Quantum Machine Learning for Land Cover Classification: Evaluating Variational Quantum Classifiers Using Sentinel-2 Imagery

1st Carlos Henrique de Moura Silva
Geoanalytical Solutions Division
Serpro – Serviço Federal de Processamento de Dados
São Paulo, Brazil
carlos-henrique.silva@serpro.gov.br

***Abstract*—The identification of soybean and forest areas using remote sensing data is of great relevance for environmental monitoring and agricultural management. This study evaluates the feasibility of classifying these land cover types using machine learning techniques in a quantum computing environment. The proposed methodology is tested in a quantum computing simulator to assess its classification performance and potential advantages over classical approaches. Preliminary results suggest that quantum machine learning could offer promising improvements in land cover classification, particularly in complex and high-dimensional datasets. This study contributes to the growing field of quantum remote sensing, providing insights into the applicability of quantum computing for large-scale land use classification task.**

Keywords— variational quantum classifiers, sentinel-2 imagery, qiskit, remote sensing classification

I. INTRODUCTION

Land cover classification using remote sensing data is fundamental for environmental monitoring, agricultural management, and tracking land-use changes [3]. Classical machine learning models, such as Support Vector Machines (SVM) and Random Forests, have demonstrated robust accuracy for classifying vegetation types based on satellite-derived spectral data [1]. Nevertheless, the growing size and complexity of remote sensing datasets pose challenges related to computational efficiency and scalability [6].

Quantum computing has emerged as a promising approach to overcome these limitations, offering significant computational advantages in specific tasks, such as optimization and classification [2]. Variational Quantum Algorithms (VQAs), particularly the Variational Quantum Classifier (VQC), integrate quantum feature mapping with parameterized quantum circuits and have potential to enhance classification accuracy and efficiency in processing complex, high-dimensional remote sensing data [4].

This study evaluates the potential of quantum machine learning, specifically VQC, for classifying soybean and forest areas using Sentinel-2 imagery. We utilize spectral bands B2 (blue), B3 (green), and B4 (red)—all with 10-meter spatial resolution—alongside annual mean Normalized Difference Vegetation Index (NDVI), known for its effectiveness in assessing plant health and canopy structure.

Our research involves training and evaluating a Variational Quantum Classifier implemented using Qiskit's quantum simulator, comparing its classification accuracy with that of a conventional neural network model. The goal is to determine the practicality and effectiveness of quantum methods for remote sensing classification tasks.

II. MATERIAL AND METHODS

The study utilized Sentinel-2 satellite imagery from the COPERNICUS/S2_SR collection, focusing on regions in Rio Grande do Sul, Brazil. Representative sampling points for soybean fields and forest formations were manually selected.

Imagery from January 1, 2020, to December 31, 2020, was obtained via Google Earth Engine (GEE), with cloud-contaminated pixels removed using the QA60 quality band mask. A median composite image was generated using spectral bands B2 (blue), B3 (green), B4 (red) and NDVI, reducing temporal noise. Histogram and kernel density estimation (KDE) plots highlighted distinct spectral differences between soybean and forest classes; forests demonstrated low reflectance in visible bands and high NDVI values, whereas soybean fields displayed the opposite characteristics (Figure 1).

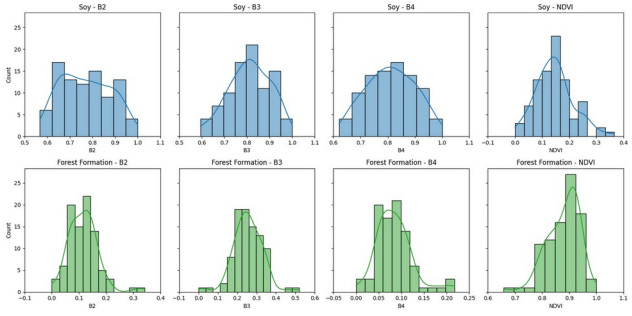

Figure 1. Histogram and kernel density estimation (KDE)

Sampling points for both classes were converted into a FeatureCollection in GEE, labeled categorically (1 for soybean, 2 for forest), and spectral data were extracted at a 10-meter spatial resolution. The dataset, exported as CSV, underwent Min-Max normalization in Python to standardize feature scales and was divided into stratified training (80%) and testing (20%) sets.

