Quantum Machine Learning for Solar Panel Fault Detection

Trevor Irvin, Glen Uehara, Sameeksha Katoch, Andreas Spanias
SenSIP Center, School of ECEE, Arizona State University, University of Cyprus

***Abstract*—Solar power is becoming a common replacement for non-renewable resources such as fossil fuels. In order to optimize energy production of arrays, it is necessary to identify faults with accuracy. A solution to this problem is the use of machine learning to identify and classify solar array faults. When given features such as voltage, temperature, and irradiance from faulty and standard operating panels, a machine learning algorithm can be trained to predict if a solar panel is faulty as well as the type of fault the solar panel is experiencing. Using classical machine learning algorithms as a baseline, experimenting with quantum machine learning may provide valuable data in determining current quantum effectiveness. This paper explores a comparison of classical and quantum machine learning in order to determine effective solutions for fault detection.**

***Index Terms*—Machine Learning, Solar Panels, Quantum, Neural Network, Photovoltaics**

I. Introduction

In order to curb the use of limited and harmful fossil fuels, a transition to reliable, renewable energy sources has become an increasingly pressing objective. One such renewable solution is capturing solar energy through photovoltaic arrays. Harnessing solar energy through the use of PV arrays is becoming an affordable method to generate renewable energy due to the decreasing cost of producing solar panels [1]. This lowering financial entry paired with solar power's large technical potential [2] gives solar focus as the leading means of generating renewable energy.

A difficulty in utilizing PV arrays is ensuring the system is running at maximum efficiency. PV array faults present an obstacle in ensuring peak efficiency. In order to monitor PV panel inputs and outputs, Smart Monitoring Devices (SMDs) can be used. Data from these SMDs can be collected and fed to a machine learning algorithm trained for the classification of faults such as a neural network [3]–[6]. Arrays can then dynamically reconfigure their connection topology according to the fault [7]–[9].

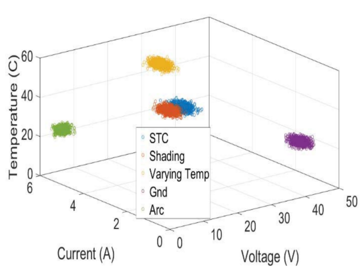

Fig. 1. Figure taken from [7] displaying K-means clusters of faults.

Possible supervised machine learning algorithms to explore include logistic regression, support vector machines (SVMs), and neural networks. Past works have found these algorithms to be effective in solving classification problems [10]. In particular, SVMs and neural networks may prove to be effective in solving problems with higher-dimensional data. Quantum algorithms based on these classical models have seen success using hybrid methods [11]–[13].

Quantum solutions can be used to process much more complex algorithms that may be unfeasible when performed on a classical machine which may lead to more accurate predictions. Additionally, the nature of quantum particles allows for more efficient finding of global minima in cost functions through the event of quantum tunneling [14].

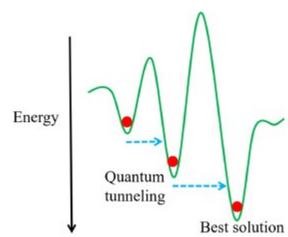

Fig. 2. Figure taken from [14] displaying quantum tunneling for cost functions.

This research proposes that quantum machine learning is an avenue worth exploring for photovoltaic fault detection. Quantum machine learning is an up-and-coming field that will prove essential to solving complex problems with large data sets which may prove useful in the area of fault detection. Hence, it is necessary to gauge the current effectiveness of quantum algorithms in solving these problems.

II. Data

The data used in this research was provided by Arizona State University. It is a simulated solar array dataset with ten inputs: Watts of energy produced, maximum voltage under load, maximum current under load, cell temperature, watts per meter squared, fill factor, gamma, peak maximum power, open circuit voltage, and short circuit current. Each set of inputs has a corresponding label split between four faults and standard test condition data. From these five labels, short circuit was chosen to be tested against the standard test condition data for the purpose of binary classification. Before being used for training, the data underwent pre-processing to normalize the input data.

III. Classical Models

To establish a baseline for future quantum tests, three classical models were implemented using Sklearn's machine learning packages within Python [15]. Each model was evaluated using its accuracy and f1-score.

The first model implemented was a logistic regression model. The hyper-parameters adjusted were the solver, the maximum iterations, and the penalty. The parameters that gave the best results were the saga solver with 5000 maximum iterations and the l1 penalty. The final test accuracy was measured to be 88.76%. The model displayed a slight tendency toward producing false negatives with an f1-score of 88.74.

