

Fully quantum vowel recognition

Mateusz Kucharski, and Radosław Idzikowski

Abstract—This paper presents the results of using a quantum machine learning algorithm for the task of vowel recognition. It compares computation times and accuracy results of different circuit depths in a simulated quantum environment. The paper proposes a fully quantum neural network using 4 qubits as feature maps with formant frequencies as input features. It compares the results of binary classification and the more problematic multiclass classification in quantum machine learning. The results show very high accuracy of around 97% for binary, 80% for 3 classes, 70% for 4 classes, and 50% for all 5 classes. The quantum neural networks were first pre-trained using a simulator and then evaluated on the real quantum computer Odra 5 to assess the impact of noise and hardware configuration on the performance of the trained quantum circuits.

Keywords—quantum computing; quantum neural network; vowel recognition; speech recognition, formant frequencies

I. INTRODUCTION

QUANTUM computing is a newly emerging technology that, for almost a century, remained in the realm of theoretical deliberations and only recently became a practical reality [1]. Access to quantum hardware is still limited and is provided primarily by companies such as IBM [2]. Recently, however, the Wrocław University of Science and Technology acquired an IQM Spark-class superconducting quantum computer, enabling research on quantum computing at the university [3]. The Odra 5 – the computer’s official name – has five qubits, which, although modest in number, are sufficient for executing real quantum computations. This qubit count is also compatible with the capabilities of classical simulation.

Quantum machine learning (QML) is an increasingly active area of research within quantum computing. It is a field that is, at the moment, extremely restricted due to how input data is encoded on the machine, where available feature maps treat each qubit as a single feature. Currently, the largest commercially available quantum machines are provided by IBM and have 127 qubits, meaning at most 127 numerical features can be encoded. Because of this restriction, most researchers in this field use hybrid quantum-classical neural networks, where most of the largest layers, including the input layer, are computed on classical hardware, and a quantum computer is used in one of the last smaller layers, for example, a convolutional layer. Thus, the quantum computer effectively serves as a system accelerator.

Authors are with Department of Control and Quantum Computing, Wrocław University of Science and Technology, Wrocław, Poland (e-mail: mateusz.kucharski@pwr.edu.pl, ORCID: 0000-0002-1838-5901; radoslaw.idzikowski@pwr.edu.pl, ORCID: 0000-0001-5232-1998).

The hybrid approach is currently the most popular, also because many machine learning problems use images as inputs. That said, it is still possible to construct a fully quantum neural network if the number of features is sufficiently small. One such problem is considered in this paper: the use of formant frequencies for vowel recognition in speech signals. As formant data can provide sufficient information on what vowels are being spoken, and after extraction is much more accessible than, for example, a full spectrogram image, it can be used as a feature input for speech recognition purposes.

Because the Odra 5 computer has only 5 qubits, it was decided to investigate its ability to recognize vowels with only a single set of 4 formant frequencies. This work addresses that gap by evaluating the performance of pre-trained circuits on the newly deployed Odra 5 quantum computer. This is the first investigation of vowel recognition using formant-based quantum feature maps executed on SPARK-class hardware.

The article is divided into several consecutive sections. Section I provides an introduction to QML, including the motivation, contributions, and structure of the paper. Next, Section II presents a systematic literature review, analyzing related works and the current state of knowledge on QML. Section III describes the dataset used in the study and indicates its source. Section IV forms the core of the paper, focusing on the adaptation, design, and implementation of the proposed solution. It covers the translation of classical data into quantum representations, the architecture of the proposed model, the training process, and concludes with quantum transpilation. All results, together with a description of the experimental protocol, are presented in Section V, which is divided into CPU-based experiments and evaluations performed on the quantum processing unit (QPU). Finally, the paper is concluded with a summary of findings and future work, presented in Section VI.

II. LITERATURE REVIEW

In previous work, a comprehensive review of quantum machine learning for speech acoustics was conducted [4]. During the review, a substantial gap in this niche was identified, along with several topics in need of expansion. Firstly, there are very few papers that even consider QML with speech analysis; the number of identified papers for the survey was around 40, some of which were discarded due to questionable quality. The second aspect in need of addressing is the lack of papers using real quantum hardware; out of all considered, only two used an actual quantum computer. The final important



shortcoming of both the technology and research is the lack of fully quantum networks. With one exception, all works at the time of writing use hybrid classical-quantum solutions. This is most likely due to limitations in both real quantum hardware and the simulators used by most researchers. While compact data encoding techniques such as amplitude encoding have been proposed, many practical QML implementations on NISQ devices rely on angle-based feature maps, where the number of encoded features typically scales with the number of available qubits [5]. Considering most speech and speaker recognition solutions nowadays use spectrogram images, each pixel represents a single feature value. Thus, even using 50x50, which is a very low resolution, would require 2500 qubits – ignoring circuit mapping and qubit connections. This is the aspect the authors of this paper aim to address, using formant frequencies as features for fully quantum vowel recognition.

