Quantum Machine Intelligence (2024) 6:68
https://doi.org/10.1007/s42484-024-00196-7

RESEARCH ARTICLE

Quarta: quantum supervised and unsupervised learning for binary classification in domain-incremental learning

Corrado Loglisci[1] · Donato Malerba[1] · Saverio Pascazio[2,3]

Received: 19 April 2024 / Accepted: 2 September 2024

Abstract
Quantum machine learning recently gained prominence due to the promise of quantum computers in solving machine learning problems that are intractable on a classical computer. Nevertheless, several studies on problems which remain challenging for classical computing algorithms are emerging. One of these is classifying continuously incoming data instances in incremental fashion, which is studied in this paper through a hybrid computational solution that combines classical and quantum techniques. Hybrid approaches represents one of the current ways for the use of quantum computation in practical applications. In this paper, we show how typical issues of domain-incremental learning can be equally addressed with the properties of quantum mechanics, until to offer often better results. We propose the framework *QUARTA* to combine algorithms of quantum supervised learning, that is, variational quantum circuits, and techniques used in quantum unsupervised learning, that is, distance estimation. We aim at keeping the classification capabilities, which have learned on previously processed data instances, preserved as much as possible, and then acquiring new knowledge on new data instances. Experiments are performed on real-world datasets with quantum simulators.

Keywords Incremental learning · Quantum classification · Quantum distance estimation

1 Introduction

Quantum machine learning has been introduced with the promise to handle machine learning problems that are intractable on a classical computer, especially those characterized by huge amounts of data. In the research on quantum computing (QC) technologies, the current status sees the era of noisy intermediate scale quantum (NISQ) computers (Preskill 2018), which are devices able to deal with low-middle size data problems. An approach which seems bringing practical advantages is instead the one of *hybrid* frameworks (Callison and Chancellor 2022) that combine classical and quantum methods and allow to exploit quantum physics properties while limiting the impact of the existing restrictions of the quantum devices. Clearly, these are not yet the technologies which will guarantee exceptional speed-ups to large data sizes over classical computing, but pave the way to the design of near future algorithms for data-intensive problems. Actually, when the complexity moves to the tractability of the problem rather than the scalability to data volumes, the current purely quantum routines are already of usefulness, for instance in cryptography (Bova et al. 2021).

One of the categories of data-intensive problems in which the research on classical computing dedicates many efforts is the incremental learning of predictive models from sequential data (Gunasekara et al. 2023). In incremental learning, data instances are acquired sequentially and therefore are not available in their whole for a batch learning process. This requires capacity to make predictions in real-time, adapt dynamically to shifting data distributions (referred to as concept drifts), and maintain computational efficiency throughout the learning process (Halstead et al. 2022,Mai et al. 2022,Gunasekara et al. 2022). Even current research in

✉ Corrado Loglisci
corrado.loglisci@uniba.it

Donato Malerba
donato.malerba@uniba.it

Saverio Pascazio
saverio.pascazio@uniba.it

[1] Department of Computer Science, Universita' degli studi di Bari 'Aldo Moro', Bari I-70126, Italy

[2] Department of Physics, Universita' degli studi di Bari 'Aldo Moro', Bari I-70126, Italy

[3] Sezione di Bari, National Institute for Nuclear Physics (INFN), Bari I-70126, Italy

Published online: 14 October 2024

deep learning struggle with incremental learning issues. This is demonstrated by a numerous list of works focused on the phenomenon of *catastrophic forgetting* (Peng et al. 2022). Specifically, catastrophic forgetting refers to the tendency of an artificial neural network to abruptly and drastically forget previously learned information when exposed to new data. In such cases, the focus shifts from designing algorithms for massive computation to ensuring the quality of models remains high over unbounded sequences of data.

What sets incremental learning apart is its resilience to the constraints of QC devices. Unlike certain computationally intensive tasks, incremental learning does not impose stringent requirements on the computational capabilities of quantum devices. Consequently, it holds promise for implementation on NISQ devices, which are currently at the forefront of QC technology. NISQ devices, while limited in their computational power compared to fully fault-tolerant quantum computers, possess sufficient capabilities to tackle moderate-scale data problems, making them viable candidates for incremental learning tasks.

