2) It takes a long time to process large volumes of log data. Large systems like Amazon Web Services (AWS) generate petabytes of log data everyday, which takes a long time to be processed for anomaly detection.

Quantum Machine Learning (QML) has not yet, to the best knowledge of the authors, been applied to system log anomaly detection. While there are works which involve application of quantum algorithms to a different form of anomaly detection, such as anomaly detection in audio samples [15] and quantum speedup in anomaly detection in sequential data [16], this is the first instance, to the authors' best knowledge, of the application of quantum machine learning models to anomaly detection in system logs. Quantum Machine Learning models have achieved promising results and have been shown to have a "quantum advantage" in the domain of finance [17], where the use of quantum algorithms is on the rise. Sengupta [18] demonstrated the superior performance of quantum neural networks over classical neural networks on the biomedical problem of Covid-19 detection using patient CT-Scan images. QML algorithms have also been used for the quantum many-body simulation problem which is exponentially expensive to perform classically. It is, therefore, imperative to also leverage the "quantum advantage" of QML models to identify anomalies within system logs, thereby introducing the use of quantum computing for problems in software engineering. This application of quantum models for log analysis is also motivated by the following reasons:

1) The quantum feature space is exponentially larger than the corresponding classical feature space. Since it is commonly known that data inseparable in a smaller feature space, might become separable in a larger feature space, there is reason to believe that quantum models might have superior performance to classical models. This also points towards the possibility of quantum models being able to more effectively learn from small datasets in comparison to classical models.
2) Quantum models, such as quantum SVMs, provide an exponential speedup to classical models [19]. This might be useful for system log anomaly detection where one has to deal with very large volumes of logs.

In this work, the idea of applying QML for system log anomaly detection has been introduced and executed successfully. The performance of the quantum models has been compared against their classical counterparts, and the quantum supremacy in performance has been demonstrated. This superior performance is due to a novel quantum embedding that has been introduced which may be used for encoding high-dimensional data within qubits, without having to perform dimensionality-reduction. It is also shown that quantum models are more effective in learning from smaller datasets in comparison to classical models. Our work demonstrates a promising direction towards solving appropriate software engineering challenges with the new quantum computing capabilities.

The rest of the paper is divided as follows: Section II-A provides a brief summary about the existing methods for anomaly detection in system logs, Section II-B provides a brief insight into quantum machine learning models. In Section III-A, the various steps involved in anomaly detection of logs are highlighted, in Section III-B, the quantum embedding used for the quantum models is explained and in Section III-C, the details about the quantum models are provided. In Section IV, the performance of the models are demonstrated. This is followed by the Conclusion (Section V) and a discussion on the future research directions (Section VI).

II. Related Work and Background

A. Related work on log-based anomaly detection

As mentioned earlier, a number of ML models such as SVMs [3] and Decision Trees [4] have been used for log anomaly detection. DeepLog [11] is an LSTM-based model which models logs as a natural language sequence. PLELog [20] is a semi-supervised mechanism which utilises semantic meaning within logs for detecting anomalies. Log Clustering [5], LogAnomaly [12] and Logsy [13] are some of the unsupervised log anomaly detection methods.

B. Background on quantum machine learning

Quantum Machine Learning is a research field that incorporates the advantages of quantum computation within machine learning. It leverages the "quantum" aspects of computation for furthering the performance of machine learning models, and for approximating functions that are hard to model classically. Garcia et al. [21] provide a systematic review for quantum machine learning models and their applications. Here, we briefly explain the working Quantum Support Vector Machines (Q-SVMs) [22] and Quantum Neural Networks [23], which are used in the experiment sections. QSVMs replace the kernel function of the classical SVMs with a quantum circuit. This quantum circuit consists of unitary quantum gates, usually qubit rotations around certain axes followed by some form of entanglement, with the specific rotation values being determined by the datapoint being encoded. The quantum circuit may or may not be parametric. Since the features are embedded into the quantum feature space, which is exponentially bigger than the classical feature space, it is easier to model functions that are difficult to model classically. In quantum neural networks, there are 3 essential steps - encoding classical data into quantum states and rotation about axes parameterized by the weights, followed by introduction of some form of entanglement. The probability of measuring the qubits in a particular state is interpreted as the prediction probability and the weights are correspondingly trained through backpropagation.

