Quantum Neural Networks

A Path to Lower Emissions Through Fuel Consumption Prediction in Shipping

Chien, S. F.; Hermans, Julien J.M.; Kana, Austin A.; Zarakovitis, Charilaos C.; Zavvos, Stathis; Lim, H. S.

DOI
10.1109/ICASSP49660.2025.10888871

Publication date
2025

Document Version
Final published version

Published in
Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2025

ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 979-8-3503-6874-1/25/$31.00 ©2025 IEEE | DOI: 10.1109/ICASSP49660.2025.10888871

Quantum Neural Networks: A Path to Lower Emissions Through Fuel Consumption Prediction in Shipping

S.F. Chien[§,†], *Julien J.M. Hermans*[‡], *Austin A. Kana*[‡],
Charilaos. C. Zarakovitis[†], Stathis Zavvos[#], H.S. Lim[⋆,†]

[§] Strategic ICT, MIMOS Berhad, Kuala Lumpur, Malaysia
[‡] Delft University of Technology, Dept. of Maritime and Transport Technology, Delft, the Netherlands
[†] Axon Logic, Dept. of Applied Mathematics, 7 Dekelias, 14122, Athens, Greece
[#] VLTN BV, De Keyserlei 58-60 Bus 19, Antwerp, 2018, Belgium.
[⋆]Faculty of Engineering and Technology, Multimedia University, Melaka, Malaysia

ABSTRACT

This paper proposes Quantum Neural Networks (QNNs) as a data-driven approach for predicting fuel consumption. We utilize various layer architecture designs available in the Torchquantum framework, including both entangled and non-entangled circuit designs. In general, QNNs can achieve comparable Root Mean Square Error (RMSE) and Mean Absolute Percentage Error (MAPE) with significantly fewer trainable parameters. Neither pure QNNs nor hybrid QNN models exhibit the underfitting tendencies seen in classical neural networks (CNNs). Notably, one of the most significant findings of this work is that hybridizing or ”dressing” the quantum circuit leads to substantial improvements in RMSE and MAPE for pure QNNs. These promising results suggest potential optimizations for reducing emissions in green shipping.

Index Terms— Classical neural network, quantum neural network, parameterized quantum circuit, NISQ devices, digital twin, fuel consumption, modeling.

1. INTRODUCTION

The maritime industry produces extensive data related to vessel operations, route specifics, port activities, and more. However, much of this data remains underutilized, primarily due to challenges such as manual processing and limited application beyond specific purposes like incident assessment or environmental impact measurement [1]. To address this challenge, researchers recommend focusing research and innovation in digitalization and data usage within the shipping industry, exploring technologies such as artificial intelligence, augmented reality, virtual reality, high-performance computing, and big data analytics [1, 2]. These technologies hold potential for various maritime applications throughout the whole lifecycle of a ship, from design to operational phase. An example of such technology is a digital twin (DT).

A DT is model based systems engineering approach where a physical entity is represented by a virtual model and connected by a bi-directional data link [3]. A DT is capable of handling big data and allows for performing simulations. Even though DTs are already being considered mainly during certain operations, it also hold great potential to be implemented during the design of a ship for investigating possible designs. The Digital Twin for Green Shipping (DT4GS) project, funded by the European Union’s Horizon research program, investigates the potential of DT to be used for green ship design and operation. Within the project, operational data from four distinctive ship types is available to be used for DT purposes. In support of the DT4GS project, [4, 5] investigated the potential of operational data originating from a 300m bulk carrier within the project to be applied for the construction of a fuel consumption model. This fuel consumption model is selected to be the basis for the potential DT. Due to the vast amounts of available data, a CNN was originally selected as the model to predict the fuel consumption of the respective vessel.