Despite its potential, incremental learning remains relatively under explored in the domain of quantum machine learning. This can be attributed in part to the inherent challenges associated with this learning paradigm.

As a basis, classical data instances can be mapped to quantum states corresponding to points of the Hilbert space, which promises a potential exponentially enlarged representation space to embed real-life data distributions (Jiang et al. 2022; Dunjko and Briegel 2018). Indeed, the features of classical data (for instance, a feature vector with three entries) can be blow up into a representational space larger than the classical one (for instance, eight dimensions, resulting from 2^3, three is the feature vector size) (Hur et al. 2022).

Second, the capacity of accommodating changes and adapt the models to new incoming data calls for techniques able of tuning the properties of the models, which is the main characteristic of variational quantum circuits (VQCs). VQCs are steady structures of gates characterized by a fixed number of free parameters trainable over an optimization process (Benedetti et al. 2019). So, when concept drifts occur, one can adapt the existing models by updating model parameters along the incoming data.

Third, in incremental learning, data does not hold the IID property; data or descriptive features can be inter-related; inter-relationships can evolve too and concern other features. This can threaten the predictive capabilities whether the models do not adequately accommodate changes related to the inter-relationships. Emblematic is that in deep learning, to capture this aspect, one has to design specific neural architectures, for instance, by means of convolutional neural networks. Instead, QC offers the possibility to straightforwardly account for inter-relationships through the gate-based operations designed for qubit registers. The emblematic case is represented by the controlled gates that can check the presence of relationships of the form *event-action*: when the event occurs (control qubits activated), the action is executed (target qubit function is applied). Specifically for the entanglement circuits, based on controlled gates, it is largely recognized their ability in capturing non-local correlations. For instance, in Sharma et al. (2022), the authors show how improving the performances in a document classification task by applying entanglement circuits on words as a means to consider forms of correlation in a high-dimensional data scenario.

In summary, while incremental learning holds considerable promise for leveraging the capabilities of QC, its adoption and exploration remain nascent. Addressing the challenges posed by data availability and variability is paramount to unlocking the full potential of incremental learning algorithms in QC environments. As research in this area progresses, it is poised to contribute significantly to the advancement of lifelong learning paradigms and their practical applications across various domains.

This manuscript proceeds as follows: Sect. 2 discusses the design decisions behind the proposed framework, *QUARTA* (QUantum-classical framework for binAry classification in IncRemental learning with disTance estimAtion). Section 3.1 described the procedure adopted by QUARTA through the execution of the each component. Section 3.2 reports technical details of each component constituting the framework. Section 4 gathers numerical experimental results concerning the performance of QUARTA on real-world datasets and comparison with alternative techniques. Section 5 overviews the scientific themes related the contribution of this manuscript. Finally, Sect. 6 closes the paper.

2 Contributions

In this paper, we investigate the machine learning problem of the binary classification in the peculiar scenario of domain-incremental, where the target (labeling) remains consistent across time, despite variations in domain distributions (data properties). This means that while the types of inputs may change, the expected outcomes or labels remain constant throughout time. Data arrive sequentially at a regular timing, without no uniform allocation of labeled and unlabeled data. There is no predetermined sequence between the operations of model learning and label prediction.

These points highlight that the classification problem at hand does not necessitate scalable computation or massive data processing typically associated with quantum computing. Instead, it aligns with the lifelong learning paradigm (Liu 2017), focusing on computation over long collections of data rather than large-scale collections. As such, our focus lies on

leveraging quantum techniques to improve classification performance, not solely on computational speed-up (Herrmann et al. 2023).