III. Approach: Quantum Log Anomaly Detection

A. Steps in System Log Anomaly Detection

The steps involved in anomaly detection in system logs are:

1) **Log Collection** - Large scale distributed systems generate great volumes of logs during their runtime. These logs are collected and labelled as normal/ anomalous for further processing.
2) **Log Parsing** - There is structure within the system logs - the logs can be separated into certain event templates (constant part) and a variable part. For example if the log is *"Received block blk_321 of size 6618745 from /10.251.126.5"*, the event would look like *"Received block * of size * from *"*.
3) **Log Partitioning and Feature Extraction** - The parsed logs are then partitioned into sequences of events based on timestamp (for the BGL logs) or on the session (HDFS logs). In order convert these sequences into numerical features that can be used by Machine Learning models, event count vectors, counting the number of occurrences of each event template, are calculated for the event sequences.
4) **Anomaly Detection** - ML models are trained using the event count vectors as input, and during inference time, the event count vectors are fed into the model to classify it as normal or anomalous.

B. Quantum Embedding - qRot

The first step in using quantum models for classifying classical data is to encode the data into qubit states, so that they can be manipulated by unitary quantum operations. This is known as quantum encoding or quantum embedding of the classical data. Methods such as Amplitude Encoding [24]

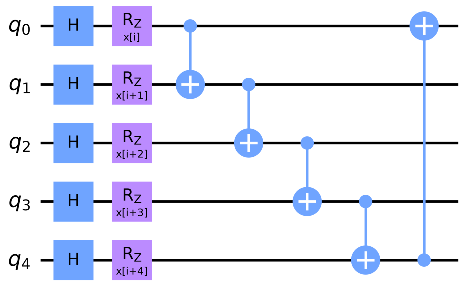

Fig. 1. 1 layer of the qRot embedding. The embedding is made up of multiple repetitions of this layer.

and Angle Encoding [25] are commonly used in Quantum Algorithms. It is also a common practice to perform some kind of dimensionality-reduction using algorithms such as Principal Component Analysis (PCA) and use the ZZFeatureMap [26] as the quantum encoding. Here, the dimensionality reduction is applied to reduce the number of features (while utilising maximum information from the data), which directly maps to the number of qubits used. However, the dimensionality reduction algorithms are designed to extract maximum classical information from data. It is not necessary that this classical information is the most informative input to the quantum model as well. Further, problems arise when one has to encode high-dimensional data without performing dimensionality reduction. Here, we introduce the qRot-embedding which uses multiple repetitions of a layer, based on qubit rotations about the Z-axis, to encode high-dimensional data without having to lose any information to dimensionality reduction. In the repetitions, the data cycles through after its completion. For example, if the data is 23-dimensional and 5 qubits are being used, then the encoding has 5 repetitions of the layer, with the features 1-20 being used in the first 4 repetitions. In the 5^{th} repetition, features 21-23 and 1-2 are used, i.e., the features cycle through. Fig. 1 shows the layer of the qRot embedding for 5 qubits. This can be modified as per the number of qubits being used. The layer consists of Hadamard gates, followed by rotations about the Z-axis parameterized by the value of the data being embedded. Next, entanglement is introduced through the Controlled-Not (CNOT) gates. This is introduced in a "ring" mannger where the state of each qubit controls the action of the NOT gate on the next qubit.

C. Quantum Models

Using the qRot embedding, 2 quantum models - Quantum Support Vector Machines (Q-SVMs) [22] and Quantum Neural Networks (QNNs) [23] were trained. As mentioned earlier, QSVMs are essentially classical SVMs, with their kernel function replaced by a quantum circuit. In the current work, the qRot embedding was used as the kernel function. The inner product $< x1|x2 >$ between two data points $x1$ and $x2$ was obtained by using the adjoint of the qRot-layer for $x1$ followed by the normal qRot-layer for $x2$ to evolve the quantum circuit, and then measuring the probability of measuring the qubit states as all zeroes. For QNN, the same qRot-embedding is used for encoding the classical data into quantum states, followed by Hadamard gates, parametric rotations around each axis and entanglement introduced by CNOT gates in each layer, repeated twice, *i.e.,* 2 layers.

