Quantum Machine Intelligence (2025) 7:11
https://doi.org/10.1007/s42484-024-00229-1

RESEARCH ARTICLE

Exoplanet discovery with variational quantum circuits

Alberto Regadío[1]

Received: 11 July 2024 / Accepted: 16 December 2024

Abstract
This manuscript addresses the automatic identification of exoplanets with the data from the Kepler mission using variational quantum circuits (VQC) implemented with the Qiskit library. The system accuracy was assessed across various configurations, including different numbers of qubits, feature maps, ansatz structures, and training algorithms, providing valuable insights into the performance of VQCs. The results of the VQC-based approach were compared to those obtained using a classical artificial neural network, namely a multilayer perceptron. The findings demonstrate that VQCs are a viable and promising method for processing astronomical data and discovering exoplanets, offering a potential advantage over traditional machine learning techniques.

Keywords Quantum algorithms · Exoplanets · Quantum machine learning · Neural network architectures

1 Introduction

The exploration and identification of exoplanets play a key role in the understanding of our universe. Missions such as Kepler and the Transiting Exoplanet Survey Satellite (TESS) have been decisive in the quest to discover planets beyond our solar system, which has been a significant breakthrough in the field of astrophysics.

The study of exoplanets unravels the mysteries of planetary formation and evolution and assessing the potential habitability of distant worlds. In consequence, comprehensive and effective classification of exoplanets not only satisfies scientific curiosity but also contributes to the broader exploration of our place in the cosmos.

The data received from the Kepler and TESS missions are increasingly extensive. Fortunately, the exploitation of machine learning in the search for exoplanets has revolutionized the ability of astrophysicists to analyze large volumes of data from these missions. Machine learning algorithms, including neural networks and supervised learning techniques, enable the identification of planetary transit signals in noisy data, improve exoplanet detection, and reduce false positives (Pearson et al 2017; Shallue and Vanderburg 2018; Zingales and Waldmann 2018; Dattilo et al. 2019; Jara-Maldonado et al. 2020; Perger et al. 2023). These methods can handle the vast and complex datasets generated by space telescopes, allowing for the identification of subtle signals that might be missed by traditional methods, and providing new opportunities for the study of distant planetary systems.

Quantum circuits hold great potential for improving machine learning—an approach that, to the best of our knowledge, has not yet been applied to exoplanet discovery. Among these, variational quantum circuits (VQCs) are a type of parameterized quantum circuit in which certain parameters are tuned to minimize (or maximize) a specific cost function. These circuits combine elements of quantum computing and classical optimization techniques similar to those used in classical machine learning, offering advantages in processing power and efficiency, potentially allowing for the analysis of even larger datasets and more complex models than currently possible with classical computing methods.

In this article, we propose using VQCs for the detection of exoplanets utilizing data from the Kepler mission. The contribution of this work is twofold: (a) from an astrophysics perspective, we demonstrate that exoplanet identification can be performed automatically using quantum circuits and (b) from a quantum circuit design standpoint, we analyze the impact of various modules for feature maps and ansatzes, offering insights into the design of VQCs.

✉ Alberto Regadío
alberto.regadio@uah.es

[1] Space Research Group, University of Alcalá, Plaza San Diego, s/n, Alcalá de Henares 28801, Madrid, Spain

Published online: 24 January 2025

The rest of the manuscript is structured as follows: Section 2 provides background on the Kepler mission and exoplanet identification using the transit method. Section 3 presents background information on quantum circuits, quantum machine learning, and the VQC approach. Following these theoretical backgrounds, Section 4 details the experimental approach, including the VQCs used. Section 5 describes the results obtained in the classification of exoplanets. Finally, Section 6 covers the conclusions.

2 Background in Kepler mission and exoplanet identification

NASA's Kepler mission has been instrumental in the search for and study of planets orbiting stars outside our solar system. Launched on March 7, 2009, the mission focused on a small segment of the sky in the constellation Cygnus, continuously observing more than 150,000 stars to detect small variations in their brightness. Equipped with a high-precision photometer, Kepler's primary objective was to determine the frequency of habitable planets in our galaxy. Although the original mission ended in 2013 due to mechanical failures, the telescope has remained functional since 2014 under the extended K2 mission.

The Kepler mission has been remarkably successful, identifying over 3,000 confirmed exoplanets and thousands of additional candidates. These discoveries have revolutionized our understanding of the prevalence and diversity of planetary systems, showing that planets are common in the Milky Way and that many may have conditions suitable for liquid water. Currently, the telescope continues to collect data in its extended mission, offering a clearer view of the abundance and variety of exoplanets.

The main method used by Kepler to detect exoplanets is the transit method. This method involves observing the brightness of a star over long periods. If a planet passes (or transits) in front of its star from our perspective, it blocks a small fraction of the starlight, causing a periodic decrease in the brightness of the star (see example in Fig. 1). By plotting stellar brightness as a function of time, one can observe a light curve with periodic dips, each corresponding to a planetary transit.

Using the transit method, valuable information about exoplanets can be obtained. The amount of light blocked during a transit reveals the planet's size, as larger planets block more light. Furthermore, the time between successive transits determines the planet's orbit and orbital period. Detailed analysis can also provide insights into the planet's atmosphere, if one is present.

However, the transit method has limitations. It can only detect planets whose orbits are aligned to pass in front of their star from our perspective. Moreover, other astronomical phenomena, such as eclipsing binary stars, can mimic planetary transit signals, requiring additional confirmation to avoid false positives.

After the initial observation and data capture by the telescope, raw data, such as that shown in Fig. 1, undergoes extensive processing. Before detection algorithms are applied, preprocessing techniques are typically performed. For the Kepler mission, examples include removing variations unrelated to planetary transits (Stumpe et al 2012b) and correcting systematic errors using Bayesian statistics (Smith et al. 2012a). Once the data is calibrated, algorithms analyze the light curves of thousands of stars, searching for periodic dimming indicative of potential transits. Methods for exoplanet identification include the box least squares (BLS) algorithm Kovács 2002 and the transit least squares (TLS) algorithm (Hippke and Heller 2019). Additional algorithms (e.g., Kovács 2005) are also used to validate potential transit events and identify false positives caused by astrophysical phenomena or instrumental issues.