- **Fully quantum approach** – [6]: A fully quantum neural network for speech emotion recognition was proposed, based on an optimized Equivariant Quantum Convolutional Neural Network (EQCNN) with Educational Competition Optimization. The system encoded pre-processed speech features into quantum states and classified eight emotions with high accuracy. All experiments were conducted on a quantum simulator, highlighting the feasibility of end-to-end quantum neural architectures in this domain.
- **Hybrid approaches on quantum hardware (Master’s thesis)** – [7]: This thesis explored the use of the quantum Fourier transform (QFT) for spectrogram generation in a hybrid quantum–classical speech recognition pipeline. The modular framework allowed evaluation on both simulated and real quantum devices, showing that Noisy Intermediate-Scale Quantum (NISQ) hardware can process band-limited signals but remains limited by noise and qubit capacity. While real hardware runs underlined current constraints, the study demonstrated the potential of quantum feature extraction and motivated further exploration of practical hybrid methods.
- **Hybrid emotion recognition on real hardware** – [8]: Hybrid quantum machine learning models were applied to speech emotion recognition, combining classical pre-processing with quantum-enhanced classification. A Hybrid Quantum Neural Network (HQNN) was deployed on both simulators and real IBM quantum hardware, achieving competitive accuracy despite noise. This work represents one of the first demonstrations that hybrid quantum models can outperform classical baselines while using fewer parameters and smaller datasets.
- **Hybrid multimodal sentiment analysis** – [9]: A hybrid quantum–classical neural architecture (QMLSC) was developed for multimodal sentiment classification using both speech and text inputs. Variational quantum circuits were embedded in a residual fusion structure, enabling effective interplay between quantum and classical layers. Results on benchmark datasets showed improved accuracy with fewer parameters and greater robustness

to quantum noise, underscoring the promise of hybrid models for emotion-related tasks.

Hybrid keyword recognition (Google Speech Commands) – [10]: A hybrid quantum–classical framework was applied to keyword recognition using spectrogram fusion. Speech samples from the Google Speech Commands dataset were processed into LPC and STFT spectrograms, with quantum circuits applied to the LPC features before fusion with classical counterparts. The model achieved an accuracy above 94%, demonstrating the potential of quantum-enhanced spectrogram representations for speech recognition.

Quantum machine learning, as a broader field, is also gaining interest despite the current shortcomings of the technology. The idea of QML has been appearing for some time now [11]; however, until relatively recently, it remained a theoretical concept. The wider use and application in specific areas, such as speech recognition, started around the year 2020, as simulator frameworks and real quantum hardware became more available, but access to them in narrower scientific circles developing these technologies came a few years earlier and allowed researchers to tackle QML since around 2017 [12]. It was also realized then that while quantum computers have enormous potential for computational acceleration, they are severely limited by the possible input states, as illustrated in the case of image recognition. However, while waiting for the development of more powerful machines, it is still important to further research in this area. Thus, researchers continue to examine both fully quantum machine learning using datasets with a small number of features, such as the Iris dataset, which can be run with a small number of qubits [13], and hybrid options that allow circumventing input limitations [14]. New tools being developed are also expanding the use of QML to more specific areas, like healthcare [15], finance [16], or acoustics, confirming the growing interest in this emerging technology [17]. A good summary can be found in [18], as it analyzes the shortcomings while highlighting the potential advantages of QML over classical ML. This underscores the importance of conducting research even in the NISQ era, since the tools and methods developed today may be leveraged in the future on larger quantum computers with fewer errors and more advanced error correction.

III. DATASET DESCRIPTION

The dataset used for this research was retrieved from Kaggle, where it was published as an open database by the original author [19]. It lacks a proper description on the website, nor was it published with a paper; however, upon inspection, it contains the exact data needed to conduct the experiments described in this paper. The dataset contains 1198 .wav files, split mostly evenly across 5 syllables with different vowels: ha, he, hi, ho, and hu. The files are monophonic, 16 kHz, and of varying length, depending on the length of the speaker’s utterance, ranging from 100 ms to 6 s. The data in the files is somewhat noisy, as different environmental sounds can be heard in the background; however, this should not be a problem for most speech recognition systems. For

