# Comparison of Tornado Damage Prediction Acccuracy between Classical and Quantum Neural Networks

Omar Alsaid Sulaiman,* Sirikarn Phuangthong,† Jorge O. Cedeño,‡ and Suhani Sunder†

(Dated: August 10, 2024)

## I. INTRODUCTION

Tornadoes are among the deadliest and most common natural disasters worldwide. According to the National Weather Service, approximately 1.000 tornadoes occur annually in the United States, resulting in an average of 80 deaths and 1.500 injuries, alongside significant architectural damage [1]. Given this, research into tornado prediction has steadily advanced over the years, leading to the development of technologies that aid meteorologists in tornado prediction.

One such technology is the Doppler radar, which determines the direction and speed of air motion and precipitation by detecting changes in wavelength emitted from precipitation [2]. While widely used, Doppler radar heavily relies on the distance between the precipitation and the radar, limiting its ability to predict weather several days ahead. Another technology, the ensemble forecast, uses computational models to predict atmospheric patterns several days in advance for tornado prediction [3]. Although this addresses some limitations of Doppler radar, and allows for more confident predictions, the nonlinearity of the ever-changing atmosphere makes accurate tornado prediction challenging.

This has led to the implementation of machine learning, where collected data are used to train models to generate real-time predictions of future weather conditions based on environmental factors. In 2023, Schumacher et al. of Colorado State University developed a random-forest-based model known as CSU-MLP, which trained on nine years of weather data and can predict up to eight days in advance [4]. While heralded as revolutionary, this model tends to under-forecast severe weather events. Another model, developed in 2020, uses deep convolutional neural networks (CNNs) along with climate grids to predict tornado outbreaks in the United States [5]. However, it is most effective at predicting large tornado outbreaks with more than 20 tornadoes, and is less feasible for smaller yet still deadly events.

The use of quantum machine learning (QML) has emerged as a possibility to address complex problems in various domains including climate change and sustainability[6]. This approach offers a promising method to overcome classical machine learning limitations in climate change research by leveraging quantum computing [7] to better understand complex climate dynamics due to its ability to process and analyze intricate data sets at an unparalleled speed. Thus, better insights into climate models would be a significant advantage in enhancing predictive accuracy which allows for more informed decision making.

In this work, we aim to create a preliminary quantum neural network [8] model using Tensorflow[9] and Pennyland[12] for tornado detection and prediction of the monetary cost related to the damage it causes. Using the dataset from Storm Prediction Center (SPC)[11] database of the U.S. tornadoes from 1950 to 2023, we will compare the predictions of a classical multi-output neural network, against a hybrid (quantum + classical) neural network to evaluate the potential benefits of using quantum assisted neural networks for damage prediction. While it is unlikely we will be able to provide comprehensive results, we will offer demo code and initial test results to demonstrate our approach.

## II. BACKGROUND

The two main components of quantum mechanics that are integral to the formation and functionality of Quantum Neural Networks (QNN) are superposition and entanglement. [13] Superposition, as defined, is the ability of quantum bits to exist in multiple quantum states at one time, unless it is specifically mea-

* United Arab Emirates
† United States of America
‡ Republic of Ecuador

sured. Entanglement, on the other hand, allows for two particles to interact with each other, despite the distance between them.

As superposition and entanglement are fundamental discussions in quantum mechanics, the mathematics of quantum mechanics and quantum computing is essential here for a foundational understanding. [13] Quantum states, as expressed in physics and mathematics, is through the use of linear algebra. Contrary to classical computing, qubits exist in a Hilbert space, where they exist as states and are expressed as vectors, where the 0 state is expressed as the vector (1, 0) and the 1 state is expressed as the vector (0, 1). These states are then usually represented as a linear combination, as attached below, which allows for the processes of superposition and entanglement to be represented, as the square of the absolute value of $\alpha$ represents the qubit upon measurement at $|0\rangle$, and the same for $\beta$, with its equivalent being $|1\rangle$ (in this case, $\alpha$ and $\beta$ are complex numbers).

$$|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle \tag{1}$$

With these two inherent properties of quantum mechanics, that allow for particles and for qubits to interact in ways that would be impossible in classical mechanics and classical computing, the development and implementation of QNNs helps significantly improve the efficiency and scope at which prediction, modeling and data analysis takes place. [15]

In order to understand the difference between CNNs and QNNs, it is important to look at the differences that they fundamentally have.