The points outlined suggest us to work on succession of sub-tasks by alternating model learning (*training session*) with label prediction (*prediction session*), dependently on label availability of the data flow. The classification model is trained at the beginning and adapted on new incoming data, while keeping knowledge learned from previously training sessions as much as possible, without counting on the availability of the previously processed data and without retraining from scratch at the arrival of new data. This way, the model is updated and ready to work on prediction sessions. Every session is associated with a small-data size sequences which comprises labeled data if it concerns the training activity, while it comprises unlabeled data if it concerns the prediction. In these terms, the method trains and updates a classifier on sub-sequences of incoming data instances marked as labeled. Then, the classifier is used to estimate the class value of unseen incoming data instances. The update is performed only when there are changing data, which are identified as labeled data distant from the class prototypes. To detect such changes, we rely on a quantum distance estimation technique, often used in quantum clustering. Thus, labeled data which turn out to be distant from the class prototypes are candidates to be deemed as changing. The class prototypes are continuously updated to accommodate the changes of the class characteristics due to the incoming labeled data.

In crafting such a method, we could only not rely on quantum circuits, because the sequence of the gate-based operations poses challenges in executing operations such as alternating between two sessions. Moreover, the current unavailability of noiseless quantum computers with thousands of qubits paves the way to hybrid classical-quantum methods. These methods aim to integrate quantum and classical computers in addressing machine learning challenges. Consequently, we propose the quantum-classical framework QUARTA, which leverages classical computing techniques to manage sessions, enhance model predictive capability during training, and maintain updated class representatives.

Within QUARTA, quantum components encompass a supervised learning approach for learning and updating a binary classification model, alongside a distance estimation technique for identifying labeled changing data. Both components operate on classical data represented as quantum states via data encoding techniques. Interestingly, they utilize two distinct encoding schemes, each representing different descriptive feature sets. This multifaceted perspective on data resonates with the multi-view learning paradigm (Zhao et al. 2017).

To sum up, the technical contributions in terms of QC of this paper are as follows:

- Development of two quantum circuits: one for a supervised learning task and the other for an unsupervised learning task. The first circuit is employed to train and update the classification model, while the second circuit is used to identify changing data that warrant model adaptation.
- Implementation of the quantum supervised learning algorithm in the form of a variational quantum circuit (VQC), leveraging its tunable parameters to adapt the classification model to labeled data.
- Adoption of different data encodings for the quantum supervised and unsupervised algorithms to represent classical data as quantum states. The quantum supervised algorithm utilizes a feature map technique, while the quantum unsupervised algorithm employs angle encoding. This choice is based on previous literature findings (Sierra-Sosa et al. 2023), indicating that the former offers better performance at the expense of efficiency, which is suitable for improving classification accuracy. Meanwhile, the latter is more efficient and robust, making it suitable for detecting significant changes in data
- Usage of entanglement forms in the feature mapping and VQC to better accommodate detected changes into the classification model. Entanglement circuits have been experimentally shown to capture and reveal pairwise correlations among classical data encoded through two-qubit registers. Strong correlations maintain expected values around entangled states, while uncorrelated data result in increased occurrences of outcomes relative to unentangled states.

3 The hybrid framework QUARTA

In the following, we first provide a procedural description, then provide a description of each component constituent of the method.

The binary classification problem investigated by QUARTA can be formulated as follows: We have data instances $\mathbf{X} \cup y$, with $x_j \in \mathbf{X}$ values of the set of descriptive features or attributes $\mathcal{F}$, while $y \in \{+1, -1\}$ denotes the class label domain, $\mathbf{X} \subseteq \mathbb{R}^{|\mathcal{F}|}$. QUARTA addresses this problem within the context of *domain-incremental learning* (Gunasekara et al. 2023), which refers to a scenario where the distribution of the class label y remains unchanged, those of x change over time.

3.1 The procedure of QUARTA

The overall procedure of QUARTA (reported in Algorithm 1) relies on the classical computing (CC) techniques which decide when activating quantum algorithms. The working

Algorithm 1 [R2A2] Procedure of QUARTA.