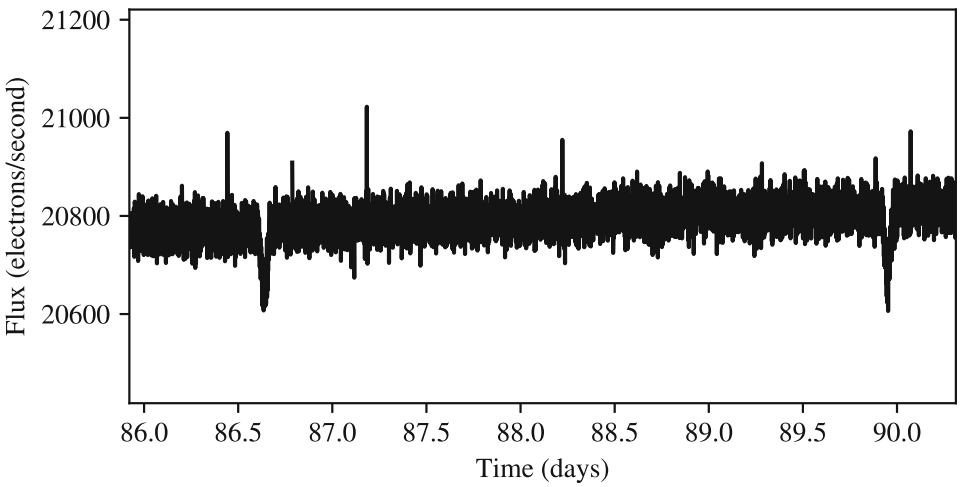

Fig. 1 Transit method. Fragment of a timeline showing a periodic decay in the brightness of the star with a period of 3.31 days (in this graph the decay is observed at day 86.64 and 89.95). The timeline resulted in the confirmation of the existence of the planet Kepler-424b. The parameters obtained from these measurements along with those of other exoplanet candidates have been used to train the proposed neural network (data extracted from Exoplanet Archive Kepler Resources)

Concisely, distinguishing planetary candidates from false positives requires analysis of both the flux time series and pixel data provided by Kepler (Batalha et al. 2013). This process produces catalogs of Kepler objects of interest (KOIs), which in practice are exoplanet candidates. Each KOI undergoes additional tests and analyses to determine its nature (Jenkins et al. 2010; Jenkins 2020). For confirmed KOIs, detailed parameters are derived, such as the orbital period, transit depth, and the host star's effective temperature. These results are cross-verified using supplemental observations and statistical analyses to ensure their reliability. The calculated parameters, along with uncertainties, are documented. Finally, the processed and validated data is compiled into public catalogs, such as the NASA Exoplanet Archive, which are accessible to the scientific community and the public.

While raw light curve data, like that shown in Fig. 1, is analyzed during the initial stages, working with parameter tables offers several advantages. These include isolating relevant data, maintaining uniformity in KOI analysis, and accelerating classification. In any case, when a KOI exhibits particularly interesting parameters, raw data is revised for a more detailed analysis.

Despite KOIs already being classified as false positives or confirmed, the use of machine learning with these data tables is valuable for several reasons: (a) it allows for independent validation and refinement of existing classifications, potentially identifying errors or inconsistencies; (b) machine learning can automate the classification process for new observations from future missions, enhancing efficiency and speed. (c) continuous improvement of detection algorithms is possible by updating models with new data, and machine learning can uncover patterns and relationships between features that traditional methods might miss, in fact, it can help to classify candidates for exoplanets as we will see in Section 5.1; (d) these models can serve as diagnostic tools, helping astronomers assess the reliability of detections and prioritize candidates for further investigation, thereby broadening and enhancing the scientific understanding of exoplanets; and (e) these techniques can analyze large volumes of data and recognize patterns that might go unnoticed with traditional methods. Examples of machine learning methods used for Kepler data include neural networks, namely, multilayer perceptrons (Pearson et al 2017) and 1-dimensional convolutional neural networks (Shallue and Vanderburg 2018).

Machine learning techniques have several advantages, such as the ability to analyze large datasets efficiently and quickly, identify planetary transits that are difficult to detect with traditional methods due to their ability to recognize complex and non-linear patterns, and continuously improve their accuracy by training with new data. However, they also have limitations: they require large amounts of labeled data to properly train models, and can require significant computational resources for both training and execution depending on the method used.

In practice, a combination of both approaches can be the most effective strategy: traditional methods can be used for initial detection, while machine learning methods can refine and validate these findings, as well as uncover transits that might have gone unnoticed. In this manuscript we will focus on the latter through quantum circuits, whose fundamentals are described in the following section.

3 Background in quantum circuits and VQCs

In this section, we delve into three key subjects that lay the groundwork for our research. Firstly, we explore the basics of quantum computing, providing insights into its fundamental principles. Following this, an introduction to quantum machine learning and variational quantum circuits (VQCs) will be provided, demonstrating their versatility as tools in quantum computation. Finally, we examine the training processes of variational quantum classifiers, revealing the techniques that train these classifiers to discover patterns related to exoplanet classification.

3.1 Quantum computing basics

A quantum computer is a type of computer that leverages the principles of quantum physics to execute specific algorithms more efficiently than classical computers.

While the bit is the minimum unit of information in classical computing, the qubit is the minimum unit of information in quantum computing. It is characterized as a two-dimensional Hilbert space housing two orthogonal bases arbitrarily denoted as $|0\rangle$ and $|1\rangle$, recognized as computational bases within two-level quantum computing. Both bits and qubits have state, but while bits have only two possible states, the state of a qubit can be represented as the linear combination of the two bases, each multiplied by complex amplitudes. This feature arises from the unique property of quantum states, known as superposition:

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

where $\{\alpha, \beta\} \in \mathbb{C}$ are the complex amplitudes whose fundamental property is that $|\alpha|^2 + |\beta|^2 = 1$.

Another property of quantum qubits is entanglement, which can be defined as the fact that, given a quantum system, the state of the subsystems (qubits) cannot be described independently. In other words, if the system is studied separately, information about the system as a whole will be lost.

Quantum computers are composed of quantum circuits. These latter are used to process qubits in the same way that classical digital circuits are used to process bits (Nielsen and

Chuang 2010). Namely, qubits are manipulated by quantum gates, which can be represented either as Hamiltonians or as unitary matrices. In quantum computing, gates are the building blocks for constructing quantum circuits, enabling the execution of quantum algorithms and computations. These gates manipulate the quantum states of the qubits, facilitating operations such as superposition, entanglement, and interference. Examples of gates include the CNOT gate and the Hadamard gate. Since gates and therefore circuits can be modeled as unitary matrices, two fundamental properties of quantum circuits are that they are reversible and that there is no easy way to copy qubits (no-cloning theorem).