Quantum Machine Learning (QML) is rapidly emerging as a powerful computational tool, as detailed in [6]. Recently developed quantum-based neural networks, such as Dissipative Quantum Neural Networks (DQNN), Quantum Recurrent Neural Networks (QRNN), and Quantum Convolutional Neural Networks (QCNN) or Quanvolutional Neural Networks, have demonstrated promising potential and efficiency for future applications [7, 8, 9, 10]. Furthermore, research by [11] has confirmed the feasibility of running QML algorithms on classical computers. This paper aims to expand those earlier studies by exploring the use of QML and compare its performance to the CNN. Recent studies [12, 13, 14, 15] have demonstrated that both purely quantum and hybrid classical-quantum machine learning approaches surpass the efficiency of traditional classical methods.These studies have motivated us to employ QNN, which may lead to a further reduction in fuel consumption through more accurate predictions. Another interesting point is that the efficacy of QNNs can now be evaluated using the metric of effective dimension, which is related to the Fisher information matrix [14]. Essentially, the greater the effective dimension, the better the performance of the neural networks. While CNNs typically operate in very high-dimensional parameter spaces, their true size, as represented by the effective dimension, is usually much smaller [14]. The primary objective of this paper extends beyond merely testing and comparing the performance of CNNs and QNNs using the specified dataset; we also broaden the investigation to include hybrid models that combine both approaches. Note that quantum computation is currently performed on Noisy Intermediate-Scale Quantum (NISQ) devices [16]. This advancement has made quantum computation feasible and has inspired the research community to develop hybrid classical-quantum approaches. The contributions of this research are threefold:

- **Performance Evaluation:** We analyze the performance of

This work was supported by the EU Horizon Europe projects DT4GS (ID:101056799) and OASEES (ID:101092702).

CNNs and QNNs, specifically in terms of the achievable loss rates, including Root Mean Square Error (RMSE) and Mean Absolute Percentage Error (MAPE).

- **Quantum Circuit Architectures:** We examine various parameterized quantum circuit (PQC) architectures, focusing on different layer designs, which include non-entanglement circuits, circuits with entanglement and single-qubit gates, and entangled-circuits with two-qubit gates.
- **Hybrid CNN-QNN Models:** We explore the potential of hybrid models combining CNNs with quantum circuits, particularly focusing on enhancing quantum circuits with additional classical layers.

2. DATA PRE-PROCESSING AND MODEL CONSTRUCTION

The operational data used in this work originates from the bunker delivery notes (BDNs) of a 300m bulk carrier which is one of the living labs in the DT4GS project. BDNs are one of the acceptable data collection methods by the International Maritime Organization (IMO) for the yearly Carbon Intensity Indicator (CII) calculation, mandatory for most transport ship types. The CII expresses the environmental impact of ship with regard to its operational profile per calendar year. This dataset was first used in [5] where the objective was to investigate the potential of this data to be used for modeling construction achieving emission reduction through design. The model selection for this investigation resulted in a fuel consumption model representing the respective ship and composed of a CNN, in combination with wind assisted ship propulsion models which are based on physical principals.

The BDNs contained over 129,000 data points, with a time interval of 5 minutes during the period between 02/06/'22 - 30/09/'23, where each data point contained over 100 different data types, such as ship speed, main engine power or fuel consumption. For the CNN input selection a Spearman rank correlation analysis was performed, identifying the data types which have a strong correlation with the ship's main engine fuel consumption, resulting in 7 input parameters for the CNN. An additional input value for the CNN was calculated by a resistance model, resulting in a total of 8 input values for the CNN. After applying the adopted preprocessing framework, the number of data points was reduced to 5,678 representing 'pure' sailing hours of the vessel. Additional information regarding the original model architecture, data preprocessing, model input selection and the case-study performed with the constructed models is provided in [4, 5].

3. PARAMETERIZED QUANTUM CIRCUIT

In quantum computation and information, the qubit (quantum bit) is the basic unit of information, analogous to the classical bit in traditional computing. However, unlike classical bits, which can be in a state of 0 or 1, qubits can exist in a superposition of both states simultaneously, and which is a linear combination of $|0\rangle$ and $|1\rangle$ states. For instance, a state of a qubit can be expressed as $|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle$, where α and β are normalized complex numbers, satisfying $||\alpha||^2 + ||\beta||^2 = 1$. With this special property, along with entanglement and quantum interference, enables quantum computers to perform certain types of computations much more efficiently than classical computers.

The manipulation of quantum states is carried out using a quantum gate, which acts as a unitary quantum operator and can be mathematically described by a unitary matrix. On the other hand, the Bloch Sphere provides a visual representation of the dynamical evolution of a qubit state. In quantum computing, commonly used gates in PQC or variational quantum circuit (VQC) include single-qubit gates $R_x(\theta)$, $R_y(\theta)$, and $R_z(\theta)$, which induce the rotation of a qubit state around the corresponding axes by an angle θ. The universal single-qubit rotation gate is known as U3(θ, ϕ, λ) where θ, ϕ, and λ are the Euler angles. However, the complexity of these operations goes beyond what can be fully explained using the relatively simple Bloch sphere diagram. Notably, all these gates maintain an equal number of inputs and outputs, ensuring that no information is lost within the quantum system. Examples of two-qubit gates are Controlled-Not (CNOT) Gate that operates on two qubits i.e., a control qubit and a target qubit; SWAP gate exchanges the states of two qubits; RZZ gate that performs rotations about the Z axis on two qubits; and etc.