Require: $\{L\_container : \langle \mathbf{X}_{L_1}, y_{L_1}\rangle, \langle \mathbf{X}_{L_2}, y_{L_2}\rangle, \ldots \langle \mathbf{X}_{L_i}, y_{L_i}\rangle, \ldots\}$
Require: $\{U\_container : \langle \mathbf{X}_{U_1}, y_{U_1}\rangle, \langle \mathbf{X}_{U_2}, y_{U_2}\rangle, \ldots \langle \mathbf{X}_{U_i}, y_{U_i}\rangle, \ldots\}$
1: $h = 1, k = 1$
2: $LD_h \leftarrow Return\_block(L\_container)$
3: $m\mathcal{F}, nm\mathcal{F} \leftarrow Feature\_selection(LD_h)$
4: $|\Phi(LD_h)\rangle \leftarrow Data\_encoding(LD_h, m\mathcal{F})$
5: $|p_{+1}\rangle, |p_{-1}\rangle \leftarrow Distance\_estimation\&Prototype\_generation(|\Phi(LD_h)\rangle)$
6: $|\Delta(LD_h)\rangle \leftarrow Feature\_aggregation(LD_h, nm\mathcal{F})$
7: $\mathcal{F} \leftarrow Binary\_classification(|\Delta(LD_h)\rangle)$
8: **while** $L\_container \neq \oslash$ *and* $U\_container \neq \oslash$ **do**
9: **if** $(LD_h$ *is ready*$)$ **then**
10: $|\Phi(LD_h)\rangle \leftarrow Data\_encoding(LD_h, m\mathcal{F})$
11: **if** $(Change\_detector(|\Phi(LD_h)\rangle, |p_{+1}\rangle, |p_{-1}\rangle)$ *raises*$)$ **then**
12: $LD_h \leftarrow Data\_replay(LD_h, \cup(LD_{h-1}, \ldots LD_{h'}))$
13: $|p_{+1}\rangle, |p_{-1}\rangle \leftarrow Distance\_estimation\&Prototype\_generation(|\Phi(LD_h) \cup \{|p_{+1}\rangle, |p_{-1}\rangle\}))$
14: **end if**
15: $h = h + 1$
16: $LD_h \leftarrow Return\_block(L\_container)$
17: **end if**
18: **if** $(UD_k$ *is ready*$)$ **then**
19: $|\Delta(UD_k)\rangle \leftarrow Feature\_aggregation(UD_k, nm\mathcal{F})$
20: $\overline{UD_k} \leftarrow Binary\_classification(|\Delta(UD_k)\rangle, \mathcal{F})$
21: $k = k + 1$
22: $UD_k \leftarrow Return\_block(U\_container)$
23: **end if**
24: **end while**

scenario is that sequential data are acquired at a regular arrival rate, which does not dictate the times of execution of either CC or QC components. Given that the data arrive in a scattered manner, without a predefined order between labeled and unlabeled instances, one cannot know in advance when the succession of CC or QC techniques to be executed or when to engage in model learning or label prediction. This is decided by the availability of labeled data and unlabeled data and by the changes underlying the data. More precisely, as the data arrive, they are separated in two different data containers, one for those labeled and the other one for those unlabeled. The use of data containers allows us to lightening the burden of the data arrival rate on the processing (lines 2, 16, and 22 in Algorithm 1)[1]. Thus, dependently on the readiness of these containers, the framework carries out either *i)* training session, in which the classification model being learned/updated (line 9). The training session involves a quantum unsupervised technique and eventually quantum supervised algorithm) or *ii)* prediction session, in which the classification model is used to estimate the class labels (line 18). Actually, in the first case (*i)*), the update of the model, through a quantum supervised approach, is activated only whether there are changes (within the current set data stored in the container of labeled data). The changes are being detected by a procedure which combines a statistical test and the quantum unsupervised technique. Note that we could accumulate labeled data which do not denote changes for a long, without the reasonable necessity of having to update the classifier. In the second case, that is, *ii)* label prediction, the quantum classifier is straightforwardly applied to the current set of data (stored in the container of unlabeled data).