D. Experiment Setup

Classical and Quantum Models are trained on 2 publicly available labelled log datasets - the HDFS Dataset and the BGL Dataset. Ideally, these models should have been trained on the entire datasets for comparision. However, due to the inability to run such large circuits on IBM Quantum's real quantum computer due to very long queuing times, the results stated here have been obtained on quantum simulators. The results of the Q-SVMs have been obtained on Pennylane's [25] lightning-qubit simulator and those of QNNs on the default-qubit simulator. Since simulating a quantum computer classically is exponentially more expensive, we have trained models on several small balanced subsets of the dataset in an attempt to establish a trend with growing dataset sizes to show that quantum models outperform classical models.

The configuration of quantum models is as described in Section III-C. The Q-SVM performance is compared with that of classical SVMs. 3 classical SVMs - Linear SVM, and SVMs with Polynomial and RBF kernels were trained and the best performance out of the 3 was used as a representative of the best classical performance. The performance of QNNs is contrasted against a classical neural network with similar configurations - having 2 layers, with hidden size as 3.

The performance of the models is evaluated in terms of the accuracy, precion, recall and F1-score. Additionally, since the data is highly unbalanced, the Matthew's Correlation Coefficient Score (MCC-score) [27], which is a suitable metric for unbalanced datasets, has also been reported for the testing set, as it maintains the original distribution of the dataset.

IV. Results

Table I shows the performance of the classical/quantum SVMs on subsets of the HDFS dataset. The metrics are reported in the format *a/b* where a represents the metric for the classical model, and b for the corresponding quantum model. The cases where the quantum model outperforms the classical model have been underlined. Models have been trained on subsets having 50, 100...250 log sequences. The testing has been carried out on a subset of over 1500 log sequences containing $\sim$70 anomalies in the test set. Due to the large number of event templates in the BGL dataset, and the exponential complexity of quantum simulations, it was unfeasible to obtain the corresponding results of Q-SVMs on the BGL dataset.

Table II shows the performance of classical/quantum neural networks on subsets of the HDFS dataset. Models have been trained on the same subsets as that of QSVMs. Table III shows the performance of classical/quantum neural networks on subsets of the BGL dataset. Models have been trained on subsets of size 50 and 100. Due to the large number of event

TABLE I
PERFORMANCE OF CLASSICAL/QUANTUM SVM ON LOGS OF HDFS DATA

Data	Train Accuracy	Train Precision	Train Recall	Train F1	Test Accuracy	Test Precision	Test Recall	Test F1	Test MCC
50	0.675 / 0.65	1 / 1	0.35 / 0.3	0.519 / 0.462	0.503 / 0.933	0.073 / 0.317	0.731 / 0.25	0.133 / 0.28	0.098 / 0.247
100	0.763 / 0.738	1 / 1	0.525 / 0.475	0.689 / 0.644	0.505 / 0.973	0.077 / 0.903	0.769 / 0.539	0.139 / 0.675	0.116 / 0.686
150	0.742 / 0.733	1 / 1	0.483 / 0.467	0.652 / 0.636	0.947 / 0.965	0.448 / 0.75	0.411 / 0.411	0.429 / 0.531	0.401 / 0.5395
200	0.75 / 0.744	1 / 1	0.5 / 0.488	0.667 / 0.656	0.957 / 0.973	0.591 / 0.929	0.513 / 0.513	0.549 / 0.661	0.529 / 0.679
250	0.77 / 0.76	1 / 1	0.54 / 0.52	0.701 / 0.684	0.957 / 0.973	0.585 / 0.929	0.5 / 0.513	0.539 / 0.661	0.518 / 0.679