An advantage of quantum circuits is their ability to harness the phenomenon of superposition. In the previous equation, we could define a state $|\psi\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$ that represents a combination of the $|0\rangle$ and $|1\rangle$ states simultaneously. In general, the number of possible states increases exponentially by a factor of 2^n, where n stands for the number of qubits involved in the computation. This superposition of states is the cornerstone of the computational power of quantum systems, allowing them to explore multiple paths simultaneously, potentially leading to exponential speed-ups in certain computational tasks compared to classical systems. Additionally, quantum superposition enables the encoding of complex information in a compact and efficient manner, facilitating the implementation of quantum algorithms for solving specific problems.

After the qubits are processed by the quantum circuits, the final step is to measure them. This yields a measurement outcome arbitrarily represented as $+1$ for the $|0\rangle$ state and -1 for the $|1\rangle$ state, following a predetermined probability distribution linked to the quantum state represented by α and β. Since these two variables stand for probability, multiple measurements are usually mandatory to determine the precise quantum state of the circuit output.

3.2 Quantum machine learning and VQC

To combine quantum computing and machine learning, we consider whether the data is generated by a quantum (Q) or classical (C) system and whether the information processing device is quantum (Q) or classical (C) (Aïmeur et al. 2006; Schuld and Petruccione 2018).

The CC approach refers to classical data being processed classically. This is the conventional method used in machine learning.

The QC approach (quantum data, classical information processing) explores how machine learning can enhance quantum computing by analyzing quantum data with classical algorithms. For example, machine learning can deduce the internal state of a quantum computer with minimal measurements.

The QQ approach examines how quantum data is processed by a quantum computer. This may involve data from the measurement of a quantum system in a physical experiment being fed into a separate quantum processing device. A more typical setup involves using a quantum computer to simulate the dynamics of a quantum system, such as a hydrogen atom.

In this article, we focus on the CQ approach, which leverages quantum computing to process classical datasets via a quantum-classical interface. One of the primary objectives of the CQ approach is to develop quantum algorithms for machine learning, encompassing both the adaptation of classical machine learning models and the creation of new models inspired by quantum computing principles (Schuld and Petruccione 2018).

In the CQ setting, hybrid quantum-classical algorithms involve scenarios where a classical algorithm optimizes a parameterized quantum circuit to solve a specific problem. Several approaches exist for implementing this paradigm (Rakyta et al. 2024). Among them, the variational quantum circuit (VQC) is one of the most widely used methods. VQCs are quantum circuits that depend on free parameters and demonstrate great potential for addressing a wide range of problems (e.g., Salmeron and Fernández-Palop (2023)). These circuits typically consist of: (a) the preparation of a fixed initial state (e.g., the zero state); (b) a feature map $S(x)$ parameterized by classical variables representing the input x; (c) a variational form $U(\theta)$, also known as an ansatz, where the parameters θ are optimized to minimize a given loss function L based on the error E; and (d) the measurement of the qubits. A scheme of the VQC can be found in Fig. 2.

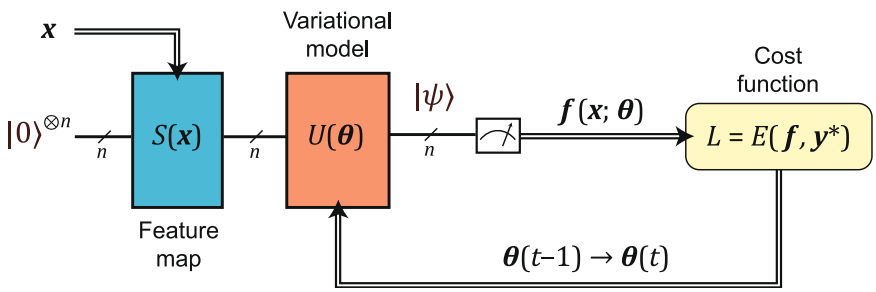

Fig. 2 Overview of the VQC used in this work. The input x is encoded into an n-qubit Hilbert space by applying the feature map $S(x)$. This encoding is then processed via a variational form, also called ansatz $U(\theta)$. Finally, a measurement is carried out whose outcome f is post-processed to calculate the loss function L and update θ

As stated above, VQCs leverage an interactive loop connecting a classical computer and a quantum device. The former manages the adjustment of ansatz parameters, guided by the measurement outcomes derived from the quantum hardware. This iterative process continues until the minimization (or maximization) of L is achieved. Some reasons for using classical optimizers are as followsPellow-Jarman et al. (2021): (a) quantum hardware is currently limited in terms of qubit count, connectivity, noise, and decoherence; (b) calculating gradients directly on quantum hardware could be complex because of the quantum nature of the operations; and (c) the combination of classical and quantum resources in hybrid algorithms can serve to take advantage of the strengths of the CQ paradigm.

VQCs are trained by a classical optimization algorithm that makes queries to the circuit. The optimization is usually an iterative scheme that searches out better candidates for the parameters θ with every step.

Our VQCs apply a parity mapping to map the bitstring to the classification, resulting in a probability vector that is interpreted as a one-hot encoded result. Because of its use for classification purposes, VQCs are also named Variational Quantum Classifiers. In this work, the cross-entropy loss function was applied to compute L, which expects labels given in a one-hot encoded format and will return predictions in the same format.

There are no explicit conventions regarding the VQC architecture, feature map, or ansatz, which allows them to encompass a wide range of possibilities that we explore below.

4 Method

Taking into account that we chose to use a VQC to perform the KOI classification, this section will explain how the data preprocessing has been carried out for its training. Then, the choice of the feature map and the ansatz will be explained. At the end of this section, the algorithm followed for the optimization of the VQC will be explained.

In addition to Qiskit, the experiments were programmed using Python 3.7 and Numpy Harris et al. (2020), Pandas Pandas (2020), and Scikit Pedregosa et al. (2011) as support libraries.[1]

4.1 Data and its encoding

For this work, we used a dataset composed of KOIs, whose features are described in Section 2. The dataset was downloaded from NASA Exoplanet Archive of Kepler Mission (see Appendix A for additional details).

[1] The code used is available on demand.