The data have a temporal connotation, although no time-stamping information is used, and are characterized by valued features. The data arrive singularly, one after the other, but are processed in sequences of consecutive data instances, referred to as *data blocks*. Data blocks (LD_h and UD_k in Algorithm 1) are a notion strictly close to the time-windows models (Gama and Gaber 2007). Similarly to the time windows, data blocks allow us to handle data instances by equally-sized sequences on which we train, update, and apply the classification model. A data block can be fulfilled with data instances taken from either two containers, which means a data block contains either labeled or unlabeled data. As a data block is being filled (the number of collected data instances is equal to the predefined size), we can proceed either with a training session (in which data blocks with labeled data are used, LD_h in Algorithm 1) or with a prediction session (in which the data blocks used contain data instances with unknown label, UD_k in Algorithm 1). The succession of training sessions and prediction sessions is not predefined, coherently with the realistic assumption according to which the distribution of labeled and unlabeled data instances is not previously established and therefore not all the data instances are labeled.

As initialization of the entire process, the framework considers the first labeled data block LD_1 under the assumption that LD_1 is representative of the features (data properties) of the next data. It uses the CC component of feature selection to rank the original descriptive features $\mathcal{F}$. Then, the framework takes the first m ranked features to be used for the QC components of data encoding and distance estimation and prototype generation, while the remaining $n - m$ (by taking them by even number) for the QC components Feature aggregation and Binary classification (line 3). Data encoding converts those data instances to define the quantum state of one qubit by means of a dense angle encoding technique (LaRose and Coyle 2020) (line 4). This choice proves to be adequate for computing quantum distances between quantum states. Our preliminary results suggest considering the value of m to be as low as possible. Subsequently, the component Distance estimation & Prototype generation builds two prototypes ($|p_{+1}\rangle, |p_{-1}\rangle$), that is, representatives of the two classes, one for each class (line 5). The data block LD_1 feeds

[1] In the Algorithm 1, the notation of the double subscript X_{L_i} and X_{U_i} refers to the i-th data instance labeled (L_i) and unlabeled (U_i), respectively. By coherence of notation, the targets of the unlabeled data are reported as y_{U_1}, but it is the information to be estimated.

also the quantum supervised part, that is, Binary classification trains the classifier from scratch on the quantum states encoding the data block LD_1 through Feature aggregation.

Specifically, Feature aggregation takes as input the data instances of LD_1 described by the subset $\mathcal{S}$ of the $n - m$ previously ranked features ($\mathcal{S}$ has even cardinality). Then, it combines them with the result to map the data instances from the space $\mathbb{R}^{|\mathcal{S}|}$ to the Hilbert space with $\mathbb{C}^{\frac{|\mathcal{S}|}{2}}$ (line 6). Subsequently, the component Binary classification learns an initial classification hypothesis function through a process which optimizes the parameters of a quantum circuit underlying the component (line 7). It results in a classifier which could already estimate unknown data instances in meanwhile acquired (the data block UD_k in Algorithm 1).

After the initialization, QUARTA continues to acquire data instances and store them within the containers (lines 8, 16, 22). In cases where a data block of unlabeled instances becomes available before a data block with labeled data, a prediction session is initiated (line 18). Thus, QUARTA first selects only the valued features $\mathcal{S}$ and then encodes the data UD_k through Feature aggregation (line 19). Finally, it utilizes the classification hypothesis function so far constructed to estimate the class labels (line 20). These operations are replicated also for the data blocks, subsequent to UD_k, that consist solely of unlabeled data. Indeed, one cannot glean information on changes within the data, and thus, there is no reasonable need to update the classifier.

Conversely, in cases where a data block of labeled data instances becomes available before a data block with unlabeled data (line 9), another training session is conducted. However, not all training sessions involve a learning step, as the respective data blocks may have no change significant enough to justify an update of the classifier. To check the presence of changes, QUARTA combines a CC technique (change detector) with the QC component of distance estimation and prototype generation. Whenever a change is detected (line 11), a training session, aiming at updating the classification model, is performed by means of the components feature aggregation and binary classification. The training set is built by the component *data replay* and is composed of the data instances of the current data block LD_h and previously processed data blocks ($LD_{h-1} \dots LD_{h'}, h' \geq 1$) (line 12). The data instances of LD_h that were closer to the prototypes are discarded, those more distant used to adapt the model. Subsequently, new prototype(s) is (are) generated from a set of labeled data instance which involves the current prototype and data instances of LD_h (line 13).