A classical neural network model was implemented using Python and TensorFlow/Keras. This model was trained using

normalized spectral features (B2, B3, B4, NDVI) and labels adjusted to a binary scale (0 for soybean, 1 for forest). The neural network comprised two hidden layers (64 and 32 neurons respectively, using ReLU activation) and an output layer with two neurons (softmax activation). Training was conducted over 50 epochs with batch sizes of 32, employing the Adam optimizer and sparse categorical cross-entropy loss. Predictions from the trained neural network were exported, reclassified into original categorical values (1 and 2), and saved in a CSV file for subsequent spatial analysis [7].

Now we describe the implementation of a Variational Quantum Classifier (VQC) using Qiskit's quantum simulator [8]. To encode the spectral features into quantum states, we employed a *ZZFeatureMap* circuit [9], which leverages key quantum phenomena—such as superposition, entanglement, and interference—to capture complex, nonlinear relationships within remote sensing data more effectively than many classical methods might.

We consider four qubits, {q0,q1,q2,q3}, each corresponding to one spectral feature (B2, B3, B4, and NDVI, respectively). Let us denote the normalized spectral feature vector by

$$\mathbf{x} = (x_0, x_1, x_2, x_3)$$

where each xi is typically scaled to a range [0,π] (or another interval) prior to encoding.

To initialize the circuit, each qubit is placed into an equal superposition state via a Hadamard gate H. In Dirac notation, the Hadamard transform on a single qubit is given by

$$H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$

Applying H to all four qubits produces an initial state:

$$|\psi_0\rangle = H^{\otimes 4}|0,0,0,0\rangle$$

This step ensures that each qubit has equal amplitude of being in |0⟩ and |1⟩, thereby enabling the subsequent encoding of feature-dependent phases. To embed each spectral feature xi into the quantum state, we apply a phase (or P) gate on each qubit, which imparts a rotation about the Z-axis. Formally, a P-gate with angle θ acts on a single qubit as:

$$P(\theta) = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{pmatrix}$$

For data encoding, we set

$$\theta_i = 2 \cdot x_i,$$

So that each qubit qi undergoes a phase rotation P(2xi). This corresponds to the purple boxes labeled "P(2.0×x[i])" in the circuit diagram. The resulting state after these single-qubit rotations is:

$$|\psi_1\rangle = \left(\prod_{i=0}^{3} P_i(2\,x_i)\right)|\psi_0\rangle$$

Here, Pi(2xi) denotes the phase gate on qubit qi with rotation angle 2xi.

A critical element of the *ZZFeatureMap* is the introduction of controlled interactions between pairs of qubits. Specifically, *controlled-ZZ* gates embody pairwise correlations among the features by adding an additional phase when both qubits are in the |1⟩ state. In practice, this is often implemented as a sequence of two-qubit gates, such as a CNOT–RZ(φ)–CNOT construction or via native hardware-efficient instructions, depending on the backend [10].

For a pair of features (xi,xj), the parameterized angle can be expressed as a function of the difference (π−xi)(π−xj), yielding a rotation:

$$\varphi_{ij} = 2\,(\pi - x_i)(\pi - x_j)$$

These controlled phases augment the overall quantum state with higher-order feature dependencies, enabling the classifier to learn more complex decision boundaries. Symbolically, we may write the multi-qubit interaction operator as:

$$U_{ZZ}(\mathbf{x}) = \prod_{i<j} \exp\left(-\tfrac{i}{2}\,\varphi_{ij}\,\sigma_z^{(i)} \otimes \sigma_z^{(j)}\right)$$

where σz(i) acts as the Pauli-Z operator on qubit i (identity on others). After these controlled-ZZ interactions, the feature-encoded state becomes:

$$|\psi(\mathbf{x})\rangle = U_{ZZ}(\mathbf{x})\,|\psi_1\rangle$$

In (Figure 2), the purple gates labeled "P(2.0×(π−x[i])×(π−x[j]))" represent this step, emphasizing that the effective phase depends on both xi and xj.