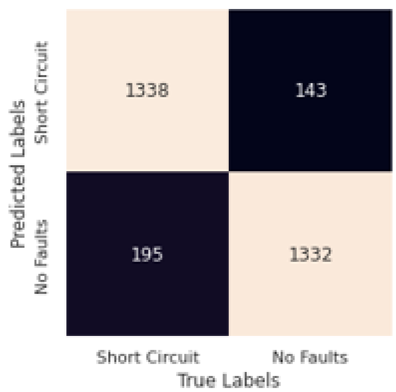

Fig. 3. Confusion matrix for logistic regression model.

The next model implemented was a state vector machine. The hyper-parameters adjusted were the kernel and the maximum iterations. The parameters that gave the best results were the RBF kernel with 5000 maximum iterations. The final test accuracy measured to be 91.29%. The model displayed a slight tendency toward producing false negatives with an f1-score of 91.25.

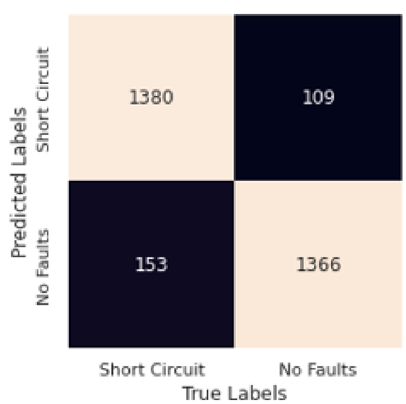

Fig. 4. Confusion matrix for support vector machine.

The final classical model tested was a neural network. The hyper-parameters adjusted were the number of hidden layers, the number of nodes, the activation function, and the maximum iterations. To find the optimal number of hidden layers and nodes a grid search was performed. The number of hidden layers within the search was constricted between one and two for the purpose of streamlining future quantum tests.

```
One hidden layer,  250  nodes. Validation accuracy =  94.35 %
One hidden layer,  300  nodes. Validation accuracy =  94.32 %
One hidden layer,  350  nodes. Validation accuracy =  94.51 %
One hidden layer,  400  nodes. Validation accuracy =  94.38 %
One hidden layer,  450  nodes. Validation accuracy =  94.58 %
One hidden layer,  500  nodes. Validation accuracy =  94.55 %
One hidden layer,  550  nodes. Validation accuracy =  94.58 %
One hidden layer,  600  nodes. Validation accuracy =  94.55 %
One hidden layer,  650  nodes. Validation accuracy =  94.58 %
One hidden layer,  700  nodes. Validation accuracy =  94.81 %
One hidden layer,  750  nodes. Validation accuracy =  94.78 %
Two hidden layers,  250  nodes. Validation accuracy =  94.68 %
Two hidden layers,  300  nodes. Validation accuracy =  94.81 %
Two hidden layers,  350  nodes. Validation accuracy =  94.35 %
Two hidden layers,  400  nodes. Validation accuracy =  94.81 %
Two hidden layers,  450  nodes. Validation accuracy =  93.55 %
Two hidden layers,  500  nodes. Validation accuracy =  94.45 %
Two hidden layers,  550  nodes. Validation accuracy =  94.58 %
Two hidden layers,  600  nodes. Validation accuracy =  94.25 %
Two hidden layers,  650  nodes. Validation accuracy =  94.71 %
Two hidden layers,  700  nodes. Validation accuracy =  94.68 %
Two hidden layers,  750  nodes. Validation accuracy =  94.61 %
```

Fig. 5. Grid Search results for neural network.

The hyper-parameters that gave the best results were two hidden layers each with 300 nodes using the relu activation function and the adam solver with 300 max iterations. The final test accuracy measured to be around 95%. The model displayed a very slight tendency toward producing false positives with an f1-score of 94.97.

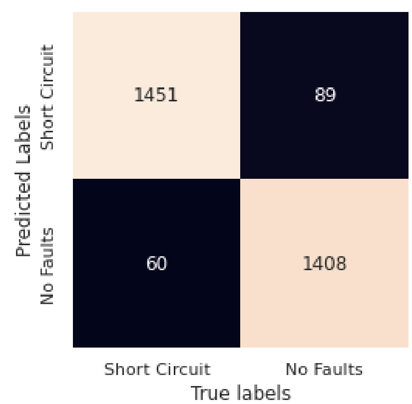

Fig. 6. Confusion matrix for neural network.