It should be noted that binary classification and distance estimation and prototype generation work on different encoded feature sets (converted by means of different encoding schemes, by the way). As our knowledge, this is one of the first attempts to leverage multi-view learning in quantum machine learning. The primary challenge in multi-view learning is devising algorithms able of learning from multiple data perspectives (views) simultaneously behind the assumption that multiple views may exhibit different statistical properties and, when processing them concurrently, they can collectively contribute to enhancing predictive accuracy. This entails leveraging the diversity inherent in each feature set view while avoiding the curse of dimensionality.

3.2 The quantum-classical components of QUARTA

QUARTA is composed of QC routines and CC techniques. The former are mainly used for machine learning steps; the latter are used as data preparation tools and decision makers for the execution of quantum circuits. As preliminary operation, QUARTA scales the values of the features of the input classical data within the range of [0,1]. In preliminary experiments, we acquainted that often the convergence of the training sessions is faster when this transformation is applied compared to the typical transformation into the range $[0,\pi]$. The component is used for each incoming data block, both those of training and those of prediction.

Data replay

It operates when it is necessary updating the model and provides data for a training session. Specifically, it handles two data samples (one for each class) of the labeled data instances taken from the previously labeled data blocks $LD_{h-1} \dots LD_{h'}, h' \geq 1$ by keeping them in the order of arrival of their corresponding data blocks. Every time a new training session starts, the oldest data instances are discarded from the sample, while the same number of data instances is taken from the currently processed data block LD_h. So, the size of the container is always constant.

In the literature, data replay (also called as data rehearsal) is one of the three approaches (Lange et al. 2022), namely, *replay methods*, *regularization-based methods*, and *parameter isolation methods* designed to *i)* deal with the catastrophic forgetting effect raising when updating neural networks and *ii)* retain knowledge of previous data blocks or experiences while learning new data blocks. Considering that the replay methods represent the solution which asks for less and leaves unchanged the number of hyper-parameters of the neural network, we lean for that approach.

Feature selection

One of the preliminary actions of any machine learning project is to minimize the number of descriptive features while preserving essential information, achieved through a suitable transformation from raw data to training data. Feature selection methods can be categorized in *filters*, *embeddings*, and *wrappers* (Chandrashekar and Sahin 2014). In the current work, by following the conventional setting of the incremental learning, we resort to a filter mechanism

which works on the first data block and selects the subset of the descriptive features which we will use in the next data blocks. The filter relies on CC techniques which consider the mutual information between the class labels, as proposed in Ross (2014). Mutual information measures the degree of relatedness between variables; it detects any sort of relationship between variables whatsoever, whether it involves the mean values or the variances or higher moments. When the variables are both discrete (one of them is the class label), it estimates the true frequencies of all combinations of values. When the variables are continuous and discrete, the techniques use a weighted form of the Jensen-Shannon divergence which is used as a measure of the dissimilarity between two or more continuous probability distributions.

Data encoding

Machine learning techniques leverage feature maps to project input (classical) data into distinct spaces, aiming to attain advantageous properties of the new spaces. Quantum encoding is a method used to encode classical data $a \in \mathbf{X}$ into quantum states within a Hilbert space $|\phi\rangle \in \mathbf{H}$ through the use of a quantum feature map $\Phi : \mathbb{R} \rightarrow \mathbf{H}$, that is, $a \rightarrow |\Phi(a)\rangle$. The quantum feature map is equivalent to applying a unitary transformation $\mathbf{U}_\Phi(a)$ to the default initial state $|0\rangle$ to produce $|\Phi(a)\rangle$. Such unitary transformations often involve parameterized gates, where the parameters often take the (normalized) values of the features of classical data (Lloyd et al. 2020; Schuld and Killoran 2019).