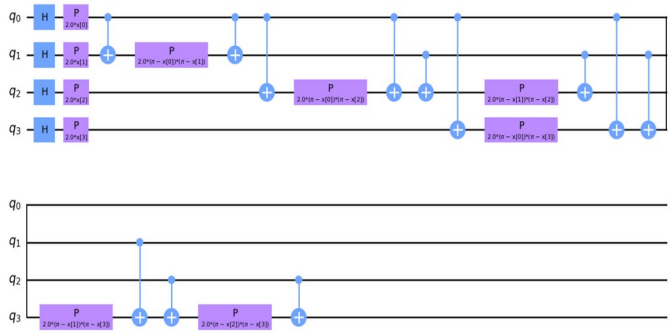

Figure 2. ZZFeatureMap

Once the data is encoded via the *ZZFeatureMap*, additional layers of parameterized single- and two-qubit gates (the "variational form") are appended to the circuit. These variational parameters are iteratively optimized based on a cost function—such as cross-entropy or mean squared error—calculated from measurement outcomes. The optimization typically relies on gradient-based methods or gradient-free techniques like COBYLA [11].

Formally, if θ denotes the set of trainable parameters in the variational ansatz, and |ψ(x;θ)⟩ represents the final state of the circuit, then the VQC outputs a measurement probability distribution:

$$p_\theta(y \mid \mathbf{x}) = \left|\langle y \mid \psi(\mathbf{x};\theta)\rangle\right|^2$$

Where y indexes the computational basis states. A classical post-processing layer then maps this probability

distribution onto predicted classes or regression values, depending on the nature of the problem.

Before embedding features into the quantum circuit, it is essential to rescale the raw spectral data so that xi lies within a practical interval. Common ranges include [0,π] or [0,2π], ensuring that the phase rotations remain within hardware-friendly bounds. The controlled ZZ interactions enable the circuit to learn joint dependencies between features. However, this can make the model more sensitive to noise on quantum hardware. Careful calibration or error mitigation techniques may be required in real devices [12].

Increasing the number of qubits or the number of entangling layers can enhance the representational power of the VQC but may also lead to higher gate errors and reduced fidelity in near-term quantum devices. A balance between depth and expressivity is often necessary to achieve optimal performance [7]. The results in this study were obtained via a quantum simulator, which provides an idealized environment. In practical scenarios on noisy quantum hardware, performance can differ due to decoherence, readout errors, and limited connectivity of qubits.

We allocate four qubits, {q0,q1,q2,q3}, each corresponding to one of the spectral features under consideration. This one-to-one mapping between qubits and features is a common practice in quantum machine learning methods, facilitating straightforward interpretation and manipulation of feature-encoded states

The *RealAmplitudes* ansatz comprises repeated layers—denoted by reps=3—in which each qubit undergoes a single-qubit rotation around the Y-axis. In matrix form, the rotation gate Ry(θ) is defined as:

$$R_y(\theta) = \begin{pmatrix} \cos\frac{\theta}{2} & -\sin\frac{\theta}{2} \\ \sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{pmatrix}$$

Within each layer, every qubit qi receives a unique rotation angle, θ[i]. Because θ[i] is treated as an independent parameter, the circuit possesses sufficient flexibility to capture intricate nonlinear dependencies among the input features. Formally, one layer of four qubits can be expressed as:

$$\bigotimes_{i=0}^{3} R_y(\theta_i)\,|\psi_{l-1}\rangle \longrightarrow |\psi_l\rangle$$

where $|\psi l-1\rangle$ and $|\psi l\rangle$ represent the states before and after the lth layer, respectively. By stacking these layers (reps=3), the circuit gains deeper expressiveness, analogous to how multiple layers in a neural network enable increasingly abstract feature representations [13].

Between each set of Ry rotations, *entangling gates* are introduced to foster correlation among the qubits. The figure (Figure 3 in the original text) illustrates how Controlled NOT (CNOT) gates are arranged such that every qubit can become entangled with every other qubit over the span of these layers.