Although current computational quantum systems boast capacities exceeding 1000 qubits (Castelvecchi 2023), this work aims to minimize the number of qubits used to keep noise as low as possible. To achieve this, we reduced the 26-dimensional dataset 7 to dimensions of 2, 5, and 8. While purely quantum methods for dimensionality reduction exist (e.g., Mengoni and Di Pierro (2019); Mahmud et al. (2024); Simon and Radons (2024)), classical principal component analysis (PCA) was employed due to its simplicity. PCA (Pearson 1901) is a statistical technique used to simplify the complexity of high-dimensional data while preserving trends and patterns. It achieves this by transforming the original variables into a new set of variables, called principal components, which are orthogonal and ordered by the amount of variance they capture in the data. The first few principal components retain the most significant aspects of the data, enabling dimensionality reduction with minimal loss of information. PCA is widely used in fields such as signal and image processing to uncover underlying structures, suppress noise, and facilitate the visualization of high-dimensional data. For this work, the PCA implementation of Scikit-learn (Pedregosa et al. 2011) was used.

With PCA the parameter size was reduced to specific dimensions (concretely 2, 5, and 8). Figures 3 and 4 show the pairwise relationships in the training and test datasets after applying dimensionality reduction to 5 and 2 dimensions, respectively (dimension 8 was omitted for clarity). It can be observed that in both cases, the distributions of both exoplanet classifications (*confirmed* and *false positive*) overlap within a specific region.

Although not shown in these figures, it is worth noting that, starting from dimension 5, at least one variable exhibits a significant correlation with the 'koi_period' parameter (the interval between consecutive planetary transits). This observation is not surprising, as the parameter features prominently in the raw dataset, as illustrated in the timeline in Fig. 1.

Now, this dataset of reduced dimension modulates the behavior of the feature map according to explained in Section 3.2. This dataset was adapted for each parameter to have values between -1 and 1 so that the feature map can better handle them (see Figs. 3 and 4).

For quantum processing Qiskit library (Javadi-Abhari et al. 2024) was chosen despite the model can be adapted to Pennylane (Bergholm et al. 2022).

4.2 Feature map

According to Section 3.2, VQCs are composed of feature maps $S(x)$ and variational forms (ansatz). The feature maps are fundamental for encoding classical data into quantum states, allowing quantum algorithms to process and analyze the dataset. These feature maps transform input data into a higher-dimensional quantum Hilbert space, enabling

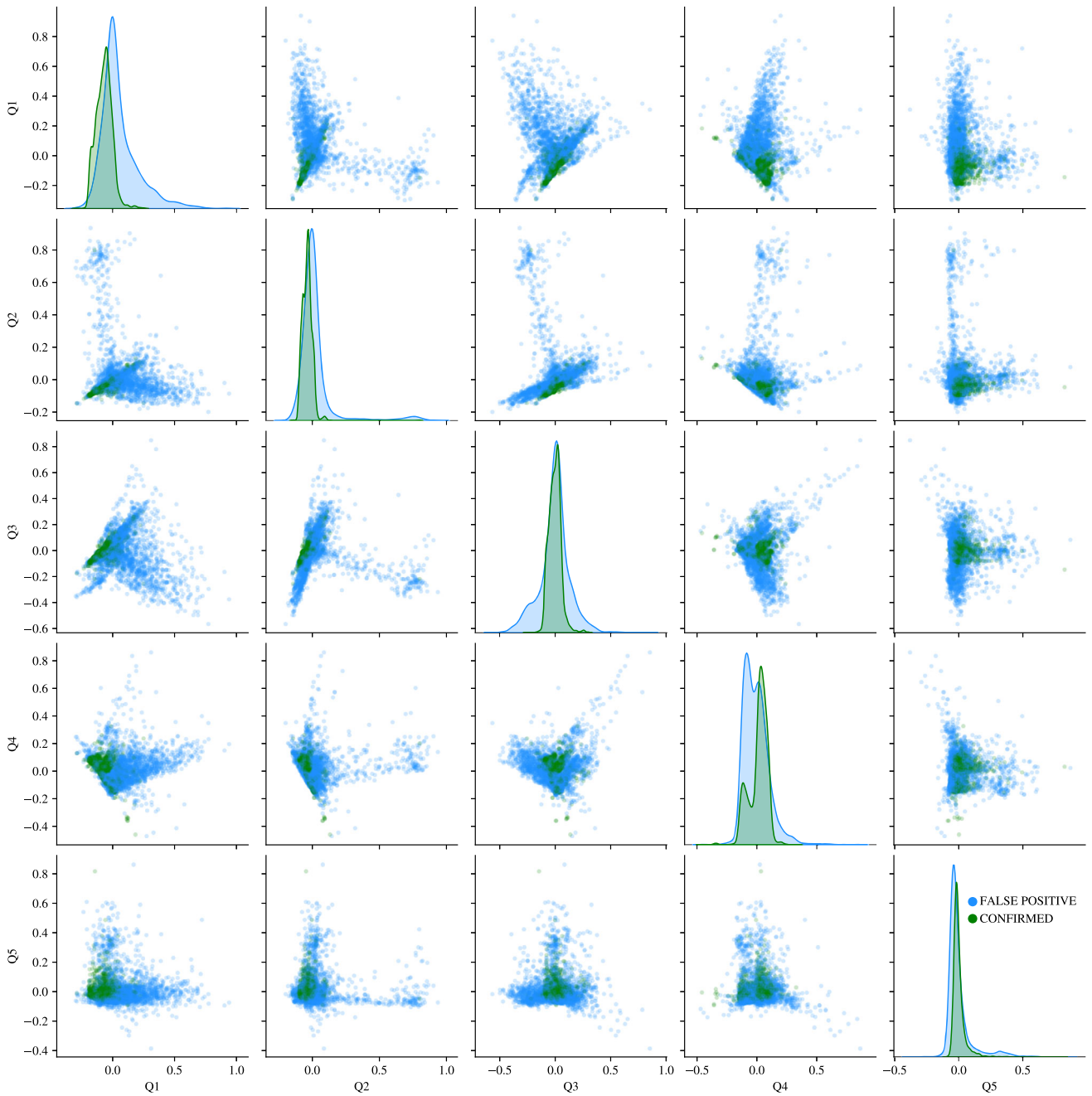

Fig. 3 Pairwise relationships in the dataset for 5 dimensions (qubits)

quantum circuits to exploit the structures and relationships within the data (Suzuki et al 2020). Different types of feature maps, such as the first-order Pauli-Z evolution circuit (ZFeatureMap) and the second-order Pauli-Z evolution circuit (ZZFeatureMap), offer varying levels of performance and accuracy in these applications.

For KOI classification, the second-order Pauli-Z evolution circuit, or ZZFeatureMap, has shown superior results compared to other feature maps, yielding improved accuracy and performance of the VQCs in all cases, as we will see in the following section. In contrast, the first-order Pauli-Z evolution circuit, or ZFeatureMap, does not incorporate entanglement and thus provides a less expressive feature map. Consequently, using the ZFeatureMap often results in significantly decreased accuracy as it fails to capture the complex patterns and correlations present in the dataset.

