Quantum and Classical Machine Learning Algorithm Comparisons for Monitoring PV Array Faults

Kaden McGuffie[1], Glen Uehara[1], Andreas Spanias[1], Sameeksha Katoch[1], Lenos Hatzidemetriou[2]

1 ASU SenSIP Center, 2 UCy KIOS Center

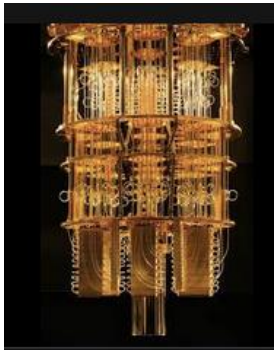
Sensor Signal and Information Processing Center: https://sensip.engineering.asu.edu/nsf-ires-project/

IRES project sponsored by NSF Award 35993

Presentation Agenda

- Benefits of machine learning for PV fault detection
- ASU Training
- Classical models
- Quantum models
- Challenges with quantum
- Project at University of Cyprus
- Findings
- Next steps
- Reflection on experience

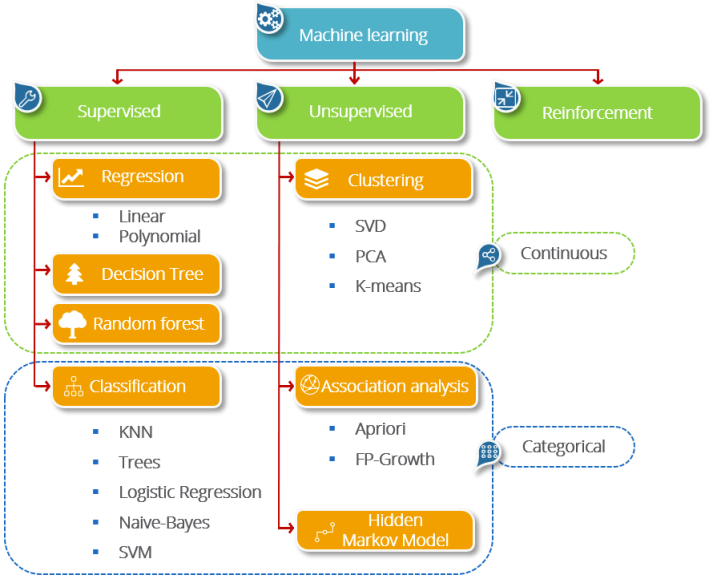
Problem Statement



- Classifying faults with machine learning is efficient
  - Manually finding faults is difficult and time consuming (expensive)
  - Loss of power production within PV arrays
  - Automatically finding faults allows for reconfiguration and faster fix time
  - Knowing the type of fault will further improve reconfiguration and fix time
- How will the shaded faults perform with different models
- Which model is superior
- Do quantum models have an edge over classical models for given problem

Training at ASU

- Python and Matlab training
- Machine learning
  - Supervised
  - Unsupervised
  - Reinforcement
- Supervised learning for fault detection
  - Support Vector Machine (SVM)
  - Logistic Regression (LR)
  - Neural Network (NN)
- Optimize each model for specific faults