Classical neural networks, on one hand, work through perceptrons, and a perceptron allows for multiple inputs to be converted into one output. [16] Added onto the multiple inputs are weights, that assign a 'value' to the input on how important or relevant that input is to the desired output, where weights are classified as real numbers. As the CNN begins to become more complex, there are multiple layers being added to the network, and in order to make sure that the output is as accurate as possible, with the least amount of error or loss, neural networks make use of two very important paths, to determine what the weight amount should be in order to achieve the most successful results: forward and back propagation. Forward propagation consists of the data being transferred in a traditional manner - from the input to the output. The equation used to describe this is:

$$a_i = W_{ij}X \tag{2}$$

Here, W is the weight, and X is the input vector.

Quantum Neural Networks, on the other hand, employ the same basics as CNNs, however, the way key processes like forward and backpropagation are carried out are different. [17] In QNNs, unlike in classical forward propagation, where inputs are passed through activation functions, instead, the use of quantum algorithms and quantized quantum parameters are applied, whilst the parameters are stored in registers whilst the algorithms are running.

Quantum algorithms here are represented as unitary operators, which in this case can be defined as:[17]

$$\hat{U}(\hat{\Phi}) := \sum_{\Phi} |\Phi\rangle\langle\Phi| \otimes \hat{U}(\Phi) \tag{3}$$

In order for forward propagation to take place, the initial input,[17] or in this case initial state engages in superposition with a quantum algorithm that results in a parametrized quantum algorithms, which is defined as:

$$\sum_{\Phi} \Psi_0(\Phi)|\Phi\rangle\hat{U}(\Phi)|\xi\rangle \tag{4}$$

Similar to how a loss function is added to a classical neural network, a loss function is applied here as well, to the algorithm, which influences the learning rate of the algorithm[17] . After the loss function has been applied, backpropagation is used, where there is a traceback to the quantum parameters that were implemented at the beginning.

As the field of quantum computing and machine learning continues to grow, there is extensive research currently being performed showing QNNs are performing much better than CNNs. [13] According to research conducted in Karlstad University and Deggendorf Institute of Technology by Nammouchi et al. Quantum Machine Learning has a significant role to play in climate change research, as quantum machine learning removes the limitations that classical computing imposes on the training and handling of data, where classical computing is unable to keep up with training complex data and training a large volume of data.

Additionally, when an intensive dive was made on the comparison of CNNs and QNNs, QNNs were found to be exponentially better performing the task, as shown by research performed by Safari et al., where QNNs were said to be the stronger set of technology, against Numerical Weather Prediction (NWP) and Artificial Neural Networks (ANN).[14] As mentioned, due to the application of quantum mechanics principles, and the fact that it makes use of atomic spins, QNNs are faster, with much better processing speeds, and produce more accurate results, with all of this achieved in a less amount of time.

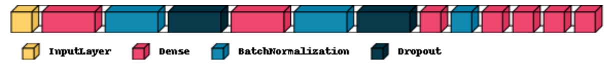


Figure 1. Classical Neural Network Layers

| Layer (type) | Output Shape | Param # | Connected to |
|---|---|---|---|
| input_layer_1 (InputLayer) | (None, 6) | 0 | - |
| dense_3 (Dense) | (None, 512) | 3,584 | input_layer_1[0][0] |
| batch_normalization_3 (BatchNormalization) | (None, 512) | 2,048 | dense_3[0][0] |
| dropout_2 (Dropout) | (None, 512) | 0 | batch_normalization_3… |
| dense_4 (Dense) | (None, 512) | 262,656 | dropout_2[0][0] |
| batch_normalization_4 (BatchNormalization) | (None, 512) | 2,048 | dense_4[0][0] |
| dropout_3 (Dropout) | (None, 512) | 0 | batch_normalization_4… |
| dense_5 (Dense) | (None, 64) | 32,832 | dropout_3[0][0] |
| batch_normalization_5 (BatchNormalization) | (None, 64) | 256 | dense_5[0][0] |
| ns (Dense) | (None, 1) | 65 | batch_normalization_5… |
| fat (Dense) | (None, 1) | 65 | batch_normalization_5… |
| inj (Dense) | (None, 1) | 65 | batch_normalization_5… |
| loss (Dense) | (None, 8) | 520 | batch_normalization_5… |