A ZZFeatureMap of 5 qubits, used latter for the results is shown in Fig. 5. It is important to highlight that repeating

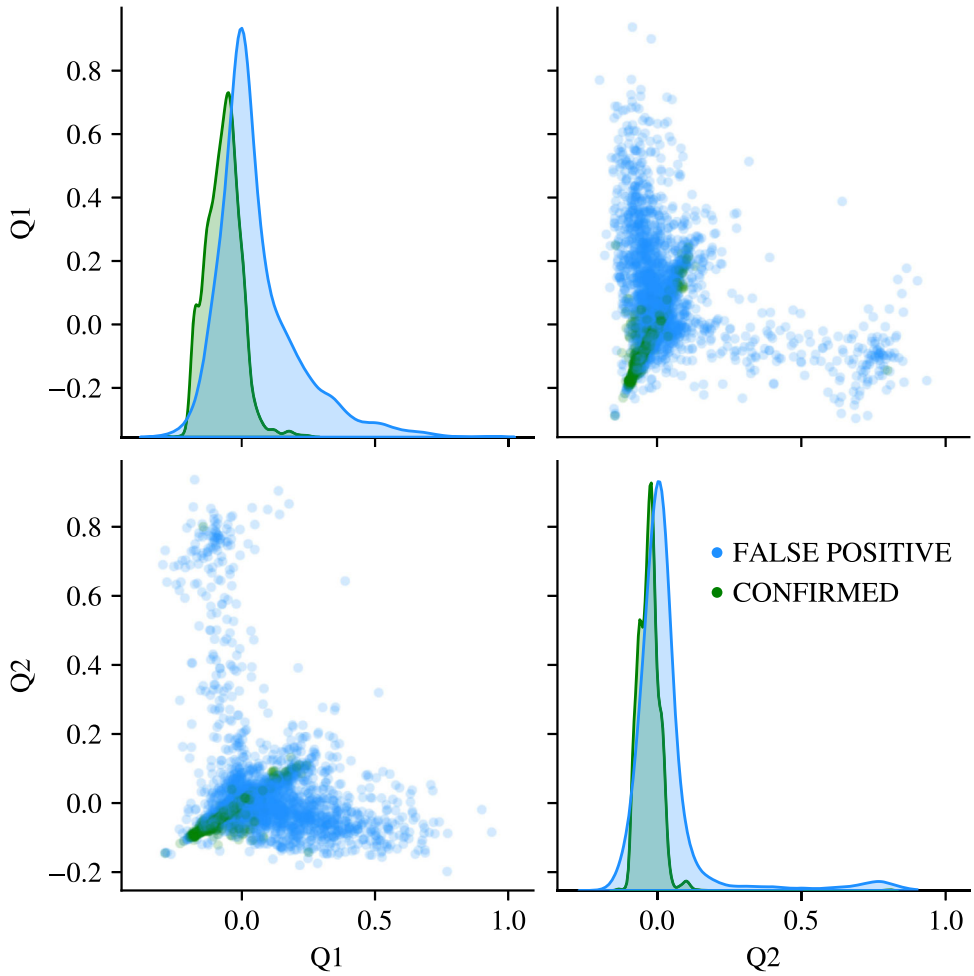

Fig. 4 Pairwise relationships in the dataset for 2 dimensions (qubits)

the ZZFeatureMap circuit does not result in an increase in VQC accuracy and become increasingly susceptible to decoherence, defined as a process by which the quantum state of a system interacts with its environment, causing the system to lose its quantum properties and transition to a classical state (Nielsen and Chuang 2010). This can arise due to various factors, such as imperfections in the hardware, environmental noise, and the sheer complexity of the circuit. Maintaining coherence becomes challenging as the number of operations and the duration of the computation increase.

4.3 Variational form

The variational form or ansatz $U(\theta)$ refers to the general structure of the quantum circuit that will be used to approximate a solution to a specific problem. It is a parametrized scheme that defines how the quantum gates will be organized and applied in the circuit. This component of the circuit is chosen to be flexible enough to be able to represent a wide variety of possible quantum states by varying its parameters.

The RealAmplitudes circuit depicted in Fig. 6 yields the best results. It is common to iterate these circuits a certain

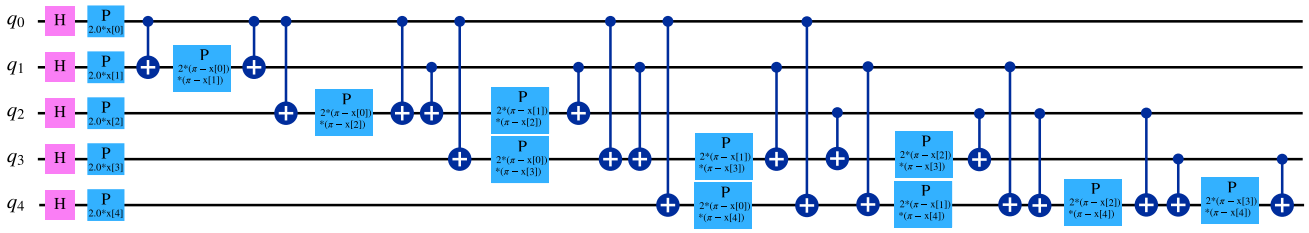

Fig. 5 Feature map used for this VQC (ZZFeatureMap)

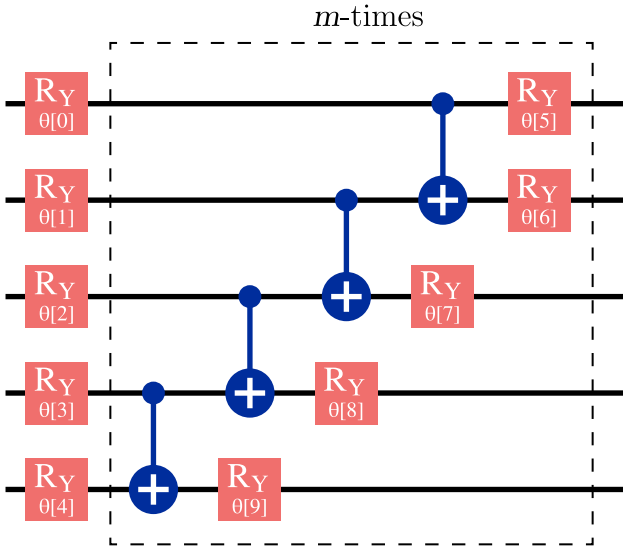

Fig. 6 Ansatz with m repetitions used in the implementation of this VQC (RealAmplitudes)

number of times, m. An increasing number of repetitions improved the results, as shown in 5.