A standard "cascading" pattern or a "ring" topology is often used to ensure thorough entanglement:

1. **CNOT(q0 → q1)**
2. **CNOT(q1 → q2)**
3. **CNOT(q2 → q3)**

(The actual sequence can vary depending on the chosen entanglement strategy.)

These CNOTs introduce nonlocal correlations, enabling the circuit to learn higher-order interactions among the features:

$$\mathrm{CNOT}(q_i \to q_j) : \alpha\,|0\rangle + \beta\,|1\rangle \ (\text{on } q_i) \implies \begin{cases} |q_j\rangle & \text{if } q_i = 0 \\ X\,|q_j\rangle & \text{if } q_i = 1 \end{cases}$$

where X denotes the Pauli-X operation (NOT gate). Over multiple layers, these interactions can significantly enrich the representational capacity of the ansatz .

The final circuit incorporates 4×reps=4×3=12 rotational parameters (plus any additional parameters if further layers or custom arrangements are used). Each parameter θ[i] is updated via the Constrained Optimization By Linear Approximation (COBYLA) algorithm [11], a gradient-free method that is particularly suitable for noisy or non-analytic objective functions—common scenarios in quantum computing experiments.

Parameters θ[i] can be initialized randomly (e.g., uniform distribution in [−π,π]) or via heuristics. After constructing the parameterized circuit, measurements yield a cost function (e.g., mean squared error or cross-entropy loss).COBYLA proposes a new set of parameters based on linear approximations of the local cost landscape.This process repeats until a stopping criterion is met (e.g., max iterations = 80) or the cost improvement becomes negligible.

Following parameter training, we evaluate the *RealAmplitudes*-based classifier on a held-out test set. The measurement outcomes on the quantum circuit are post-processed—often translated into a probability distribution over labels—which is then used to compute accuracy or other performance metrics.

It is noteworthy that while the simulations for this work were run in a noise-free or low-noise environment (such as Qiskit's statevector or QASM simulators), real hardware implementations would involve errors and decoherence effects. Techniques like error mitigation [12] or hardware-efficient ansätze [14] can help bridge the gap between idealized simulations and actual quantum processors.

By combining multi-layered Ry rotations with carefully arranged CNOT gates, the *RealAmplitudes* ansatz achieves a flexible yet powerful parameterization of the quantum state. Each repetition layer enriches the entanglement structure and expressiveness, while independent rotation angles θ[i] allow the circuit to learn nuanced patterns. With COBYLA optimization over 80 iterations, the trained model captures both linear and nonlinear relationships within the spectral data, ultimately achieving robust predictive accuracy on the test dataset (*Figure 3*).

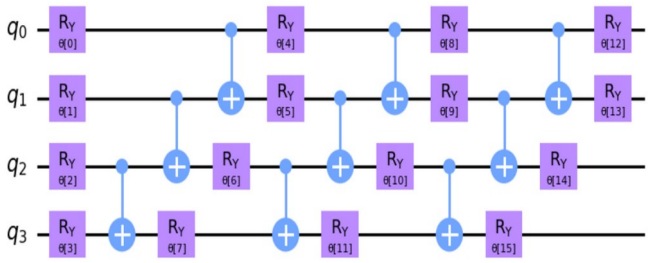

Figure 3. Ansatz

Final model predictions (quantum and classical) were spatially analyzed and visualized by rasterizing the classification outputs using GEE.

III. RESULTS AND DISCUSSION

To evaluate the performance of the classification models, an area of approximately 44.1 hectares located in the municipality of Júlio de Castilhos, Rio Grande do Sul, Brazil, was selected (Figure 4).

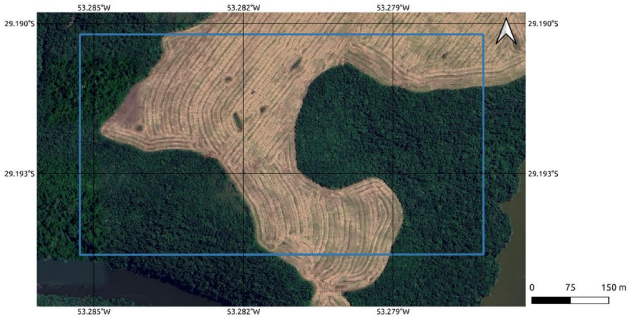

Figure 4. Area Selected

This region encompasses two distinct land cover types: forest formation and soybean cultivation. The reference data for assessing model accuracy was derived from the MapBiomas Collection 9 classification [5], which provides land cover information at a 30-meter spatial resolution, categorizing pixels as either soybean or forest formation (Figure 5).