There are various techniques to design data encoding, from the simpler ones like amplitude and angle to those more complex, such as Hamiltonian embedding and re-uploading (Pérez-Salinas et al. 2020). Although those more sophisticated may contribute to improving the expressiveness of quantum circuits and accuracy of data representation, they require additional computational times, which in a scenario of continuous computation is not desirable. This is the reason why we propose angle encoding techniques for the distance estimation (Section *Quantum distance estimation & Prototype generation* below). Angle encoding relies on parameterized rotation gates R_k and maps a single real-value a of a classical feature into a quantum state identifiable in the geometrical representation of the Bloch sphere (Nielsen and Chuang 2011):

$$a \rightarrow R_k(\psi) = e^{-i\frac{\psi}{2}\sigma_k} \tag{1}$$

with k=x, y, z axis of the three-dimension space of the sphere, σ_k is the Pauli matrix representation of the rotation gate R_k, ψ is the normalization of the classical data value a and is the parameter used for the angle of rotation in R_k.

In the Bloch sphere, a quantum state can be expressed as a vector denoted with two spherical coordinates $0 \leq \theta < \pi$, $0 \leq \phi < 2\pi$ corresponding to the angle of *magnitude* (drawn with respect to the z-axis) and angle of phase (drawn with respect to the x-axis).

The angle encoding technique proposed here is designed to better leverage the Hilbert space. The circuit is composed of the gates R_y and R_z. By thinking in terms of Bloch sphere, R_y applies a rotation which leads the vector to range between the basis states and therefore contributes to the encoding through a magnitude angle specific for the parameter. R_z applies a rotation which by itself does not change the magnitude (and therefore the amplitudes) but contributes to the encoding through an appropriate relative phase specific for the parameter. The relative operator can be so formulated.

$$\mathbf{U}_\Phi(\mathbf{X}) = \left(R_z(\psi^m)R_y(\psi^m)\right)\ldots\left(R_z(\psi^j)R_y(\psi^j)\right) \ldots\left(R_z(\psi^1)R_y(\psi^1)\right)H|0\rangle \tag{2}$$

with ψ_i parameter which takes the normalized value of $x_i \in \mathbf{X}$ out of the m-selected features.

In simpler terms, the operator enqueues gate blocks composed of two rotational gates, specifically $R_z()$ and $R_y()$, on the same qubit, for each valued feature in $\mathbf{X}$. A gate block allows us to exploit the range $[0, \pi]$ for the magnitude and associates a periodicity of π to the magnitude. Also, it gives a wider variability to relative phase, both over $[0, \pi]$ and over $(\pi, 2\pi]$. One can experimentally prove that this does not happen even with other rotation-based encodings built upon two gates. For instance, the variant of the operator (2) in which the rotations are inverted, that is, $R_y(\psi_i)R_z(\psi_i)H$, associates a range shorter than $[0, \frac{7}{10}\pi]$ to the magnitude. This choice guarantees that each unique input x will have a unique quantum encoding without requiring an arbitrarily large number of qubits but related to the number of features representing the data instance. We also conjecture that this choice augments the discriminatory power of the component of quantum distance estimation and prototype generation.

Feature aggregation

To handle the presence of the *curse-of-dimensionality*, one can resort to feature aggregation or dimensionality reduction, which transform high-dimensional data into a lower-dimensional representation, leveraging sparsity and redundancy. Feature aggregation, traditionally originating from multimedia data processing, involves the fusion of highly correlated low-level features to define high-level features. This provides a more abstract view of the properties of the data (Guo et al. 2021). Dimensionality reduction aims to project the data into a different feature space, giving priority to data correlation rather than robustness and outlier sensitivity (Yu et al. 2023 Lloyd et al. 2014). However, dimensionality reduction builds new representational spaces where the new features combine linearly those original while neglecting non-linear correlations. This is the reason why we propose a hierarchical feature aggregation offered by the neural-based operation pooling (Galanis et al. 2022). Indeed,