Similar results are obtained when using the EfficientSU2 circuit shown in Fig. 7 for various qubits and repetitions as shown in Section 5. However, the number of parameters of this circuit is increased more with the number of repetitions (10 per m) with respect to the RealAmplitude circuits (5 per m), as shown in Figs. 6 and 7 without improving the results significantly. Therefore, we will focus on the results with RealAmplitude as ansatz circuits.

After selecting both parts of the circuit, we integrated them into a single quantum circuit to form the VQC. The adjustment of its parameters is discussed in the next section.

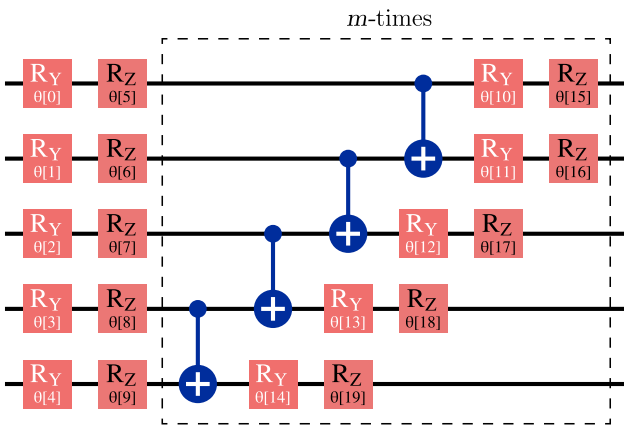

Fig. 7 Alternative ansatz with m repetitions used in the implementation of this VQC (EfficientSU2)

5 Results

Having chosen the feature map and the variational form to compose the VQC, the next step is to train it with a supervised method with the dataset preprocessed with PCA representing KOIs. Because of the time-consuming nature of gradient descent training for VQCs, a common gradient-free technique in VQC training called Constrained Optimization BY Linear Approximations (COBYLA) was chosen.

COBYLA is an optimization algorithm designed for constrained problems where the objective function and constraints are non-linear and potentially non-smooth (Powell 1994). It operates without requiring gradient information, making it valuable for optimizing functions that are difficult to differentiate. COBYLA works by iteratively calculating linear approximations to the objective and constraint functions and then solving a series of linear equations. One of the key advantages of COBYLA is its ability to handle constraints directly, ensuring that all iterations satisfy them within a specified tolerance. As mentioned in Section 3.2, the VQC was trained with cross-entropy because it is a common loss function for classification problems.

The training dataset consisted of 2500 samples, and the test dataset also contained 2500 samples. The number of training epochs in all cases was 800, which was sufficient for convergence.

Table 1 shows the training results and validation results. We obtained an accuracy rate of approximately 80 percent. The accuracy value during training is similar to the accuracy value with the test data in all cases, indicating that overfitting did not occur. In fact, this similarity between training and test accuracy suggests that the model generalizes well to unseen data, revealing its robustness and reliability in classifying KOIs.

For comparison purposes with classical neural networks (NNs), a multilayer perceptron with Tensorflow (Abadi et al.

Table 1 Training results for several VQCs after 800 epochs with COBYLA

Qubits	Ansatz (m)	Parameters	Accuracy (Train)	Accuracy (Validation)
2	2	6	0.7132 ± 0.0023	0.6883 ± 0.0019
	5	12	0.7164 ± 0.0028	0.6883 ± 0.0008
	10	22	0.7149 ± 0.0002	0.6877 ± 0.0001
5	2	15	0.7302 ± 0.0141	0.7109 ± 0.0211
	5	30	0.7890 ± 0.0149	0.7735 ± 0.0127
	10	55	0.8053 ± 0.0021	0.7946 ± 0.0036
8	2	24	0.7101 ± 0.0350	0.7045 ± 0.0243
	5	48	0.7673 ± 0.0225	0.7626 ± 0.0231
	10	88	0.7849 ± 0.0146	0.8016 ± 0.0152

Table 2 Training results for a classical neural network after 50 epochs with gradient descent

Number of inputs	Layers	Parameters	Accuracy (Test)	Accuracy (Validation)
2	2-2-2	12	0.7736 ± 0.0029	0.7543 ± 0.0032
	2-3-2	17	0.7711 ± 0.0015	0.7636 ± 0.0074
	2-5-2	27	0.7718 ± 0.0019	0.7649 ± 0.0110
5	5-3-2	26	0.7975 ± 0.0007	0.7733 ± 0.0030
	5-5-2	42	0.7768 ± 0.0112	0.7772 ± 0.0069
	5-10-2	82	0.8047 ± 0.0060	0.7879 ± 0.0143
8	8-3-2	35	0.8632 ± 0.0040	0.8616 ± 0.0134
	8-5-2	57	0.8623 ± 0.0018	0.8713 ± 0.0052
	8-10-2	112	0.8620 ± 0.0018	0.8629 ± 0.0049

2015) has also been implemented with a similar number of parameters to those used in the VQCs of Table 1. All these NNs consist of a hidden layer of varying number of neurons in that layer. The activation function of the layers is *tanh* function for the hidden layer (because it was the one that gave the best results) and *sigmoid* function at the output (to be able to compare results directly with the VQCs). It was trained using gradient descent and the cross-entropy loss function.

The data used to train the NN and the VQC were the same. The results obtained with the NN are shown in Table 2. We noted that, for this particular problem, the results are slightly better than those obtained with VQC and that the results with NN depend more on the number of inputs than on the number of the network parameters. The best result obtained for the classification of KOIs is found in the last line of Table 1.

Thus, although the results between NN and VQC are comparable, work in quantum machine learning remains to be done to improve those of the latter.

Regarding entanglement in a VQC, an insufficient degree of entanglement can limit the circuit's ability to represent the correct solution to a problem. It is well-established that CNOT gates introduce entanglement between two qubits, making the configuration of CNOT gates within a quantum circuit a key strategic choice in the design of quantum algorithms and architectures. In the feature map, initial entanglement is introduced to the inputs (Fig. 5). In the ansatz used, CNOT gates are applied between adjacent qubits (Fig. 6), causing the entanglement to propagate locally. With additional iterations of the ansatz, the entanglement spreads throughout the system.