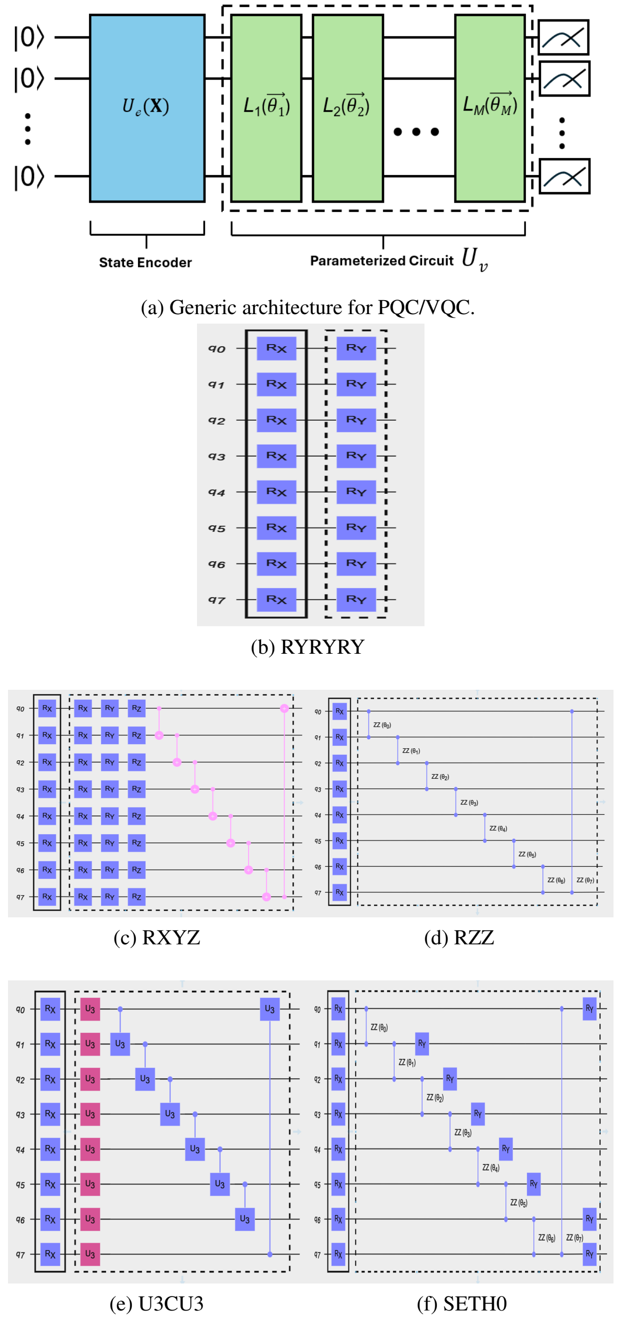
3.1. Functioning of a PQC or VQC

PQC OR VQC consist of learnable parameters that can be manipulate through iterative optimizations. The loss rate calculation portion is handled by classical computing resources. All these parameterized will be optimized via classical machine learning such as gradient-based and non-gradient-based algorithms. As shown in Fig. 1(a), the design of a Variational Quantum Circuit (VQC) consists of three distinct components: the encoding circuit, the parameterized circuit, and the quantum measurement layer.