Figure 2. Classical Neural Network Structure

## III. METHODOLOGY

### A. Classical Neural Network

The data used to train the neutral network model was obtained from the database created Storm Prediction Center(SPC) of the United States[11], which consists of 70,022 records of historical tornado from 1950 to 2023. The relevant information for our model, which are used as inputs of our model, consists of the tornado's length in miles (len), width in yards (wid), magnitude with the scale of 0 to 5 (mag), year occurred (yr), state that was affected (st), and the geographical coordinates (latitude and longitude). To address the goal of tornado damage predictions, the outputs of interests are then consist of number of injuries (inj), fatalities (fat), number of states affected by tornadoes (ns), and the classification of the estimated property loss (loss). The distribution of both the inputs and outputs are chosen to create a multi-output neural network that will be used to estimate the possible damage cause by tornadoes. Pre-processing involves transforming economic loss data into millions and categorizing them into ranges from 0.0 to 7.0 to facilitate classification. Post-processing results in 55,198 viable tornado records for the model. The data is then split into training (80% of database) and testing sets (20% of database), followed by normalization to standardize the input features.

The construction of neural network model involves the use of TensorFlow[10] and Keras open packages, where the model architecture, where model architecture, optimizers, and loss functions are defined. To prevent overfitting and enhance the performance of the model, we optimize the training with callbacks to create checkpoints at the most probable epochs, terminate the training if the loss is too great, and increase the learning rate when reaching training plateau. The model's performance is then evaluated on the test set, generating metrics to access its accuracy and effectiveness in predicting or classifying tornado data.

The multi-output classical neural network built for this comparison consists of two identical blocks, each composed of a dense layer with 512 neurons, a batchnormalization layer, and a dropout layer of 40% to reduce the chances of overfitting, followed by a third block of a 64 neurons dense layer, accompanied with a batchnormalization layer. Finally, the model has four output layers, three of which correspond to a regression output, hence consisting of a singular neuron, while the classification layer has 8 neurons, a neuron corresponding to each class of property loss amount range.

### B. Hybrid Quantum-Classical Neural Network

The hybrid quantum-classical neural network was created by the implementation of quantum layers into the classical neural network previously cre-

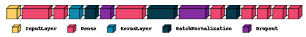


Figure 3. Quantum-Classical Neural Network Layers

```
Model: "model"
__________________________________________________________________________________________________
 Layer (type)                   Output Shape         Param #     Connected to
==================================================================================================
 input_1 (InputLayer)           [(None, 59)]         0           []

 dense (Dense)                  (None, 512)          30720       ['input_1[0][0]']

 dense_1 (Dense)                (None, 2)            1026        ['dense[0][0]']

 keras_layer (KerasLayer)       (None, 2)            4           ['dense_1[0][0]']

 batch_normalization (BatchNorm  (None, 2)           8           ['keras_layer[0][0]']
 alization)

 dropout (Dropout)              (None, 2)            0           ['batch_normalization[0][0]']

 dense_2 (Dense)                (None, 512)          1536        ['dropout[0][0]']

 batch_normalization_1 (BatchNo  (None, 512)         2048        ['dense_2[0][0]']
 rmalization)

 dropout_1 (Dropout)            (None, 512)          0           ['batch_normalization_1[0][0]']

 dense_3 (Dense)                (None, 64)           32832       ['dropout_1[0][0]']

 batch_normalization_2 (BatchNo  (None, 64)          256         ['dense_3[0][0]']
 rmalization)

 ns (Dense)                     (None, 1)            65          ['batch_normalization_2[0][0]']

 fat (Dense)                    (None, 1)            65          ['batch_normalization_2[0][0]']

 inj (Dense)                    (None, 1)            65          ['batch_normalization_2[0][0]']

 loss (Dense)                   (None, 4)            260         ['batch_normalization_2[0][0]']

==================================================================================================
Total params: 68,885
Trainable params: 67,729
Non-trainable params: 1,156
```

Figure 4. Quantum-Classical Neural Network Structure

```
Classification report:
              precision    recall  f1-score   support

         0.0       0.74      0.84      0.79      7281
         1.0       0.47      0.30      0.37      2613
         2.0       0.43      0.49      0.46      2102
         3.0       0.43      0.19      0.26       684
         4.0       0.34      0.09      0.15       148
         5.0       0.19      0.45      0.27        22
         6.0       0.00      0.00      0.00        57
         7.0       0.69      0.79      0.73       893

    accuracy                           0.64     13800
   macro avg       0.41      0.39      0.38     13800
weighted avg       0.61      0.64      0.62     13800
```

Figure 5. Classical Neural Network Confusion Matrix

ated in part A. This method allows a fair comparison, as by including quantum layers into the control model, the results of quantum computing enhancement can be clearly visualize. To do so, two approaches were made, with our initial approach utilize Tensorflow-Quantum[9], while the second approach utilize Pennylane[12].