IV. Quantum Models

Tests using quantum models were performed using a combination of Qiskit and PyTorch python packages. Two models were tested: a quantum support vector machine and a quantum neural network (QNN). All tests were performed on quantum simulators. The same data was used as in the previous classical tests. As this is not data in quantum space, the data was first converted into Hilbert space before being used in the quantum circuit. After running through the circuit, the data is then converted back to classical data for evaluation.

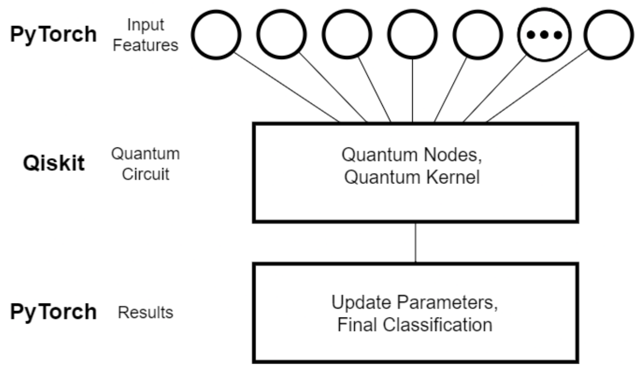

Fig. 7. Diagram of hybrid algorithm.

The first model tested was a quantum support vector machine. The hyper-parameters adjusted were the number of q-bits, number of shots, and the quantum simulator. The best results for the QSVM used 2 q-bits, 1024 shots, and the state-vector simulator. The QSVM had a final test accuracy of 93.58% with an f1-score of .94 with a slight tendency to produce false positives. The simulator produced results in good time allowing it to use a larger number of shots. Adjusting q-bits and shots above this amount produced negligible changes in results.

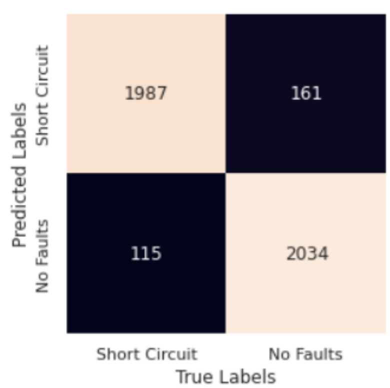

Fig. 8. Confusion matrix for quantum support vector machine.

The final model tested was a quantum neural network. The hyper-parameters adjusted were the number of q-bits, number of shots, the quantum simulator, the learning rate, and the activation function. The results for the QNN were obtained with 2 q-bits using 256 shots on the QASM simulator with a learning rate .03 and the sigmoid activation function. The QNN had a final test accuracy of around 55-60% with an f1-score of around .21 with a heavy tendency toward producing false negatives. The number of shots, q-bits, and hidden layers were limited due to the long run times of the simulator. Additionally, the number of hidden layers was chosen to match the parameters of the classical neural network.

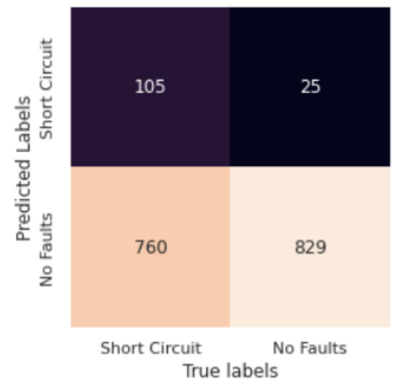

Fig. 9. Confusion matrix for quantum support vector machine.

V. Conclusion and Future Work

Of the classical models tested, the neural network produced the best results. However, the QSVM vastly outperforms against the QNN. Adjusting hyper-parameters and modifying the data set including dropping less relevant features failed to improve the accuracy. Previous research using QNNs and solar data produced results with accuracies over 90% [12]. It is possible that similar results are not being achieved due to the smaller size of our data set or because of sacrifices made to achieve reasonable run-times.

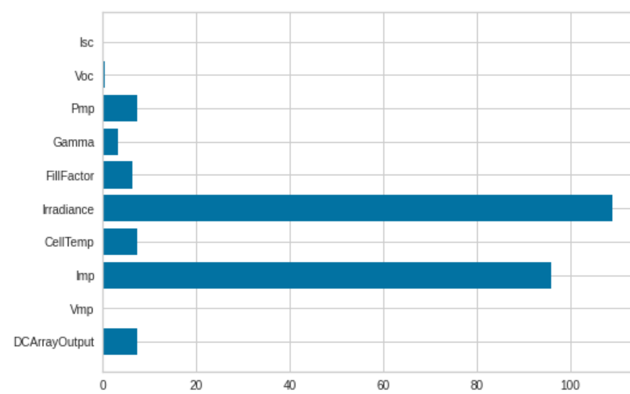

Fig. 10. Feature ranking of the solar data set according to 'most important'.