The first component is the state encoding module $U_e(\mathbf{x})$, which transforms classical input information $\mathbf{x}$ into qubit states $U_e(\mathbf{x})|0\rangle^{\otimes n}$ enabling further processing in the subsequent module. This second module is known as the variational module or ansatz U_v. The design of the ansatz layer is arbitrary and often application-dependent. The internal design can include either entangled or non-entangled states. If entanglement is required, the CNOT or RZZ gate can be used, and if qubit state rotations are needed, R_x, R_y or R_z are deployed. It is important to note that repeating the layer design is permitted and often enhances the performance of the circuit. This repetition allows for the incorporation of additional learning parameters. In short, the variational module U_v can be written as $U_v(\Theta) = L_M(\vec{\theta_M})L_{M-1}(\vec{\theta_{M-1}})\cdots L_1(\vec{\theta_1})$, where M represents the total number of layers and Θ is the collection of all trainable parameters in different layers $\{\vec{\theta_1}, \vec{\theta_2}\cdots\vec{\theta_M}\}$.The final module is the measurement module $\mathcal{M}$, functioning to extract the expectation value of the qubit states within the quantum circuit. Unlike classical computers, quantum results are obtained by running the circuit multiple times, a process known as 'shots.' Consequently, the results are the expectation values of each qubit. Typically, the Pauli-Z expectation is a widely adopted choice in QML applications. The merit of PQC lies in its capability to seamlessly integrate with other classical components, such as matrix product states and deep neural networks. This flexibility permits data pre-processing, including dimensionality reduction (circuit dressing) [17], as well as post-processing to achieve scaling objectives. The operation of the PQC used in this work can be expressed as $\overrightarrow{g(\mathbf{x};\Theta)} = \left(\left\langle \hat{Z}_1 \right\rangle, \cdots, \left\langle \hat{Z}_Q \right\rangle\right)$, where $\left\langle \hat{Z}_l \right\rangle = \left\langle 0 \left| U_e^\dagger(\mathbf{x})U_v^\dagger(\Theta)\hat{Z}_l U_v(\Theta)U_e(\mathbf{x}) \right| 0 \right\rangle$ and Q represents the total number of qubits measured in the system. The expectation values $\left\langle \hat{Z}_l \right\rangle$ can be obtained analytically when the circuit is simulated classically. Similar to CNN, the quantum circuit parameters are updated for each training epoch during the simulation. The quantum circuit computation is optimized using an optimizer that guides the search direction towards optimal values based on the target outputs.

Table 1: Total Trainability Parameters, Quantum Circuit Depth, RMSE, and MAPE for CNNs, QNNs, and Hybrid QNNs

	CNN-20	CNN-8	RYRYRY	RXYZ	RZZ	SETH0	U3CU3	DRESSED-10	Hybrid 4Q-SETH0	Hybrid 8Q-SETH0	Hybrid 8Q-XYZ
Total Trainable Parameters	701	185	16	48	16	32	96	210	64	212	228
Quantum Circuit Depth	-	-	2	22	16	18	18	18	8	18	22
RMSE (Train)	101.40	125.67	87.52	79.13	105.26	48.67	37.73	27.74	32.59	29.42	35.29
RMSE (Validation)	69.84	38.52	112.30	97.03	135.52	76.54	45.06	29.96	45.87	38.70	36.84
MAPE (Train)	5.76	7.21	5.14	4.53	6.16	2.53	1.88	1.44	1.77	1.58	1.93
MAPE (Validation)	3.37	2.18	9.52	8.46	11.15	5.21	3.07	1.94	3.24	2.61	2.77
RMSE (Test)	23.52	46.59	135.10	118.62	112.52	49.86	33.19	27.23	19.23	20.05	27.03
MAPE(Test)	1.56	3.34	8.43	9.98	9.67	4.45	2.80	2.35	1.92	1.83	2.53



(a) Generic architecture for PQC/VQC.

(b) RYRYRY

(c) RXYZ

(d) RZZ

(e) U3CU3

(f) SETH0

Fig. 1: Generic architecture for PQC and various ansatz designs.