robots.net

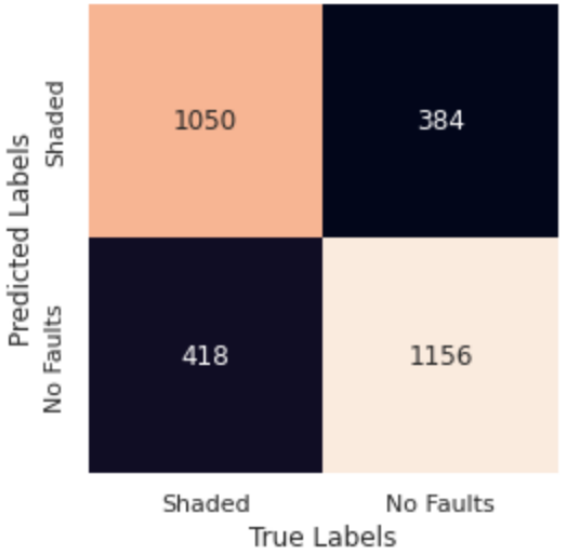
Logistic Regression

- LR uses the sigmoid function for its threshold

$$Sigmoid(x) = \frac{1}{1 + e^{-\theta x}}$$

- Key Hyperparameters
  - Solver
  - Penalty

- Shaded uses lbfgs, no penalty
  - Average accuracy 76%



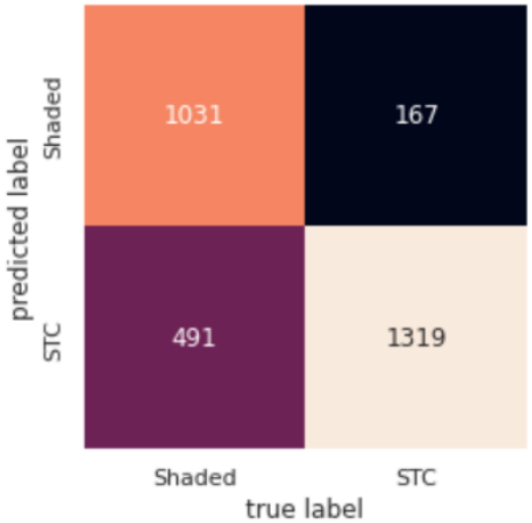
Support Vector Machine

- Key hyperparameters
  - Kernel
  - Degree (if polynomial)

$$K(x, x') = e^{-\gamma||x-x'||^2} \qquad \gamma = \frac{1}{n\,features * \sigma^2}$$

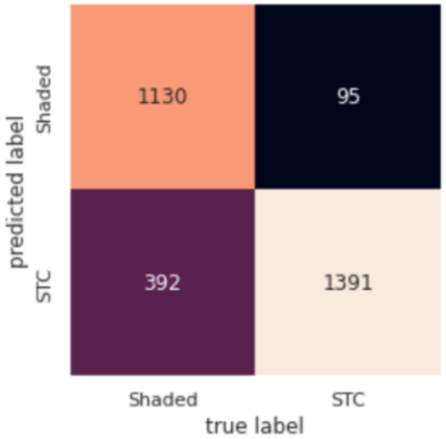
https://towardsdatascience.com/svm-classifier-and-rbf-kernel-how-to-make-better-models-in-python-73bb4914af5b

- Shaded uses rbf kernel
  - Average accuracy 75%
  - False Negatives



Shaded Neural Network

- Hyperparameters
  - Activation
  - Solver
  - Hidden layers
    - Nodes
- Test result average 83%



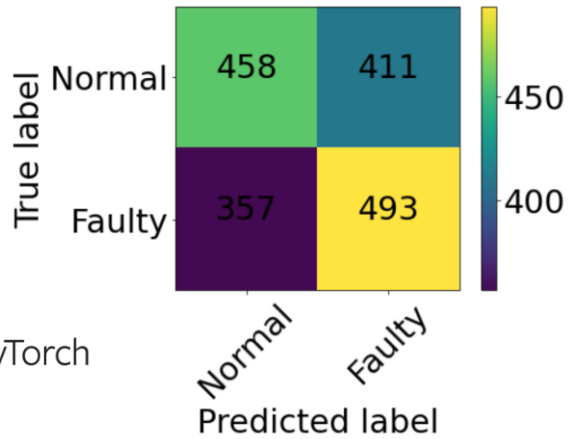

```
Two hidden layers,  150  nodes.  activation='relu', solver='adam'
```