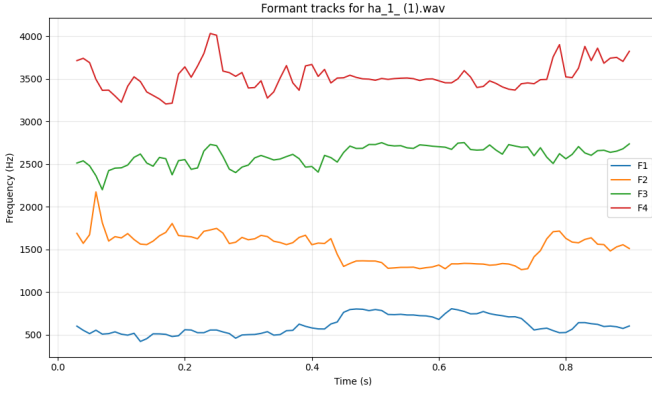


Fig. 1. Example of formant series in a “ha” sound extracted using Praat

the vowel recognition network proposed in this paper, the data were prepared as follows. Using Praat’s Python library `parselmouth` [20], the formant data for each file were extracted. Due to the varied file lengths, the middle 5 formant sequences for each file were extracted, each treated as a separate entry for the network’s input. An example of a formant frequency series is shown in Fig. 1. Formant extraction was performed using Praat’s standard Burg algorithm through the `Parselmouth` Python interface, with a time step of 0.01 s and an analysis window length of 0.025 s. These parameters provided a sufficiently fine temporal resolution to capture short-term vowel dynamics while keeping spectral estimates stable. The first four formants (F1–F4) were retained, as they carry most of the information relevant for vowel identity.

Since the recordings differ in duration and do not always begin or end precisely at the vowel, the extracted formant tracks often include leading and trailing regions containing silence or background noise. Although Praat’s algorithm is designed to avoid spurious formant detection in non-speech regions, it may still produce unreliable values when applied to noise. To mitigate this issue and ensure a more consistent representation of the vowel nucleus across all recordings, only the middle five formant frames per file were selected. This choice reduced the influence of onset and offset variability while providing a uniform input structure for the network. The number of formant frequencies saved was 4, yielding 5960 entries with 4 features each, allowing for a feature set that can be input to the Odra 5 quantum computer.

IV. QUANTUM NEURAL NETWORK

A fully quantum neural network was chosen, meaning that the entire network is represented as a quantum circuit (ansatz), which is processed using a simulator and, when available, on a real quantum computer. An example quantum circuit is shown in Figure 2; for readability, it has been divided into three sections by barriers. When executing the ansatz on a QPU, it is advisable to remove barriers, as they can affect the quantum circuit’s execution time. The first section, known as the feature map, is responsible for translating classical data into quantum data. The second section contains the network elements responsible for “decision-making”. The final

section performs the qubit measurements. The entire design and implementation were carried out using the popular Qiskit framework.

A. Data translation

A crucial step is translating classical data into quantum data. In classic machine learning, feature values are often normalized to the range $[0, 1]$. In this case, however, the data must be normalized to the interval $[0, 2\pi]$, since the input to the circuit consists of Z-gates, which act as rotation angles for specific qubits to set their states appropriately. It is clearly visible here that each qubit must be assigned exactly one feature, establishing a direct correspondence between the dimensionality of the input data and the number of qubits used. The two most popular feature maps considered were `ZFeatureMap` and `ZZFeatureMap`, both available in the Qiskit Python library. The main difference between them is that `ZFeatureMap` simply encodes input values into single-qubit Z-gates, whereas `ZZFeatureMap` introduces entanglement between qubits. According to the literature, `ZFeatureMap` (illustrated in the example ansatz in Fig. 2) typically performs better when there is little or no correlation between features, a situation that is relatively rare. In preliminary experiments, `ZZFeatureMap` (shown in Fig. 3) indeed performed significantly better; therefore, only this approach was ultimately included in the results.

B. Architecture

The ansatz used in this study was the `RealAmplitudes`¹ ansatz provided by Qiskit [21]. The research team plans to conduct future work to develop a dedicated ansatz tailored for machine learning tasks; however, this will require additional testing and validation. For this reason, the authors chose to employ the widely used, well-established ansatz available in quantum libraries. The `RealAmplitudes` ansatz is a parameterized quantum circuit commonly applied in many quantum algorithms.

It consists of single-qubit R_y rotation gates combined with CNOT (CX) entanglements. In this configuration, the qubits are entangled in a cascaded manner, which poses no issue for a simulator. On real quantum hardware, qubit connectivity is often limited—after transpilation, this can significantly increase the circuit depth and degrade its stability. In the machine learning task, the rotation angles θ of the R_y gates are treated as the network’s trainable parameters.