TABLE II
PERFORMANCE OF CLASSICAL / QUANTUM NEURAL NETWORKS ON LOGS OF THE HDFS DATASET

Size	Train Accuracy	Train Precision	Train Recall	Train F1	Test Accuracy	Test Precision	Test Recall	Test F1	Test MCC
50	0.675 / 0.675	1 / 1	0.35 / 0.35	0.5185 / 0.5185	0.507 / 0.89	0.08 / 0.1282	0.8077 / 0.1923	0.1456 / 0.1538	0.1326 / 0.0998
100	0.675 / 0.75	0.675 / 1	0.675 / 0.5	0.675 / 0.6667	0.529 / 0.928	0.08 / 0.2917	0.8077 / 0.2692	0.1456 / 0.28	0.1326 / 0.2424
150	0.65 / 0.6583	1 / 0.6462	0.3 / 0.7	0.4615 / 0.672	0.938 / 0.9387	0.2424 / 0.4167	0.1053 / 0.5263	0.1468 / 0.4651	0.1311 / 0.4364
200	0.75 / 0.6	1 / 0.9	0.5 / 0.225	0.6667 / 0.36	0.944 / 0.9667	0.413 / 1	0.25 / 0.3421	0.3115 / 0.5098	0.2939 / 0.5749

TABLE III
PERFORMANCE OF CLASSICAL / QUANTUM NEURAL NETWORKS ON LOGS OF THE BGL DATASET

Size	Train Accuracy	Train Precision	Train Recall	Train F1	Test Accuracy	Test Precision	Test Recall	Test F1	Test MCC
50	0.775 / 0.8	0.867 / 0.929	0.65 / 0.65	0.743 / 0.765	0.675 / 0.6875	0.416 / 0.439	0.423 / 0.45	0.42 / 44	0.194 / 0.227
100	0.6 / 0.825	0.666 / 1	0.4 / 0.65	0.5 / 0.788	0.675 / 0.71	0.41 / 0.452	0.519 / 0.443	0.458 / 0.448	0.234 / 0.251

templates and large running time of the simulator for the BGL dataset, it was infeasible to obtain the results for larger subsets.

From the results, it can be clearly seen that for a given size of the training data, quantum models consistently outperform classical models in terms of accuracy, F1-score and MCC-score. The generalization of the model improves as the size of the train set increases. This is indicated in the steady improvement of the test accuracy, f1-score and MCC-score with the increase in the train size. Extrapolating from this trend, the quantum model performance should improve further when trained on larger datasets and this performance is expected to be better than that of the classical models. Since the current experiments have been performed on quantum simulators, it is infeasible to train quantum models on large datasets, and training quantum models on physical quantum hardware on large datasets remains one of the future directions. Additionally, it can be seen that quantum models are able to generalize well on very small subsets too. As stated earlier, this can be attributed to the large quantum feature space, thereby allowing quantum models to fit small datasets better.

> Quantum Models learn more effectively and have better generalization than classical models when trained on very small datasets. The performance of both classical and quantum models improves with increase in training data. However, quantum models consistently outperform classical models for all dataset sizes and are expected to ourperform classical models on larger datasets.

V. CONCLUSION

In this work, Quantum Machine Learning has been successfully applied to the problem of System Log Anomaly Detection. Through the use the novel qRot embedding proposed in this work, quantum models have been demonstrated to outperform classical models. While the used data subsets are small and the model performance (F1-score, MCC-score) may not be very high, extrapolating from the apparent trends it is expected that quantum models will perform better than the classical models on large datasets too. Additionally, with the ability to fit smaller datasets more effectively, quantum models may require less labelled data to achieve similar performance as the classical models. This would be a useful reduction in the costs and manual labour necessary to obtain labelled logs for supervised log anomaly detection. Our work sheds light on a promising direction of applying quantum machine learning to software engineering tasks that are challenging for classical computing paradigms.

VI. FUTURE PLANS

The goal of this work was to explore quantum machine learning for solving the challenging problem of system log anomaly detection. While the idea has been introduced, a full-scale incorporation of quantum machine learning into log anomaly detection largely rests on the widespread availability of quantum computers in the future. The current work is limited by the exponential cost of simulating a quantum system classically. Therefore, the future plans for this research would be to perform experiments on real quantum hardware on larger datasets. A more comprehensive evaluation of the performance of QML models can be carried out by training a more diverse set of quantum models on multiple log datasets. Another future research direction of this work is to exploit quantum parallelism to speed up the process of system log anomaly detection and compare the efficiency of quantum models to the classical models. The authors believe that this might be a interesting initial step towards the incorporation of quantum computation in software engineering and hope that this work can inspire new innovations and optimizations which can be applied to problems in Software Engineering in the NISQ era of quantum computing.