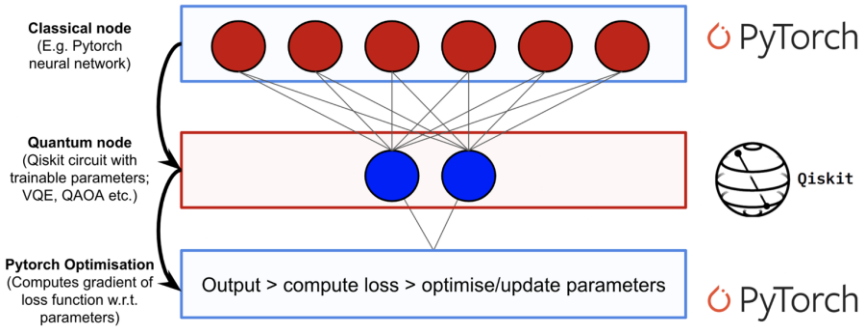
```
One hidden layer,  25  nodes. Validation accuracy =  80.55 %
One hidden layer,  50  nodes. Validation accuracy =  78.86 %
One hidden layer,  100  nodes. Validation accuracy =  80.35 %
One hidden layer,  150  nodes. Validation accuracy =  80.52 %
One hidden layer,  200  nodes. Validation accuracy =  80.35 %
One hidden layer,  250  nodes. Validation accuracy =  80.45 %
One hidden layer,  300  nodes. Validation accuracy =  81.02 %
Two hidden layers,  25  nodes. Validation accuracy =  82.51 %
Two hidden layers,  50  nodes. Validation accuracy =  82.01 %
Two hidden layers,  100  nodes. Validation accuracy =  82.18 %
Two hidden layers,  150  nodes. Validation accuracy =  82.51 %
Two hidden layers,  200  nodes. Validation accuracy =  81.42 %
Two hidden layers,  250  nodes. Validation accuracy =  81.55 %
Two hidden layers,  300  nodes. Validation accuracy =  81.32 %
```

Quantum Neural Network

- Inconsistent results
- Average accuracy below 60%
- Was not fixed with different
  - Backend
  - Learning rate
  - Epochs
  - Features included
  - Classes included

Shaded





https://qiskit.org/textbook/ch-machine-learning/machine-learning-qiskit-pytorch.html

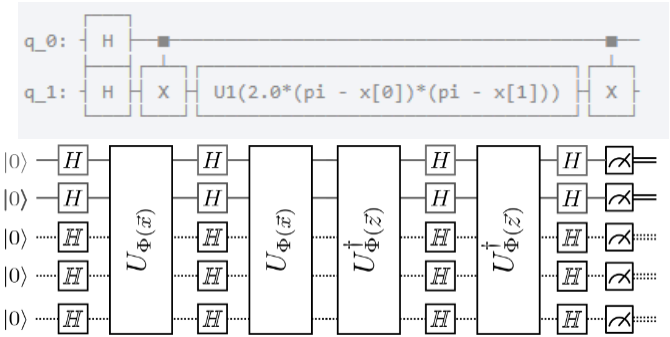
Quantum Support Vector Machine

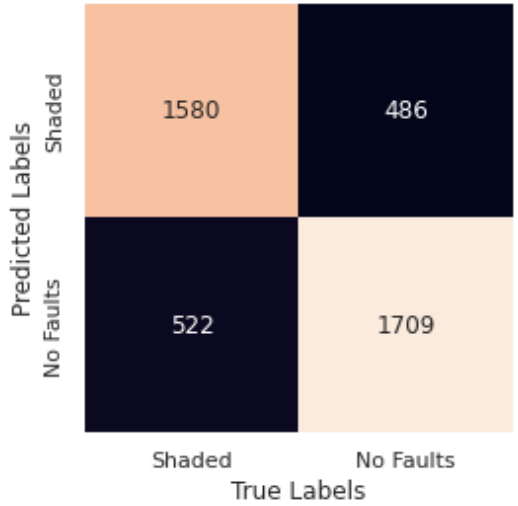
- Hyperparameters
  - # of Qubits
  - Simulator
- Shaded Results
  - Average accuracy 77%
- Better than classical version

```
adhoc_feature_map = ZZFeatureMap(feature_dimension=2, reps=2, entanglement="linear")

adhoc_backend = QuantumInstance(
    BasicAer.get_backend("statevector_simulator"), shots=1024, seed_simulator=seed, seed_transpiler=seed
)

adhoc_kernel = QuantumKernel(feature_map=adhoc_feature_map, quantum_instance=adhoc_backend)
```