In this study, a single repetition of the `RealAmplitudes` ansatz is treated as one layer. However, this should not be confused with layers in classical machine learning—here, each additional layer contributes only four trainable parameters. Consequently, constructing an adequately expressive quantum network requires a significantly greater number of layers compared to a classical neural network. An example circuit combining a `ZFeatureMap` with two `RealAmplitudes` layers (in this case, a total of 12 trainable parameters for 4 features

¹<https://quantum.cloud.ibm.com/docs/en/api/qiskit/qiskit.circuit.library.RealAmplitudes>

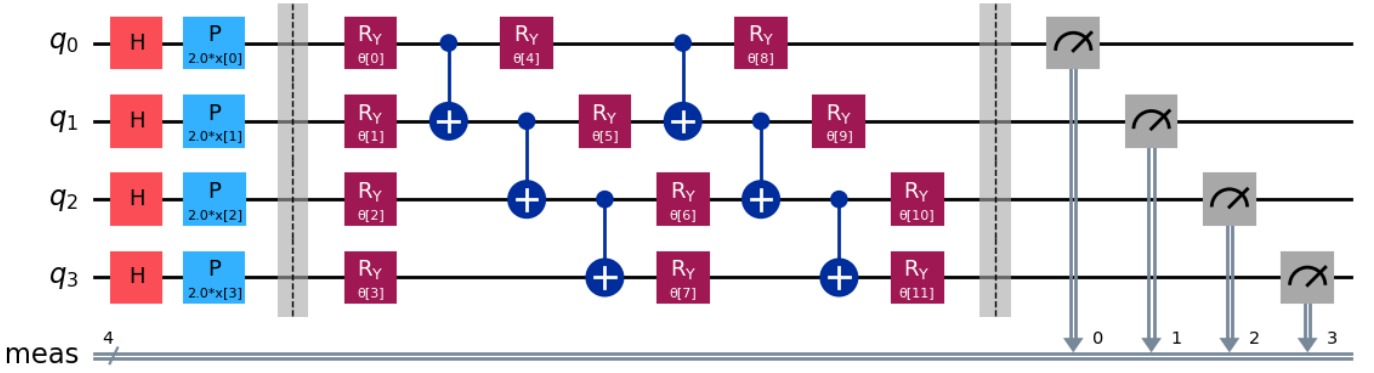


Fig. 2. Full quantum circuit containing a ZFeatureMap, two layers of RealAmplitudes, and measurement

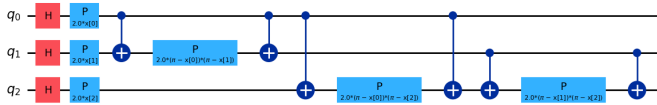


Fig. 3. Example of a ZZFeatureMap for 3 qubits

and 2 layers: 4 base parameters plus 2×4 for each layer), followed by state measurement, is shown in Figure 2.

C. Training process

During the network training experiments, the team intended to compare multiple available optimizers but encountered compatibility issues with the optimizers in existing libraries. The available optimizers were unable to train the network, either stopping after a single iteration or continuing but not training at all. Because of this, the team implemented two optimizers from scratch – a standard gradient descent and the Adam optimizer.

For the purpose of this research, multiple experiments were conducted with varying numbers of layers, depending on task complexity, most commonly 10, 25, 50, 100, 150, and 200. Cross-entropy loss was adopted as the loss function for all training procedures.

D. Compilation

When running quantum circuits on real quantum computers, an essential step is the compilation of the ansatz, known as transpilation. This process adapts the ansatz to the specific architecture of a quantum computer, as each device has its own dedicated set of supported gates. The most well-known and widely used method for this purpose is the SABRE algorithm [22], which is the default in the Qiskit framework. Despite its greedy nature, SABRE consistently delivers high-quality results.

The greatest challenge during transpilation is the lack of direct qubit connections. In such cases, qubits are typically reordered, or additional qubits are used as intermediaries. In our example circuit from Fig. 2 (a very simple ZFeatureMap with only two RealAmplitude layers), the initial circuit depth is 7 gates, but after transpilation, it increases to 24 (with

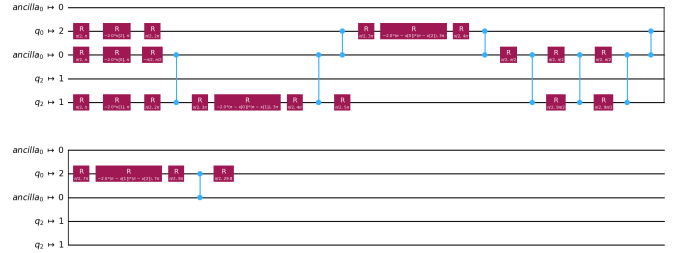


Fig. 4. Transpiled ZZFeatureMap for 3 qubits

qubit number 2 being the most heavily loaded). The transpiled version of the same circuit is shown in Fig. 5. For this example, we included ZFeatureMap data for transpilation, which is much simpler than the ZZFeatureMap commonly used in experiments. Fig. 4 shows the latter after transpilation.