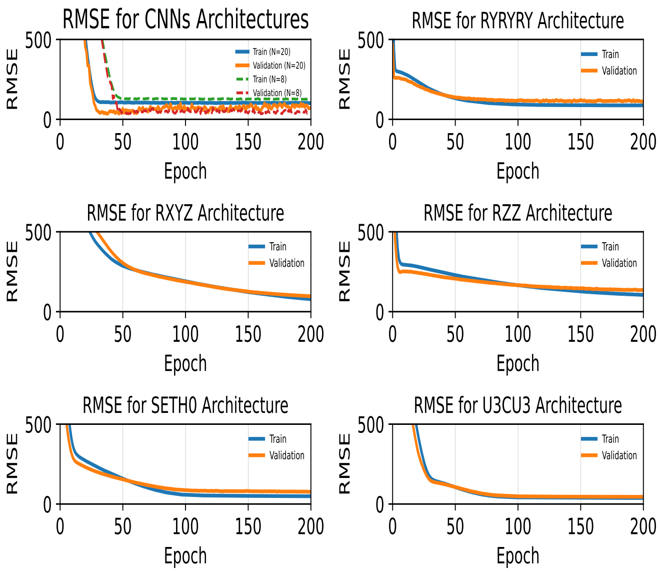


Fig. 2: RMSE for CNNs and pure quantum circuits

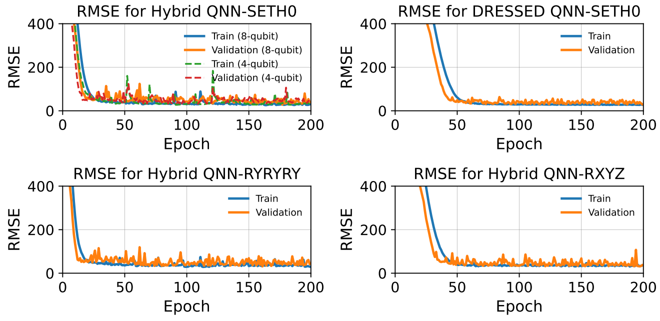


Fig. 3: RMSE for Hybrid quantum circuits

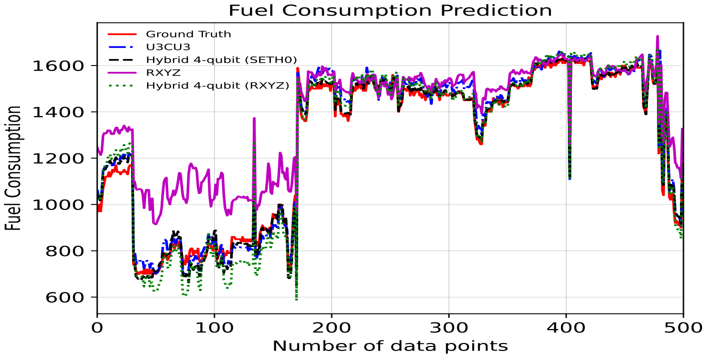


Fig. 4: Fuel Consumption Prediction.

4. SIMULATION AND RESULT DISCUSSION

Simulations for this task were carried out on a system equipped with an Intel(R) Xeon(R) Gold 6246 CPU @ 3.30GHz and an NVIDIA GeForce RTX 2080 GPU, both running at 3.30GHz. There are five distinct PQC architectures with varying layer designs available in the TorchQuantum framework [18]. As shown in Fig. 1, the encoders for all architectures utilize only rotation-X gates. Fig. 1(b) represents the RYRYRY layer, which consists solely of $R_Y(\theta)$ gates without any entanglement. It is important to note that the dotted blocks can be repeated; however, in this study, we limited the repetitions to two blocks throughout. Fig.1(b) is an architecture that with $R_X(\theta)$ $R_Y(\theta)$, and $R_Z(\theta)$ gates and following with circular entanglements. Fig.1(d) is solely based on RZZ-gate that has only one learning parameter each. Fig. 1(e) shows a more complicated layer design that with U3-gate that comprises 3 learning parameters. Moreover, entagmglement is worked against another U3-gate as well. Finally, the last figure show a special design that engtangled the qubits with RZZ-gate and later each qubit is rotated by a $R_Y(\theta)$ gate. Fig. 1(c) illustrates an architecture that incorporates $R_X(\theta)$ $R_Y(\theta)$, and $R_Z(\theta)$ gates, followed by circular entanglements. Fig. 1(d) is based solely on the RZZ gate, with each gate having only one learning parameter. Fig. 1(e) presents a more complex layer design featuring the U3 gate, which includes three learning parameters. Additionally, entanglement occurs between U3 gates. Lastly, the final figure shows a specialized design where qubits are entangled using an RZZ gate, followed by individual qubit rotations with an $R_Y(\theta)$ gate.

A plot of RMSE vs number of epochs for classical neural networks and pure quantum neural networks approaches are depicted in Fig.2. Using the Keras framework developed by [5], both CNNs with 20 and 8 neurons in the hidden layers produce low RMSE but exhibit underfitting, performing better on the validation set (10%) than the training set (90%). According to [5], these errors occur because of the training error is measured during each epoch, while the validation error is assessed after each epoch [19, 5]. Since data regularization is not applied during the validation phase, and the errors are measured at different points in time, a slight shift in the error plots may appear. It is important to note that conducting additional cross-validation can help further validate the accuracy of the chosen network. However, this issue is resolved when the QNN is applied. This is evident from the remaining figures for the five previously mentioned QNN architectures, where the loss rate stabilizes by the end of the iterations. The advantage of CNNs is that they achieve stable RMSE in around 50 iterations, whereas QNNs require at least 100 iterations to reach the same stability.