One way to quantify the degree of entanglement is by using the entanglement entropy. For two qubits, this value is 1 when the qubits are maximally entangled and 0 when there is no entanglement. We measured entanglement entropy for a system of 2 qubits (representing the entanglement of one qubit with respect to another) for three repetitions of the ansatz, as shown in Table 1 ($m = 2, 5, 10$). Given as input the KOI dataset, the mean entanglement entropy of the output is 0.30 ± 0.12, 0.35 ± 0.08, and 0.47 ± 0.21, respectively. These results suggest that increasing the ansatz complexity increases the level of entanglement, which intrinsically improves VQC performance beyond adding more parameters to it.

In this work, both quantum and classical classifiers were simulated without noise. Classical neural networks, as used in Table 1, are inherently robust to noise (Goodfellow et al. 2016). In contrast, quantum noise manifests in various forms, such as gate errors and decoherence, which can degrade the effectiveness of quantum models. Although these pose challenges to the implementation of quantum learning systems, they may also offer advantages during training by helping to mitigate overfitting. Indeed, it has been demonstrated that quantum neural networks can exhibit robustness to both noise and decoherence (Nguyen et al. 2020). However, in other VQCs, performance deteriorates significantly in the presence of quantum noise (Qi et al. 2025). The outcome largely depends on the noise model used Nielsen and Chuang (2010) and, most importantly, on the deployment of the VQC on actual quantum hardware.

5.1 Classification of exoplanets candidates

Apart from KOIs classified as *confirmed* and *false positive* (Section 4.1), in Exoplanet Archive of Kepler Mission (Appendix A) there are KOIs classified as *canditate*. It means that the object has features that suggest it could be an exoplanet, such as regular transit events that are consistent with planetary transits. However, additional observations or analysis are required to rule out other possible causes (e.g., binary star systems, background stars, or instrumental noise) and to confirm it as an actual planet, so it has not yet been confirmed. KOIs with label *candidate* were removed from both training and test datasets to carry out the training and test process described in Section 5.

In this part of the experiment, we evaluated several KOIs that had been removed from the training and test sets because we did not know whether they were false positives or not. These candidates are evaluated by both VQC and NN, specifically by the 55-parameter VQC from Table 1 and the 74-parameter NN from Table 2 for having the same number of entries and similar accuracy results with the test data. The results of this comparison are shown in Fig. 8.

Although both types of classifiers agree on most classifications, there is a significant discrepancy between them. This suggests that while the VQC may not consistently achieve

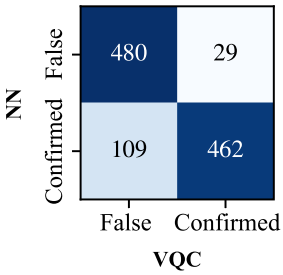

Fig. 8 Classification made by NN and VQC of exoplanet candidates from the dataset used of KOIs

high accuracy, it can uncover patterns that the classical NN might struggle to detect.

6 Conclusions

This article presents an exoplanet detection system trained on data from the Kepler mission. The system is based on a variational quantum circuit (VQC) implemented with the Qiskit library. Despite recent advancements in the number of qubits available in quantum circuits, we have prioritized maintaining a simple architecture with few qubits. Given the high dimensionality of the dataset relative to the number of qubits, principal component analysis (PCA) was used for dimensionality reduction. Accuracy rates were tested with different number of qubits and various architectures. The analysis of the impact of the number of parameters on the ansatz on accuracy yielded valuable insights. For this specific problem, adding more parameters to the feature map stage did not lead to improved accuracy in either the training or validation stages. However, a notable increase in accuracy was achieved by adding more parameters in the ansatz stage. The results were compared with those of a classical NN, namely a multilayer perceptron. While both systems produced similar results with a small number of parameters, as the number of parameters increased, the classical neural network began to outperform our VQCs. Therefore, further research into the design and training of VQCs is needed to achieve performance comparable to or surpassing that of a neural network with a high number of parameters. Nonetheless, it was demonstrated that VQCs are a viable method for processing KOIs and discovering exoplanets, capable of revealing patterns that are challenging for classical NNs to detect.

Appendix A: Kepler data used for the training

From plots of the light curves like those in Fig. 1, we obtain quantitative values that provide detailed information about the stellar system and the exoplanet. These parameters are accessible through the mission web page. Specifically,

The dataset for training and test the VQC was taken from Kepler mission NASA Exoplanet Archive https://exoplanetarchive.ipac.caltech.edu/cgi-bin/TblView/nph-tblView?app=ExoTbls&config=koi.

The name of the parameters selected for this experiment were as follows: “koi_period”, “koi_period_err1” (the interval between consecutive planetary transists), “koi_impact”, “koi_impact_err1”, “koi_impact_err2” (the sky-projected distance between the center of the stllar disc and the center of the KOI disc at conjunction normalized by the stellar radius), “koi_duration”, “koi_duration_err1” (the duration of the observed transits), “koi_depth”, “koi_depth_err1” (the fraction of stellar flux lost at the minimum of the KOI transit), “koi_prad”, “koi_prad_err1”, “koi_prad_err2” (radius of the KOI), “koi_teq” (temperature of the KOI), “koi_insol”, “koi_insol_err1”, “koi_insol_err2” (insolation flux), “koi_model_snr” (transit depth normalized by the mean uncertainty in the flux during the KOI transit), “koi_steff”, “koi_steff_err1”, “koi_steff_err2” (photospheric temperature of the start), “koi_slogg”, “koi_slogg_err1”, “koi_slogg_err2” (logarithm of the acceleration due to gravity at the surface of the star), “koi_srad”, “koi_srad_err1”, “koi_srad_err2” (photospheric radius of the star). The “_err1” and “_err2” suffixes are the lower and upper uncertainly of these parameters, respectively. This gives a dataset of dimensionality 26 that was reduced to a dimensionality equal to the number of qubits of the circuit using the PCA algorithm as explained in Section 4.1. Some KOI has “NaN” values; in this case, these values were filled with the mean value of the parameter of all the datasets.

The rest of the parameters were discarded because they offered little information with respect to the first ones. Parameters such as “koi_fpflag_nt”, “koi_fpflag_ss”, “koi_fpflag_co”, and “koi_fpflag_ec” were discarded too because, obviously, it increases significantly the accuracy results.

The exoplanet classification was carried out using the “koi_disposition” one-encoded parameter.

Author contribution A sole author.

Funding Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature.

Declarations

Competing interests The author declares no competing interests.