Figure 5. MapBiomas

Two raster files with a higher spatial resolution of 10 meters were generated to represent predictions from two different classification approaches: a classical neural network (Figure 6)

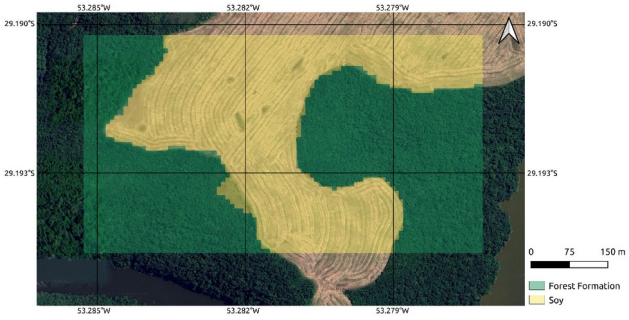

Figure 6. Classical Neural Network

Variational Quantum Classifier (VQC) (Figure 7) implemented through the Qiskit Machine Learning package, run on a quantum simulator. The accuracy of these predictive models was quantitatively assessed by comparison against the MapBiomas raster, treated as the ground truth.

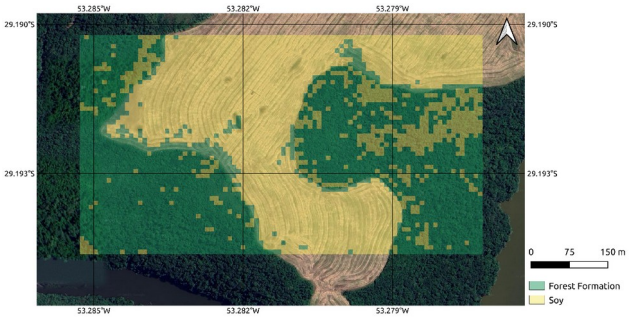

Figure 7. VQC

The classification results indicate significant differences between the classical and quantum models. The classical neural network achieved a high accuracy, correctly classifying 92.23% (1,910 pixels) of soybean areas and misclassifying only 7.77% (161 pixels) as false positives. Additionally, it demonstrated a robust performance in identifying forest formation, correctly classifying 94.70% (2,215 pixels) with only 5.30% (124 pixels) falsely identified as soybean.

Conversely, the quantum classifier (VQC) exhibited comparatively lower accuracy. For soybean predictions, the model accurately classified 76.58% (1,769 pixels) of the area, but it incorrectly classified a substantial 23.42% (541 pixels). The VQC performed better in classifying forest formation, with an accuracy of 87.57% (1,839 pixels), but still presented a considerable false positive rate of 12.43% (261 pixels) (Figure 8).

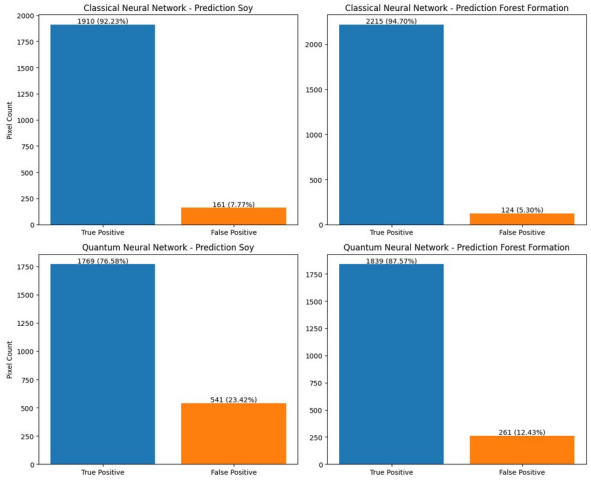

Figure 8. Classification Results

The discrepancies observed between the classical neural network and the VQC predictions underscore the challenges inherent in quantum classifiers, particularly in complex agricultural landscapes characterized by mixed land-use types. The superior performance of the classical neural network in this specific scenario might be attributed to its capability to manage spatial complexities and heterogeneity within high-resolution imagery more effectively. Nonetheless, the quantum approach provides valuable insights and indicates potential avenues for future research aimed at improving quantum machine learning techniques for environmental and agricultural applications.