Importantly, every additional gate in a quantum circuit introduces extra noise, which increases the error in the circuit's output. Moreover, two-qubit gates, such as CNOT, introduce up to 10 times more noise than single-qubit gates. Therefore, qubit-swapping operations are very costly, but unfortunately necessary given the sparse connectivity architecture of most quantum devices.

V. EXPERIMENTS

There are many parameters that could be tested for their influence on the training process of the quantum neural network, but due to long training times, some were postponed for future research. Among the most important were the number of iterations, layers, and classes. The number of iterations is important: too few will prevent the training process from finishing, and too many will take significant computational time. After several tests, it was decided to run all experiments with 250 iterations, as it was observed to be the optimal stopping point – early tests ran as long as 500 and even 1000 iterations did not yield results better than 250. The number of layers varied from 10 to 200. This is probably the most important parameter in this research, as increasing the number of layers adds quantum gates to the circuit, thereby prolonging computation time and potentially increasing quantum noise in real hardware. This research was conducted using a quantum

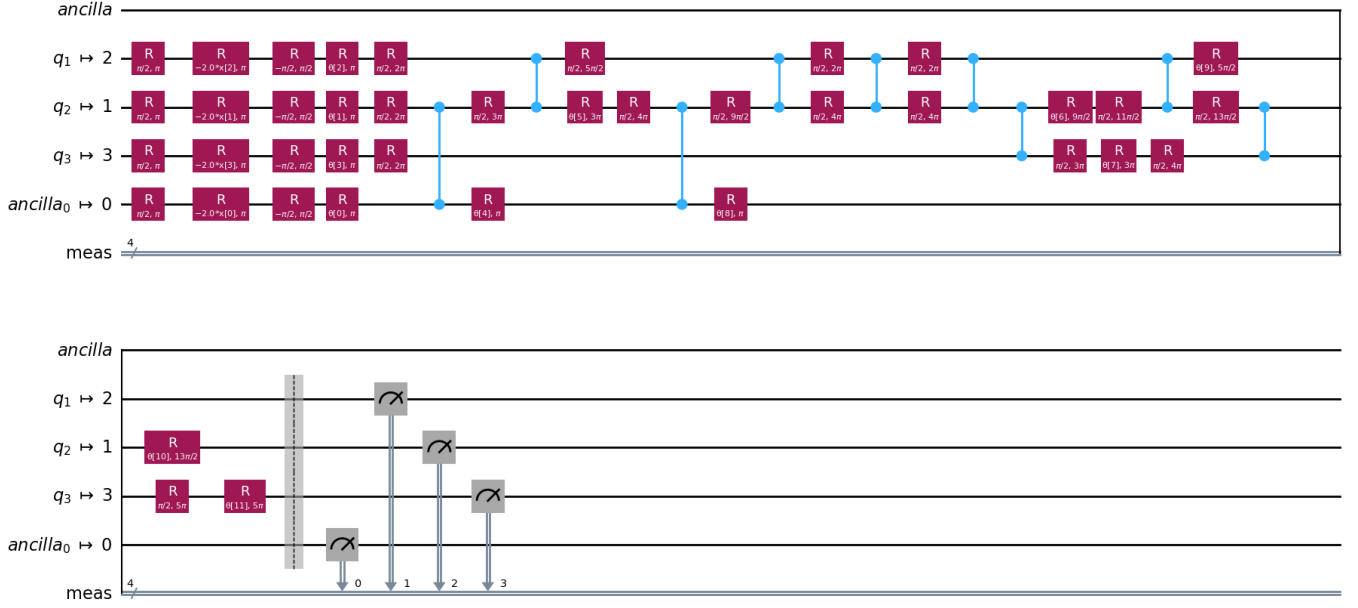


Fig. 5. Transpiled example circuit from Fig. 2, adapted for execution on the Odra 5 quantum computer.

simulator; when running such circuits on real hardware, they would likely need to be transpiled, further increasing their size. The experiments were conducted with 2, 3, 4, and all 5 classes. Since most research focuses on binary classification problems—owing to the typically poor performance of multiclass cases—it was decided to evaluate the extent of the performance drop. For the training procedure, the Adam optimizer was used, while cross-entropy was adopted as the loss function.