References

[1] N. Adiga and the Blue Gene/L Team, "An overview of the bluegene/l supercomputer," in *Proceedings of the 2002 ACM/IEEE Conference on Supercomputing*, SC '02, (Washington, DC, USA), p. 1–22, IEEE Computer Society Press, 2002.

[2] S. He, J. Zhu, P. He, and M. R. Lyu, "Experience report: System log analysis for anomaly detection," in *2016 IEEE 27th International Symposium on Software Reliability Engineering (ISSRE)*, pp. 207–218, 2016.

[3] Y. Liang, Y. Zhang, H. Xiong, and R. Sahoo, "Failure prediction in ibm bluegene/l event logs," in *Seventh IEEE International Conference on Data Mining (ICDM 2007)*, pp. 583–588, 2007.

[4] M. Chen, A. Zheng, J. Lloyd, M. Jordan, and E. Brewer, "Failure diagnosis using decision trees," in *International Conference on Autonomic Computing, 2004. Proceedings.*, pp. 36–43, 2004.

[5] Q. Lin, H. Zhang, J.-G. Lou, Y. Zhang, and X. Chen, "Log clustering based problem identification for online service systems," ICSE '16, (New York, NY, USA), p. 102–111, Association for Computing Machinery, 2016.

[6] J.-G. Lou, Q. Fu, S. Yang, Y. Xu, and J. Li, "Mining invariants from console logs for system problem detection," in *Proceedings of the 2010 USENIX Conference on USENIX Annual Technical Conference*, USENIXATC'10, (USA), p. 24, USENIX Association, 2010.

[7] W. Xu, L. Huang, A. Fox, D. Patterson, and M. I. Jordan, "Detecting large-scale system problems by mining console logs," SOSP '09, (New York, NY, USA), p. 117–132, Association for Computing Machinery, 2009.

[8] Z. Chen, J. Liu, W. Gu, Y. Su, and M. R. Lyu, "Experience report: Deep learning-based system log analysis for anomaly detection," 2021.

[9] X. Zhang, Y. Xu, Q. Lin, B. Qiao, H. Zhang, Y. Dang, C. Xie, X. Yang, Q. Cheng, Z. Li, J. Chen, X. He, R. Yao, J.-G. Lou, M. Chintalapati, F. Shen, and D. Zhang, "Robust log-based anomaly detection on unstable log data," ESEC/FSE 2019, (New York, NY, USA), p. 807–817, Association for Computing Machinery, 2019.

[10] S. Lu, X. Wei, Y. Li, and L. Wang, "Detecting anomaly in big data system logs using convolutional neural network," in *2018 IEEE 16th Intl Conf on Dependable, Autonomic and Secure Computing, 16th Intl Conf on Pervasive Intelligence and Computing, 4th Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress(DASC/PiCom/DataCom/CyberSciTech)*, pp. 151–158, 2018.

[11] M. Du, F. Li, G. Zheng, and V. Srikumar, "Deeplog: Anomaly detection and diagnosis from system logs through deep learning," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, CCS '17, (New York, NY, USA), p. 1285–1298, Association for Computing Machinery, 2017.

[12] W. Meng, Y. Liu, Y. Zhu, S. Zhang, D. Pei, Y. Liu, Y. Chen, R. Zhang, S. Tao, P. Sun, and R. Zhou, "Loganomaly: Unsupervised detection of sequential and quantitative anomalies in unstructured logs," in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pp. 4739–4745, International Joint Conferences on Artificial Intelligence Organization, 7 2019.

[13] S. Nedelkoski, J. Bogatinovski, A. Acker, J. Cardoso, and O. Kao, "Self-attentive classification-based anomaly detection in unstructured logs," 2020.

[14] V.-H. Le and H. Zhang, "Log-based anomaly detection with deep learning," in *Proceedings of the 44th International Conference on Software Engineering*, ACM, may 2022.

[15] Z. Chai, Y. Liu, M. Wang, Y. Guo, F. Shi, Z. Li, Y. Wang, and J. Du, "Quantum anomaly detection of audio samples with a spin processor in diamond," 2022.

[16] M.-C. Guo, H.-L. Liu, S.-J. Pan, W.-M. Li, S.-J. Qin, X.-Y. Huang, F. Gao, and Q.-Y. Wen, "Quantum algorithm for anomaly detection of sequences," 2022.