References

Abadi M et al (2015) TensorFlow: large-scale machine learning on heterogeneous systems. Software available from tensorflow.org . https://www.tensorflow.org/

Aïmeur E, Brassard G, Gambs S (2006) Machine learning in a quantum world. In: Advances in Artificial Intelligence: 19th Conference of the Canadian Society for Computational Studies of Intelligence, Canadian AI 2006, Québec City, Québec, Canada, June 7-9, 2006. Proceedings 19, Springer, pp 431–442

Bergholm V et al (2022) PennyLane: automatic differentiation of hybrid quantum-classical computations

Batalha NM, Rowe JF, Bryson ST, Barclay T, Burke CJ, Caldwell DA, Christiansen JL, Mullally F, Thompson SE, Brown TM et al (2013) Planetary candidates observed by Kepler. III. analysis of the first 16 months of data. The Astrophysical Journal Supplement Series 204(2):24

Castelvecchi D (2023) IBM releases first-ever 1,000-qubit quantum chip. Nature 624(7991):238–238

Dattilo A, Vanderburg A, Shallue CJ, Mayo AW, Berlind P, Bieryla A, Calkins ML, Esquerdo GA, Everett ME, Howell SB et al (2019) Identifying exoplanets with deep learning. II. two new super-earths uncovered by a neural network in k2 data. Astron J 157(5):169

Goodfellow I, Bengio Y, Courville A (2016) Deep learning. MIT press

Harris CR et al (2020) Array programming with NumPy. Nature 585(7825):357–362. https://doi.org/10.1038/s41586-020-2649-2

Hippke M, Heller R (2019) Optimized transit detection algorithm to search for periodic transits of small planets. Astron Astrophys 623:39

Jenkins JM et al (2010) Overview of the Kepler science processing pipeline. Astrophys J Lett 713(2):87. https://doi.org/10.1088/2041-8205/713/2/L87

Javadi-Abhari A, Treinish M, Krsulich K, Wood CJ, Lishman J, Gacon J, Martiel S, Nation PD, Bishop LS, Cross AW, Johnson BR, Gambetta JM (2024) Quantum computing with Qiskit. https://doi.org/10.48550/arXiv.2405.08810

Jenkins JM (2020) Kepler data processing handbook: overview of the science operations center, id. 2. Kepler Science Document KSCI-19081-003, 2

Transiting exoplanet discovery using machine learning techniques (2020) Jara-Maldonado M, Alarcon-Aquino V, Rosas-Romero R, Starostenko O, Ramirez-Cortes J.M. A survey. Earth Sci Inform 13:573–600

Kovács G, Bakos G, Noyes RW (2005) A trend filtering algorithm for wide-field variability surveys. Mon Not R Astron Soc 356(2):557–567

Kovács G, Zucker S, Mazeh T (2002) A box-fitting algorithm in the search for periodic transits. Astron Astrophys 391(1):369–377

Mengoni R, Di Pierro A (2019) Kernel methods in quantum machine learning. Quantum Mach Intell 1(3):65–71

Mahmud J, Mashtura R, Fattah SA, Saquib M (2024) Quantum convolutional neural networks with interaction layers for classification of classical data. Quantum Mach Intell 6(1):11

Nguyen NH, Behrman EC, Steck JE (2020) Quantum learning with noise and decoherence: a robust quantum neural network. Quantum Mach Intell 2(1):1

Nielsen MA, Chuang IL (2010) Quantum computation and quantum information: 10th anniversary edition. Cambridge University Press

Pandas (2020) Pandas-dev/pandas: pandas. Zenodo. https://doi.org/10.5281/zenodo.3509134

Perger M, Anglada-Escudé G, Baroch D, Lafarga M, Ribas I, Morales J, Herrero E, Amado PJ, Barnes J, Caballero J et al (2023) A machine learning approach for correcting radial velocities using physical observables. Astron Astrophys 672:118

Pearson K (1901) On lines and planes of closest fit to systems of points in space. The London, Edinburgh, and Dublin philosophical magazine and journal of science 2(11):559–572

Pellow-Jarman A, Sinayskiy I, Pillay A, Petruccione F (2021) A comparison of various classical optimizers for a variational quantum linear solver. Quantum Inf Process 20(6):202

Powell MJ (1994) A direct search optimization method that models the objective and constraint functions by linear interpolation. Springer

Pearson KA, Palafox L, Griffith CA (2017) Searching for exoplanets using artificial intelligence. Mon Not R Astron Soc 474(1):478–491. https://doi.org/10.1093/mnras/stx2761. https://academic.oup.com/mnras/article-pdf/474/1/478/22141859/stx2761.pdf

Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E (2011) Scikit-learn: machine learning in python. J Mach Learn Res 12:2825–2830

Qi H, Xiao S, Liu Z, Gong C, Gani A (2025) A high-efficiency variational quantum classifier for high-dimensional data. J Supercomput 81(1):154

Rakyta P, Morse G, Nádori J, Majnay-Takács Z, Mencer O, Zimborás Z (2024) Highly optimized quantum circuits synthesized via data-flow engines. J Comput Phys 500:112756

Salmeron JL, Fernández-Palop I (2023) Variational quantum circuit topology grid search for hypocalcemia following thyroid surgery. Mathematics 11(17) . https://doi.org/10.3390/math11173659

Schuld M, Petruccione F (2018). Supervised learning with quantum computers Springer. https://doi.org/10.1007/978-3-319-96424-9

Simon L, Radons M (2024) On neural quantum support vector machines. Quantum Mach Intell 6(1):3

Smith JC, Stumpe MC, Van Cleve JE, Jenkins JM, Barclay TS, Fanelli MN, Girouard FR, Kolodziejczak JJ, McCauliff SD, Morris RL et al (2012) Kepler presearch data conditioning ii-a Bayesian approach to systematic error correction. Publ Astron Soc Pac 124(919):1000

Stumpe MC, Smith JC, Van Cleve JE, Twicken JD, Barclay TS, Fanelli MN, Girouard FR, Jenkins JM, Kolodziejczak JJ, McCauliff SD et al (2012) Kepler presearch data conditioning i-architecture and algorithms for error correction in Kepler light curves. Publ Astron Soc Pac 124(919):985

Shallue CJ, Vanderburg A (2018) Identifying exoplanets with deep learning: a five-planet resonant chain around Kepler-80 and an eighth planet around Kepler-90. Astron J 155(2):94

Suzuki Y, Yano H, Gao Q, Uno S, Tanaka T, Akiyama M, Yamamoto N (2020) Analysis and synthesis of feature map for kernel-based quantum classifier. Quantum Mach Intell 2:1–9

Zingales T, Waldmann IP (2018) Exogan: retrieving exoplanetary atmospheres using deep convolutional generative adversarial networks. Astron J 156(6):268