The experiments for this research were carried out using both a simulator without noise model (SamplerV2 from the qiskit_aer.primitives package) on a desktop computer with an Intel i7-8665U, 1.90GHz, 2112 MHz CPU, and 16 GB RAM, as well as the Odra 5 QPU. Executing the circuits on both the simulator and the real quantum computer required a specific software environment, namely Python 3.11 with the following package versions: iqm-client 29.14.0, iqm-data-definitions 2.15, iqm-exa-common 26.33.0, iqm-station-control-client 9.8.0, qiskit 1.2.4, qiskit-aer 0.15.1, and dotenv 0.9.9.

A. Results

During the experiments, three metrics were checked: accuracy, loss, and training time. Final accuracy results for all setups on the testing dataset are presented in Fig. 6. Although the initial small sizes of the circuit yield very low accuracy or even return outputs close to random (the accuracy of a 5-class network at 10 layers is around 20%), the values rise fairly quickly to very good (97% for binary classification) or acceptable levels. The network also seems to reach its maximum potential around 50 layers, and adding more layers does not increase accuracy or decrease loss. The outcome, however, is very dependent on the number of classes: binary

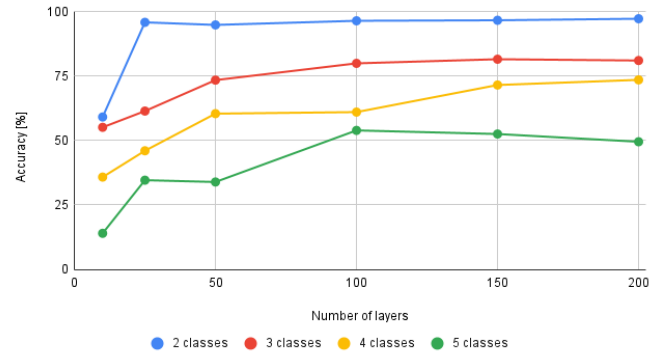


Fig. 6. Classification accuracy depending on the number of layers in QNN for different numbers of classes, obtained using the simulator.

classification works very well, and the achieved accuracy decreases as more classes are added.

Another parameter measured during this research was the time required for the training process. As mentioned previously, the simulations were run on a desktop computer, demonstrating that it is possible to simulate quantum neural networks without powerful computing devices. The times for all training processes are presented in Fig. 7. They are presented for all training combinations of layer and class numbers. The results show an exponential increase in time as the number of layers increases, and while there is a slight deviation at the longest-running circuits with 200 layers (perhaps caused by different processes running on the computer at the time), the number of classes does not seem to affect the time. This behavior occurs because, regardless of the number of classes, the number of

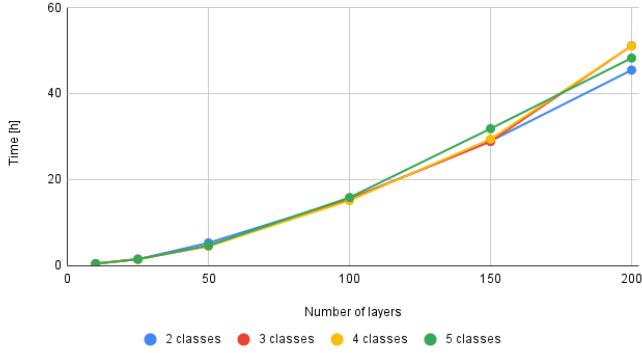


Fig. 7. Training time of the QNN on the simulator depending on the number of layers, for different numbers of classes.

trainable parameters remains unchanged, as it depends only on the number of features and the number of layers.

B. Odra 5 deployment

As the Odra 5 computer became available towards the end of this research, the authors decided to test the circuits they had created on real hardware. Unfortunately, at this point, access to the computer is very limited by the queue system under active development, and ensuring uninterrupted calculations for long periods is hard. The computer also requires daily calibration, which should be kept in mind as well. Because of this, it was decided to run tests only with circuits pre-trained by the simulator. There are several problems with this approach, such as the available gates, the topology, the need for transpilation, and the fact that it is an NISQ machine itself. The authors plan to address some of these problems in future work.

TABLE I. Results and associated errors pertain to circuits trained using a simulator; the final testing was conducted on the Odra 5 quantum computer

No. Classes	25 layers		50 layers		100 layers	
	accuracy	error	accuracy	error	accuracy	error
2 classes	53.80	± 2.04	51.91	± 1.80	48.49	± 2.22
3 classes	34.01	± 3.80	36.44	± 0.88	36.27	± 0.72
4 classes	26.09	± 1.15	25.03	± 0.89	25.00	± 1.47
5 classes	22.18	± 0.22	21.93	± 0.56	21.73	± 0.48

The results obtained from the quantum computer were collected and are presented in Table I. All values represent averages over 10 measurements, with 1024 shots used to stabilize the results. The standard deviation of the measured sample was used as the error. As shown, the results differ noticeably from those obtained with the simulator. The main reasons include the simulator's failure to account for noise, differences in topology, and the limited set of available gates.