Contrary to the QNN, the QSVM produced results that surpassed its classical counterpart. The performance of the QSVM shows the possibility that a quantum model may be able to perform as well as or better than a classical model. The good performance of the QSVM and classical neural network suggest that if the quantum neural network's problems can be diagnosed, it may be the best solution for this problem.

For future work, researchers should perform tests using real solar array data as opposed to a simulated, hand-picked data set. Additionally, future tests should be performed on real quantum computers instead of quantum simulators. Researchers should also conduct future tests using Monte Carlo Simulations rather than taking a simple average.

REFERENCES

[1] N. Kannan and D. Vakeesan, "Solar energy for future world:-a review," *Renewable and Sustainable Energy Reviews*, vol. 62, pp. 1092–1105, 2016.

[2] G. R. Timilsina, L. Kurdgelashvili, and P. A. Narbel, "Solar energy: Markets, economics and policies," *Renewable and Sustainable Energy Reviews*, vol. 16, no. 1, pp. 449–465, 2012. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1364032111004199

[3] K. Jaskie and A. Spanias, "Positive and unlabeled learning algorithms and applications: A survey," in *2019 10th International Conference on Information, Intelligence, Systems and Applications (IISA)*. IEEE, 2019, pp. 1–8.

[4] ——, "Positive unlabeled learning," *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 16, no. 1, pp. 2–152, 2022.

[5] U. S. Shanthamallu, A. Spanias, C. Tepedelenlioglu, and M. Stanley, "A brief survey of machine learning methods and their sensor and iot applications," in *2017 8th International Conference on Information, Intelligence, Systems & Applications (IISA)*. IEEE, 2017, pp. 1–8.

[6] U. S. Shanthamallu and A. Spanias, "Machine and deep learning algorithms and applications," *Synthesis Lectures on Signal Processing*, vol. 12, no. 3, pp. 1–123, 2021.

[7] S. Rao, A. Spanias, and C. Tepedelenlioglu, "Solar array fault detection using neural networks," in *2019 IEEE International Conference on Industrial Cyber Physical Systems (ICPS)*. IEEE, 2019, pp. 196–200.

[8] S. Rao, S. Katoch, V. Narayanaswamy, G. Muniraju, C. Tepedelenlioglu, A. Spanias, P. Turaga, R. Ayyanar, and D. Srinivasan, "Machine learning for solar array monitoring, optimization, and control," *Synthesis Lectures on Power Electronics*, vol. 7, no. 1, pp. 1–91, 2020.

[9] H. Braun, S. Buddha, V. Krishnan, C. Tepedelenlioglu, A. Spanias, M. Banavar, and D. Srinivasan, "Topology reconfiguration for optimization of photovoltaic array output," *Sustainable Energy, Grids and Networks*, vol. 6, pp. 58–69, 2016.

[10] S. B. Kotsiantis, I. Zaharakis, P. Pintelas *et al.*, "Supervised machine learning: A review of classification techniques," *Emerging artificial intelligence applications in computer engineering*, vol. 160, no. 1, pp. 3–24, 2007.

[11] S. Endo, Z. Cai, S. C. Benjamin, and X. Yuan, "Hybrid quantum-classical algorithms and quantum error mitigation," *Journal of the Physical Society of Japan*, vol. 90, no. 3, p. 032001, 2021.

[12] G. Uehara, S. Rao, M. Dobson, C. Tepedelenlioglu, and A. Spanias, "Quantum neural network parameter estimation for photovoltaic fault detection," in *2021 12th International Conference on Information, Intelligence, Systems & Applications (IISA)*. IEEE, 2021, pp. 1–7.

[13] G. S. Uehara, A. Spanias, and W. Clark, "Quantum information processing algorithms with emphasis on machine learning," in *2021 12th International Conference on Information, Intelligence, Systems & Applications (IISA)*. IEEE, 2021, pp. 1–11.

[14] A. Ajagekar and F. You, "Quantum computing assisted deep learning for fault detection and diagnosis in industrial process systems," *Computers Chemical Engineering*, vol. 143, p. 107119, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0098135420308322

[15] S. Raschka and V. Mirjalili, *Python machine learning: Machine learning and deep learning with Python, scikit-learn, and TensorFlow 2*. Packt Publishing Ltd, 2019.

ACKNOWLEDGMENT

This research is funded by NSF Grant 1854273.