Figure 3 demonstrates that hybrid QNNs and heavily dressed QNNs with SETH0, RYRYRY, and RXYZ layer designs outperform pure QNNs in terms of RMSE. Additionally, hybrid-QNNs accelerate the RMSE convergence rate, reaching a level comparable to or even faster than CNNs. The 8-qubit hybrid-QNN exhibits less RMSE fluctuation compared to the 4-qubit version, as seen in the top-left figure. This is because an 8-qubit design contains much more trainable parameters than a 4-qubit design. Notably, both architectures have only one classical layer at the input. According to Table 1, the training/validation RMSEs at the end of the iterations are 29.42/38.70 for the 8-qubit model and 32.59/45.87 for the 4-qubit model, indicating that more trainable quantum parameters lead to better results. When an additional layer is added to the quantum circuit's input, as shown in the top-right figure, both training and validation RMSEs exhibit minor improvements over the single-layer dressed 8-qubit structure, reducing to 27.74 and 29.96, respectively. A key finding is that hybrid-QNNs significantly reduce RMSEs for poorly performing pure QNN architectures like RYRYRY and RXYZ. For the RYRYRY architecture, the training/validation RMSEs decrease from 87.52/112.30 to 30.19/42.58, while for RXYZ, they drop from 79.13/97.03 to 35.29/36.84.

To further demonstrate the advantage of hybrid QNNs, a regression prediction plot for the remaining test dataset is shown in Figure 4. The pure quantum architecture, RXYZ, performs poorly in the first two hundred data points compared to the ground truth. However, its performance improves significantly when hybridized with a 4-qubit dressed circuit. Both the pure U3CU3 and the 4-qubit hybrid with a SETH0 layer architecture serve as strong predictors, closely matching the ground truth. This is supported by the data in Table 1, where both models achieve very low MAPEs of 2.80% and 1.92%, respectively.

4.1. Computation Complexity Analysis

The advantages and disadvantages of CNNs and QNNs are challenging to compare directly. A common method to assess their efficacy is by examining the number of iterations and the total number of trainable parameters [12, 13, 20]. As shown in Table 1, CNNs with 20 hidden layer nodes achieve the best MAPE of 1.56% but have the highest number of trainable parameters, totaling 701. As the size or number of layers increases, the number of trainable parameters grows non-linearly. For example, when the number of hidden nodes increases from 20 to 50 or 100, the trainable parameters jump to 3,251 and 11,501, respectively. This non-linear increase does not occur in QNN circuits. The Hybrid 4-qubit and 8-qubit architectures with SETH0 layers contain only 64 and 212 trainable parameters, respectively, compared to the best CNN configuration. These values represent only 9.1% and 30.2% of the trainable parameters in the CNN with 20 hidden nodes per layer, yet their MAPEs are nearly-omparable. It's important to note that, when viewed from a different perspective, the Hybrid 4-qubit SETH0 architecture might be considered the best among the QNN designs. It requires fewer trainable parameters and only has a quantum circuit depth of eight. Increasing the quantum circuit depth could introduce more noise, potentially leading to computation errors and an unreliable trained model.

5. CONCLUSION

This study introduced different types of QNNs and hybrid QNNs for fuel consumption prediction. The performance in terms of RMSE and MAPE for various QNN layer designs was thoroughly examined. Unlike CNNs, QNNs do not show signs of underfitting, which is crucial for achieving a reliable machine learning model. Both entangled and non-entangled QNN models can be enhanced through the use of either hybridized or dressed circuit techniques. In fact, QNNs offer greater design flexibility than CNNs, thanks to the strategic arrangement of quantum gates, entanglement settings, and level of quantum depth. Additionally, QNNs consistently require fewer trainable parameters than CNNs to achieve comparable results. Future research could further explore the effective dimension using the Fisher information matrix as a metric to assess the performance of both CNNs and QNNs. This would offer more detailed insights that could inform the design of more efficient QNN models for optimizing emission reduction, contributing to greener shipping.

6. REFERENCES

[1] A. Swider, Y. Wang, and E. Pedersen, “Data-driven vessel operational profile based on t-sne and hierarchical clustering,” *OCEANS 2018 MTS/IEEE Charleston*, pp. 1–7, 2018.