As expected, the performance of the pre-trained circuits executed on the real quantum computer was significantly worse than on the simulator. In the two-class (binary) classification task, accuracy dropped from around 95% to roughly 50%—essentially halving—regardless of the number of layers.

While the simulator clearly demonstrated increasing accuracy with the addition of more layers, on the real quantum device, using a circuit not optimized for its architecture, the accuracy remained nearly constant across different depths. This is because, after transpilation, adding more layers compounded the noise. A similar pattern was observed for the 2-, 3-, 4-, and 5-class problems, where the results were very close to those of a random classifier, yielding approximately 50%, 33.3%, 25%, and 20%, respectively. This performance degradation is likely due to the excessive depth of the investigated circuits, combined with the fact that a noise model was not incorporated into the simulator during the training phase.

Nevertheless, it is worth noting that the circuits behaved relatively stably: in most cases, the standard deviation was small, with the worst case being ± 4.80 for the three-class problem with 25 layers. In one trial, the highest accuracy achieved was approximately 79% for the two-class problem with 25 layers.

The Odra 5 is an IQM SPARK 5 quantum computer [3], consisting of only five qubits. Among them, only the centrally positioned qubit QB3 is connected to all others via tunable couplers. Unfortunately, the circuits we designed contained connections between qubits that are not directly linked in the device's topology. As a result, after transpilation [23], numerous additional SWAP operations were required to emulate these connections. This significantly increased the depth of the quantum circuits, an effect further amplified by the need to substitute unsupported gates with available ones.

VI. CONCLUSIONS

This paper presents a fully quantum neural network for vowel recognition that uses formant frequencies as input features, enabling the construction of low-qubit quantum circuits. This approach not only enables simulation on regular desktop computers, but also prepares for deployment on real quantum hardware, as such hardware is still limited in size.

Using the simulator, very promising results were achieved—around 95% accuracy for binary classification, even with networks of only 25 layers. Circuits with just 10 layers proved far too shallow to yield satisfactory performance, regardless of the number of classes. A clear increase in accuracy was observed as more layers were added, up to about 100, after which performance generally stabilized. Subsequently, due to hardware access limitations, only the pre-trained circuits were evaluated on the real quantum computer. However, because the simulator did not account for noise, actual connectivity topology, and the restricted set of available gates, the performance on the hardware dropped by nearly half.

As future work, we plan to address these challenges by designing a dedicated quantum circuit tailored to the architecture of SPARK 5-type quantum computers, such as Odra 5. Unfortunately, exclusive access to the hardware and the elimination of communication delays are not feasible, making full training directly on the quantum computer unlikely. Therefore, we intend to incorporate the actual SPARK 5 architecture and its supported gates into the simulator. Pre-training of the

new circuits will be carried out in this enhanced environment, followed by fine-tuning the quantum neural network on the quantum hardware using a limited number of iterations prior to evaluation.

After addressing these issues, we can again focus on research into speech, as it is both a little-explored topic in QML applications and feasible given the importance of formant frequencies. The first topic to explore is a fully quantum formant-based word recognition – although it will be hard to implement on a computer like Odra 5. The second is a hybrid quantum-classical network that combines a classically trained spectrogram-based part with a formant-based quantum part, in the style of one of the authors' previous works [24].