Our first approach consists of implementing the quantum layer to the classical neural network with Tensorflow-Quantum, as Tensorflow-Quantum, Tensorflow, and Keras are commonly used in conjunction. For this method, we created a quantum circuit with trainable parameters in order to implement it into quantum layers. Keras was then used to convert the the quantum circuits into layers and implement it into the classical neural network. However, we encountered several issues with the code, as there are incompatibilities between the quantum layers and classical layers. We assumed that it is due to unsuccessful conversion, and due to the lack of time, we were unable to address the cause of this issue.

Our second attempt of implementation utilizes Pennylane and Keras, where two-qubits quantum circuits were converted into layers and integrated into the classical neural network. The quantum device chosen for this conversion is Pennylane's standard qubit-based device, and the calculation of the expected value of Pauli-Z measurement was performed on each qubit. This measurement was implemented into the weights of the model, along with our choice of number of quantum layer as 2, to create the quantum layers compatible in the hybrid model. In order to obtain initial results in the given time constraints and computational cost, we trained the model with only 10 epochs. Note that instead of the optimal choice of 8 neurons in our model, we were only able to run the model with 4 neurons due to shape-size problem encountered. **However, we strongly recommend you to run this with higher numbers of epochs to see how this model performs!**

## IV. RESULTS

The performance of the classical neural network can be seen in the heatmap and the evaluation snapshot in figure 6 and figure 7, respectively. The model's prediction scored 0.62 accuracy on the classification report, while the loss accuracy shown on the evaluation is 0.6360. Although we could not capture the complete data of the quantum-classical neural network, we noted that the quantum-classical neural network reached around 0.67 loss accuracy with four epochs. This initial discovery shows that the quantum-classical neural network has the potential to enhance the accuracy of the model, but further investigation in imperative. **Once again, we strongly recommend you to run this with the same number of epochs to see how this model performs.**

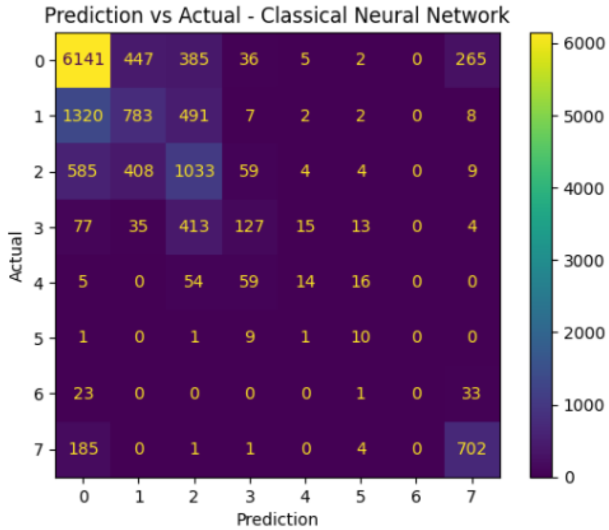


Figure 6. Classical Neural Network Heatmap

```
fat_mae: 0.1802 - inj_mae: 1.2820 - loss: 6.6499 - loss_accuracy: 0.6360 - ns_mae: 0.0187
```

Figure 7. Classical Neural Network Evaluation

## V. CONCLUSION

In this preliminary work, we have discussed our purpose of study, its importance, and the decision to implement quantum computing into machine learning model for weather prediction. As discussed in the introduction, machine learning models are an effective method for tornado prediction due to its ability to predict approximately a week in advance with pattern recognition. However, its greatest limitation includes the inability to predict non-linear patterns, such as atmospheric pattern, lowers its accuracy and power to predict further than 8 days.
Quantum Neural Networks (QNNs) leverage the principles of superposition and entanglement, fundamental to quantum mechanics, to significantly outperform classical neural networks (CNNs). Unlike CNNs, which rely on traditional forward and back-propagation, QNNs utilize quantum algorithms and parameters, enabling faster processing, greater accuracy, and more efficient handling of complex and large datasets. This makes QNNs particularly powerful in applications like climate change research and numerical weather prediction, where classical methods fall short. Overall, the integration of quantum mechanics into neural networks represents a significant advancement in computational capabilities.