[2] S. Mouzakitis, C. Kontzinos, J. Tsapelas, I. Kanellou, G. Kormpakis, P. Kapsalis, and D. Askounis, *Enabling Maritime Digitalization by Extreme-Scale Analytics, AI and Digital Twins: The Vesselai Architecture*, 09 2023, pp. 246–256.

[3] F. Mauro and A. Kana, “Digital twin for ship life-cycle: A critical systematic review,” *Ocean Engineering*, vol. 269, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0029801822027627

[4] J. J. M. Hermans and A. A. Kana, “Retrofit modeling for green ships,” *International Marine Design Conference*, June 2024. [Online]. Available: https://proceedings.open.tudelft.nl/imdc24/article/view/890

[5] J. Hermans, “Retrofit modeling for green ships,” Master’s thesis, TU Delft, March 2024, report number: MT.23/24.021.M. [Online]. Available: http://resolver.tudelft.nl/uuid:09e80f1a-6968-4a57-8141-5c2539c98bb4

[6] D. Peral-García, J. Cruz-Benito, and F. J. García-Peñalvo, “Systematic literature review: Quantum machine learning and its applications,” *Computer Science Review*, vol. 51, p. 100619, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1574013724000030

[7] K. Beer and G. Müller, “Dissipative quantum generative adversarial networks,” 2021. [Online]. Available: https://arxiv.org/abs/2112.06088

[8] J. Bausch, “Recurrent quantum neural networks,” *ArXiv*, vol. abs/2006.14619, 2020. [Online]. Available: https://arxiv.org/abs/2006.14619

[9] I. Cong, S. Choi, and M. D. Lukin, “Quantum convolutional neural networks,” *Nature Physics*, vol. 15, pp. 1273 – 1278, 2018.

[10] M. P. Henderson, S. Shakya, S. Pradhan, and T. Cook, “Quanvolutional neural networks: powering image recognition with quantum circuits,” *Quantum Machine Intelligence*, vol. 2, 2019.

[11] P. Bermejo, P. Braccia, M. S. Rudolph, Z. Holmes, L. Cincio, and M. Cerezo, “Quantum convolutional neural networks are (effectively) classically simulable,” 2024. [Online]. Available: https://arxiv.org/abs/2408.12739

[12] S. Chien, D. Chieng, S. Y. Chen, C. C. Zarakovitis, H. S. Lim, and Y. Xu, “Applying hybrid quantum lstm for indoor localization based on rssi,” in *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024, pp. 131–135.

[13] S. F. Chien, H. S. Lim, M. A. Kourtis, Q. Ni, A. Zappone, and C. C. Zarakovitis, “Quantum-driven energy-efficiency optimization for next-generation communications systems,” *Energies*, vol. 14, no. 14, p. 4090, 2021.

[14] A. Abbas, D. Sutter, C. Zoufal, A. Lucchi, A. Figalli, and S. Woerner, “The power of quantum neural networks,” *Nature Computational Science*, vol. 1, no. 6, pp. 403–409, 2021.

[15] S. Y.-C. Chen, S. Yoo, and Y.-L. L. Fang, “Quantum long short-term memory,” in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 8622–8626.

[16] J. Preskill, “Quantum computing in the nisq era and beyond,” *Quantum*, vol. 2, p. 79, 2018.

[17] S. Y.-C. Chen, C.-H. H. Yang, J. Qi, P.-Y. Chen, X. Ma, and H.-S. Goan, “Variational quantum circuits for deep reinforcement learning,” *IEEE Access*, vol. 8, pp. 141 007–141 024, 2020.

[18] H. Wang, Z. Liang, J. Gu, Z. Li, Y. Ding, W. Jiang, Y. Shi, D. Z. Pan, F. T. Chong, and S. Han, “Torchquantum case study for robust quantum circuits,” in *Proceedings of the 41st IEEE/ACM International Conference on Computer-Aided Design*. IEEE, 2022, pp. 1–9.

[19] Rosebrock, “Why is my validation loss lower than my training loss?” *https://pyimagesearch. com/2019/10/14/why-is-my-validation-loss-lower-than-my-training-loss/*, 2022.

[20] S. Yen-Chi Chen, T.-C. Wei, C. Zhang, H. Yu, and S. Yoo, “Hybrid quantum-classical graph convolutional network,” *arXiv e-prints*, pp. arXiv–2101, 2021.