[17] M. Pistoia, S. F. Ahmad, A. Ajagekar, A. Buts, S. Chakrabarti, D. Herman, S. Hu, A. Jena, P. Minssen, P. Niroula, A. Rattew, Y. Sun, and R. Yalovetzky, "Quantum machine learning for finance," 2021.

[18] K. Sengupta, "Quantum algorithm for quicker clinical prognostic analysis: An application and experimental study using ct scan images of covid-19 cases," *BMC Medical Informatics and Decision Making*, 07 2021.

[19] G. Gentinetta, A. Thomsen, D. Sutter, and S. Woerner, "The complexity of quantum support vector machines," 2022.

[20] L. Yang, J. Chen, Z. Wang, W. Wang, J. Jiang, X. Dong, and W. Zhang, "Plelog: Semi-supervised log-based anomaly detection via probabilistic label estimation," in *2021 IEEE/ACM 43rd International Conference on Software Engineering: Companion Proceedings (ICSE-Companion)*, pp. 230–231, 2021.

[21] D. P. García, J. Cruz-Benito, and F. J. García-Peñalvo, "Systematic literature review: Quantum machine learning and its applications," 2022.

[22] V. Havlíček, A. D. Córcoles, K. Temme, A. W. Harrow, A. Kandala, J. M. Chow, and J. M. Gambetta, "Supervised learning with quantum-enhanced feature spaces," *Nature*, vol. 567, pp. 209–212, mar 2019.

[23] Y. Kwak, W. J. Yun, S. Jung, and J. Kim, "Quantum neural networks: Concepts, applications, and challenges," 2021.

[24] S. Ashhab, "Quantum state preparation protocol for encoding classical data into the amplitudes of a quantum information processing register's wave function," *Physical Review Research*, vol. 4, feb 2022.

[25] V. Bergholm, J. Izaac, M. Schuld, C. Gogolin, S. Ahmed, V. Ajith, M. S. Alam, G. Alonso-Linaje, B. AkashNarayanan, A. Asadi, J. M. Arrazola, U. Azad, S. Banning, C. Blank, T. R. Bromley, B. A. Cordier, J. Ceroni, A. Delgado, O. Di Matteo, A. Dusko, T. Garg, D. Guala, A. Hayes, R. Hill, A. Ijaz, T. Isacsson, D. Ittah, S. Jahangiri, P. Jain, E. Jiang, A. Khandelwal, K. Kottmann, R. A. Lang, C. Lee, T. Loke, A. Lowe, K. McKiernan, J. J. Meyer, J. A. Montañez-Barrera, R. Moyard, Z. Niu, L. J. O'Riordan, S. Oud, A. Panigrahi, C.-Y. Park, D. Polatajko, N. Quesada, C. Roberts, N. Sá, I. Schoch, B. Shi, S. Shu, S. Sim, A. Singh, I. Strandberg, J. Soni, A. Száva, S. Thabet, R. A. Vargas-Hernández, T. Vincent, N. Vitucci, M. Weber, D. Wierichs, R. Wiersema, M. Willmann, V. Wong, S. Zhang, and N. Killoran, "Pennylane: Automatic differentiation of hybrid quantum-classical computations," 2018.

[26] A. Abbas, S. Andersson, A. Asfaw, A. Corcoles, L. Bello, Y. Ben-Haim, M. Bozzo-Rey, S. Bravyi, N. Bronn, L. Capelluto, A. C. Vazquez, J. Ceroni, R. Chen, A. Frisch, J. Gambetta, S. Garion, L. Gil, S. D. L. P. Gonzalez, F. Harkins, T. Imamichi, P. Jayasinha, H. Kang, A. h. Karamlou, R. Loredo, D. McKay, A. Maldonado, A. Macaluso, A. Mezzacapo, Z. Minev, R. Movassagh, G. Nannicini, P. Nation, A. Phan, M. Pistoia, A. Rattew, J. Schaefer, J. Shabani, J. Smolin, J. Stenger, K. Temme, M. Tod, E. Wanzambi, S. Wood, and J. Wootton., "Learn quantum computation using qiskit," 2020.

[27] D. Chicco and G. Jurman, "The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation," *BMC Genomics*, vol. 21, 01 2020.