REFERENCES

- [1] W. Bożejko, S. Trotskiy, M. Uchroński, and M. Wodecki, "Optimizing two-machine scheduling in flexible manufacturing systems using autonomous ai and quantum computing," *Neurocomputing*, p. 132067, 2025.
- [2] "Ibm quantum," <https://quantum.ibm.com/>, 2021.
- [3] J. Rönkkö, O. Ahonen, V. Bergholm, A. Calzona, A. Geresdi, H. Heimonen, J. Heinsoo, V. Milchakov, S. Pogorzalek, M. Sarsby *et al.*, "On-premises superconducting quantum computer for education and research," *EPJ Quantum technology*, vol. 11, no. 1, p. 32, 2024.
- [4] M. A. Kucharski, "A survey on quantum machine learning in speech acoustics," in *2025 IEEE 25th International Symposium on Computational Intelligence and Informatics (CINTI)*. IEEE, 2025, pp. 235–240.
- [5] R. Idzikowski, M. A. Kucharski, K. Pempera, and M. Jaroszczuk, "A survey on quantum machine learning applications in medicine and healthcare," *Applied Sciences*, vol. 16, no. 3, p. 1630, 2026.
- [6] G. Balachandran, S. Ranjith, G. Jagan, and T. Chenthil, "Advanced speech emotion recognition utilizing optimized equivariant quantum convolutional neural network for accurate emotional state classification," *Knowledge-Based Systems*, vol. 316, p. 113414, 2025.
- [7] M. Strobl, "Quantum computing assisted speech processing," 2022, mSc Thesis.
- [8] S. Mittal, Y. Chand, M. Kumar, and N. K. Kundu, "Hybrid quantum machine learning based human speech emotion recognition," in *2025 IEEE International Conference on Acoustics, Speech, and Signal Processing Workshops (ICASSPW)*. IEEE, 2025, pp. 1–5.
- [9] Y. Li, Y. Qu, R.-G. Zhou, and J. Zhang, "Qmlsc: A quantum multimodal learning model for sentiment classification," *Information Fusion*, vol. 120, p. 103049, 2025.
- [10] L. Miller, T. K. Patel, G. Uehara, S. Naik, and A. Spanias, "Quantum machine learning and spectrogram fusion for speech recognition," in *2024 58th Asilomar Conference on Signals, Systems, and Computers*. IEEE, 2024, pp. 674–678.
- [11] S. Gambis, "Machine learning in a quantum world," in *Advanced in Artificial Intelligence, Proceedings of the 19th Conference of the Canadian Society for Computational Studies of Intelligence, Canadian AI*, 2006, pp. 7–9.
- [12] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, "Quantum machine learning," *Nature*, vol. 549, no. 7671, pp. 195–202, 2017.
- [13] M. Zhang, X. Chen, and Z. Liu, "Quantum neural network image three-classification model based on the iris dataset," in *2024 5th International Conference on Computer Engineering and Application (ICCEA)*. IEEE, 2024, pp. 841–846.
- [14] M. Alam and S. Ghosh, "Deepqmlp: A scalable quantum-classical hybrid deep neural network architecture for classification," in *2022 35th International Conference on VLSI Design and 2022 21st International Conference on Embedded Systems (VLSID)*. IEEE, 2022, pp. 275–280.
- [15] T. O. Fatunmbi, "Integrating quantum neural networks with machine learning algorithms for optimizing healthcare diagnostics and treatment outcomes," *World Journal of Advanced Research and Reviews*, 2023.
- [16] D. Emmanoulopoulos and S. Dimoska, "Quantum machine learning in finance: Time series forecasting. arxiv. 2022," *arXiv preprint arXiv:2202.00599*, 2022.
- [17] D. Garcia, J. Cruz-Benito, and F. Garcia-Penalvo, "Systematic literature review: Quantum machine learning and its applications (2022)," *arXiv preprint arXiv:2201.04093*, 2022.
- [18] M. Cerezo, G. Verdon, H.-Y. Huang, L. Cincio, and P. J. Coles, "Challenges and opportunities in quantum machine learning," *Nature computational science*, vol. 2, no. 9, pp. 567–576, 2022.
- [19] D. A. Rubiano, "Dataset of vowels," <https://www.kaggle.com/datasets/darubiano57/dataset-of-vowels>, 2022, accessed: 2025-07-28.
- [20] P. Boersma, "Praat, a system for doing phonetics by computer," *Glott. Int.*, vol. 5, no. 9, pp. 341–345, 2001.
- [21] A. Javadi-Abhari, M. Treinish, K. Krsulich, C. J. Wood, J. Lishman, J. Gacon, S. Martiel, P. D. Nation, L. S. Bishop, A. W. Cross, B. R. Johnson, and J. M. Gambetta, "Quantum computing with Qiskit," 2024.
- [22] F. Hua, M. Wang, G. Li, B. Peng, C. Liu, M. Zheng, S. Stein, Y. Ding, E. Z. Zhang, T. Humble *et al.*, "Qasmtrans: A qasm quantum transpiler framework for nisq devices," in *Proceedings of the SC'23 Workshops of the International Conference on High Performance Computing, Network, Storage, and Analysis*, 2023, pp. 1468–1477.
- [23] A. Gnatowski and T. Niżyński, "Quantum circuits compilation for superconducting quantum processing units: A systematic mapping study," in *International Conference on Intelligent Systems in Production Engineering and Maintenance*. Springer, 2025, pp. 269–284.
- [24] M. Kucharski and E. Skubalska-Rafajłowicz, "Speech recognition using spectrogram images and formant data with neural networks," in *International Conference on Dependability and Complex Systems*. Springer, 2025, pp. 129–137.