These findings highlight the importance of model selection and resolution considerations in satellite imagery classification tasks, demonstrating that classical approaches remain highly effective, while quantum methods, though currently less accurate, offer promising directions for methodological advancements

IV. CONCLUSIONS

This study evaluated the potential of Variational Quantum Classifiers (VQC) compared to classical neural network models in classifying land cover types, specifically soybean fields and forest formations, using Sentinel-2 satellite imagery. While the classical neural network demonstrated superior accuracy, correctly classifying over 90% of both land cover types, the quantum classifier yielded lower accuracy, particularly in distinguishing soybean fields. The observed differences highlight current limitations of quantum classifiers in handling complex spatial and spectral variations inherent in agricultural landscapes.

Despite these limitations, the quantum approach presents promising opportunities for advancements in computational efficiency and scalability, especially as quantum computing technologies mature. Future research should focus on improving quantum classification methods by optimizing quantum circuit design, exploring alternative quantum feature encodings, and evaluating the impact of quantum noise mitigation strategies. Additionally, migrating from quantum simulations to experiments on real quantum hardware platforms represents an essential next step, allowing assessment of model robustness under realistic quantum noise and connectivity constraints. Such research will be crucial for determining the practical viability and future applications of quantum computing in environmental remote sensing and land-use classification tasks. The codes used in this article are at https://github.com/kikosmoura/quantum_simulator.

REFERENCES

[1] Belgiu, M., & Drăguţ, L. (2016). Random forest in remote sensing: A review of applications and future directions. ISPRS Journal of Photogrammetry and Remote Sensing, 114, 24-31.

[2] Biamonte, J., et al. (2017). Quantum machine learning. Nature, 549(7671), 195-202.

[3] Foody, G. M. (2002). Status of land cover classification accuracy assessment. Remote Sensing of Environment, 80(1), 185-201.

[4] Havlíček, V., et al. (2019). Supervised learning with quantum-enhanced feature spaces. Nature, 567(7747), 209-212.

[5] MapBiomas. (2022). Coleção 9 da Série Anual de Mapas de Cobertura e Uso do Solo do Brasil. MapBiomas. Retrieved from https://mapbiomas.org.

[6] Zhang, L., et al. (2016). Deep learning for remote sensing data: A technical tutorial on the state of the art. IEEE Geoscience and Remote Sensing Magazine, 4(2), 22-40.

[7] Chollet, F. (2021). Deep Learning with Python (2nd ed.). Manning Publications.

[8] H. Abraham et al., *Qiskit: An Open-source Framework for Quantum Computing*, [Online]. Available: https://qiskit.org.

[9] M. Schuld and N. Killoran, "Quantum machine learning in feature Hilbert spaces," *Phys. Rev. Lett.*, vol. 122, 040504, 2019.

[10] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*, Cambridge University Press, 2010.

[11] M. Powell, "A view of algorithms for optimization without derivatives," *Mathematics Today-Bulletin of the Institute of Mathematics and its Applications*, vol. 43, pp. 170–174, 2007.

[12] K. Temme, S. Bravyi, and J. M. Gambetta, "Error mitigation for short-depth quantum circuits," *Phys. Rev. Lett.*, vol. 119, 180509, 2017.

[13] M. Cerezo et al., "Variational quantum algorithms," *Nature Reviews Physics*, vol. 3, pp. 625–644, 2021.

[14] Chen, C., Tang, K., Zhou, Y., Yi, K. Y., Zhang, X., Zhang, X., ... & Yu, D. (2024). Hardware-Efficient Stabilization of Entanglement via Engineered Dissipation in Superconducting Circuits. arXiv preprint arXiv:2407.13321.