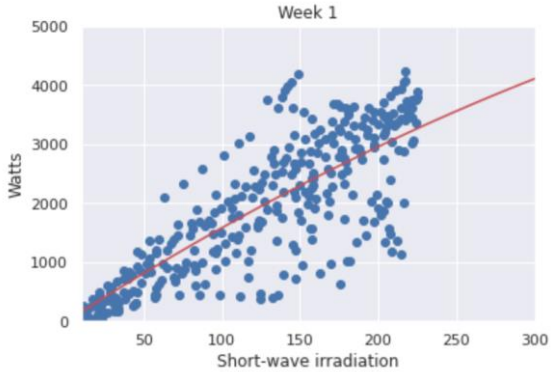


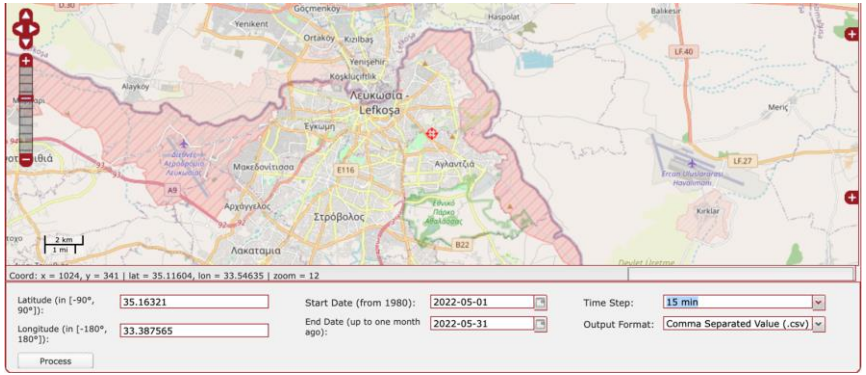


https://medium.com/@patrick.huembeli/introduction-into-quantum-support-vector-machines-727f3ccfa2b4

PV Power Generation Model

- Use data to predict PV power generation
  - Specific coordinates within Nicosia
- Input data from Soda Pro
- Power generation data provided by Lenos
- Preprocessing data
  - Two sites for input data
    - Humidity and Temperature data
    - Irradiance data
  - Time delay
  - Concatenating arrays





https://soda-pro.com/

PV Power Generation Model

- Different times of year
  - Whole year data
  - April-May
- 19 features used
  - Best with all features
- MLPRegressor NN
  - Activation function
  - Solver
  - Learning rate
- Whole year
  - Relu, lbfgs, invscaling
- April-May
  - Relu, adam, constant

```
X_train, X_test, y_train, y_test = train_test_split(X, y)
regr = MLPRegressor(activation = "relu", solver = 'lbfgs', learning_rate = 'invscaling', max_iter=10000).

regr.predict(X_test)
regr.score(X_test, y_test)
```

```
0.9266247497359663
```

```
[293] X_train, X_test, y_train, y_test = train_test_split(X, y)
      regr = MLPRegressor(solver = 'adam', max_iter=10000).fit(X_train, y_train)
      regr.predict(X_test)
      regr.score(X_test, y_test)
```

```
0.9624118000045835
```

Key Findings

Fault Detection

- Shaded faults performed best with classical neural network
  - Two layer, 150 nodes, relu, adam
- QSVM outperformed SVM
  - It is possible that QNN could outperform NN
  - Will possibly continue work on QNN

PV Power Generation Model

- All features helped predictions
- Select time period models will outperform year long predictions
- MLPRegressor NN predicts well

Next Steps

- Possible continuation with SenSIP
- Continue to learn about quantum computing
- Quantum Neural Network
- Solidify machine learning skills
- Learn more about hardware of solar arrays

Reflection on Experience and Self Assessment

- Basics of Quantum Computing and Machine Learning
- Preprocessing data
- How to problem solve with code
- KIOS Operations, projects, and research
- Visiting major cities in Cyprus and learning their history
  - Visited cultural and historical sites
- Living by myself in a foreign country
- How much I have yet to learn
- Scheduling my time
- The enjoyment of learning something new