While we successfully created a classical neural network model, we are unable to make a comparison between the models to investigate if the quantum-hybrid alternative does better than the classical neural network due to errors in the implementation of quantum layers into the classical neural network. We assumed this is due to the model's inability to convert the classical inputs into quantum inputs and back successfully, along with the computational power being far too great. This is our imperative future work, as initial goal was to see if quantum computing properties could enhance machine learning model in prediction of tornado damage.

While we successfully created both classical and quantum-classical neural network models, we were unable to make a fair comparisons between the models to investigate if the quantum-hybrid alternative does better than the classical neural network due lack of time and computational costs. However, the Quantum-classical neural network does show promises in enhancing the loss accuracy of the model. In the future, we plan to run the quantum-classical neural network models with 20 epochs and 8 qubits in order to compare both neural network models more fairly.

---


[1] National Weather Service. *Tornadoes FAQ*. Available at `https://www.weather.gov/lmk/tornadoesfaq#`, accessed July 2024.

[2] Illinois Extension. *Weather Radar*. Available at `https://extension.illinois.edu/weather/weather-radar`, accessed July 2024.

[3] Biello, D. *Can tornado prediction be improved?*. Nature, available at `https://doi.org/10.1038/news.2011.317`, 2011.

[4] Hill, A. J., R. S. Schumacher, and I. L. Jirak. *A New Paradigm for Medium-Range Severe Weather Forecasts: Probabilistic Random Forest–Based Predictions*. Wea. Forecasting, 38, 251–272, `https://doi.org/10.1175/WAF-D-22-0143.1`, 2023.

[5] McGuire, M. P., and T. W. Moore. *Prediction of tornado days in the United States with deep convolutional neural networks*. Computers & Geosciences, 159, 104990, https://doi.org/10.1016/j.cageo.2021.104990, 2022.

[6] Nammouchi, Amal and Kassler, Andreas and Theocharis, Andreas. 2023. *Quantum Machine Learning in Climate Change and Sustainability: A Short Review*. Proceedings of the AAAI Symposium Series.

[7] Singh, Manmeet and Dhara, Chirag and Kumar, Adarsh and Gill, Sukhpal Singh and Uhlig, Steve. 2022. *Quantum artificial intelligence for the science of climate change*. CRC Press.

[8] Cong, I., S. Choi, and M. D. Lukin. *Quantum convolutional neural networks*. Nat. Phys., 15, 1273–1278, https://doi.org/10.1038/s41567-019-0648-8, 2019.

[9] Abadi, M., et. al. *TensorFlow: Large-scale machine learning on heterogeneous systems*. https://doi.org/10.5281/zenodo.4724125, 2015.

[10] Javadi-Abhari, A. et al.*Quantum computing with Qiskit*. https://doi.org/10.48550/arXiv.2405.08810, 2024.

[11] Storm Prediction Center. *Storm prediction center maps, graphics, and Data Page*. Available at https://www.spc.noaa.gov/wcm/#data, accessed Aug. 2024.

[12] Pennylane, Available at https://pennylane.ai, accessed Aug. 2024.

[13] Nammouchi, A., et. al. *Quantum machine learning in climate change and sustainability: a review*. https://doi.org/10.48550/arXiv.2310.09162, 2023.

[14] A. Safari and A. A. Ghavifekr. *Quantum neural networks (QNN) application in weather prediction of smart grids*, 2021 11th Smart Grid Conference (SGC), Tabriz, Iran, Islamic Republic of, 2021, pp. 1-6, doi: 10.1109/SGC54087.2021.9664117.

[15] EITCA Academy. (n.d.). *How do the phenomena of superposition and entanglement enable quantum computers to perform certain calculations more efficiently than classical computers?*. Retrieved August 8, 2024 from https://eitca.org/artificial-intelligence/eitc-ai-tfqml-tensorflow-quantum-machine-learn introduction-eitc-ai-tfqml-tensorflow-quantum- introduction-to-google-ai-quantum/examination-review-introduction-to-google-ai-q how-do-the-phenomena-of-superposition-and-enta

[16] Venayagamoorthy, G. K. (2004, September). *Teaching neural networks concepts and their learning techniques*. In Proceedings of the 2004 American Society for Engineering Education Midwest Section Conference (Vol. 20).

[17] Verdon, G., Pye, J., et. al. *A universal training algorithm for quantum deep learning*. https://doi.org/10.48550/arXiv.1806.09729, 2018.