the pooling operator may keep under control the generation of the new features by constructing progressively new features over a hierarchical representation. Additionally, it improves the robustness to small translations or shifts in the data and, specifically for quantum solutions, allows to handle the problem of representing *one-feature with one-qubit* when the features are numerous. The pooling layers are used prominently to design convolutional neural networks thanks to the capacity of selecting representative values from neighboring points, thus reducing the computational load in subsequent layers and extracting high-order correlations (Hur et al. 2022). In QC, the pooling techniques are typically used in quantum convolutional neural networks to reduce the dimensions especially when the number of qubits equals the number of the features. The blueprint is of tracing out half of the qubits, while retaining as much information as the other half of the qubits. In fact, the pooling operator compresses the pair of features on two qubits into one qubit, so the information of a pair of quantum states is conveyed on one of the two qubits. The disregarded qubit will be neglected in the future calculations or measurements for the rest of the quantum circuit. **[R2A4]** The operator can be seen as a quantum feature map $\Delta : \mathbb{R}^2 \rightarrow \mathbf{H}$ which encodes a pair of classical data $x_i \in \mathbf{X}, x_{i+1} \in \mathbf{X}$ into quantum states within a Hilbert space $|\Delta\rangle \in \mathbf{H}$. For the purpose of this work, we refer to the structure proposed in Vatan and Williams (2004), which has also been proven to be optimal (Madden and Simonetto 2022). The optimality ensures that no smaller circuit, using the same family of gates, can achieve the same unitary operation. The circuit is based on a two-qubit design built upon controlled-not and single-qubit parameterized gates. The corresponding unitary operator can be formulated as follows.[2]

$$\mathbf{U}_{\Delta(x_i,x_{i+1})} = (I^i \otimes {R_y}^{i+1}(\lambda))\; CX^{i,i+1}\; (R_z^i(\omega) \otimes R_y^{i+1}(\phi))$$
$$CX^{i+1,i}\; (I^i \otimes {R_z}^{i+1}(-\frac{\pi}{2}))|0^2\rangle \qquad (3)$$

with ω and ϕ parameters which take the normalized values of the classical data $x_i, x_{i+1} \in \mathbf{X}$. The parameter λ has value as (1-x_i) (1-x_{i+1}). Each gate $R_y()$ and $R_z()$ is a specific case of the generic Pauli rotation gate R_k above introduced and work similarly. The two-qubit gate $CX^{i,i+1}$ has the control qubit denoted as q_i and target qubit denoted as q_{i+1} (analogously for $CX^{i+1,i}$).

In simpler terms, the circuit implementing the operator (3) establishes a dependence between the quantum state of q_i and quantum state of q_{i+1}, that is, the quantum state of q_{i+1} would affect the one of q_i. Then, the qubit q_{i+1} effectively traces out from the circuit, aggregating the information conveyed by two features on the two qubits. Specifically, the gates R_z induce a phase change in both quantum states, while $R_z^i(\omega)$ operates dependently on the parameter ω that takes the value from the feature associated to q_i. On the other hand, the gate $CX^{i+1,i}$ establishes a relationship between the "portion" of the base state $|1\rangle$ of q_{i+1} and the quantum state of q_i. This relationship is reinforced by the action of the gate $R_y^{i+1}(\phi)$, inducing a change in the magnitude of qubit q_{i+1}. Consequently, due to the influence of $CX^{i+1,i}$, the magnitude of qubit q_i is also altered. Subsequently, $CX^{i,i+1}$ plays a crucial role in solidifying this interdependence in the reverse direction, functioning akin to $CX^{i,i+1}$. Intuitively, in this context, the quantum state of qubit q_{i+1} assimilates information from the quantum state of qubit q_i. This interaction is ultimately guided by qubit q_{i+1} through the magnitude alteration brought by the ${R_y}^{i+1}(\lambda)$.

It is important to emphasize that in the current study, this circuit differs from the conventional use of standard *ansatz*, where parameters are typically fine-tuned through an optimization process. Instead, the parameters λ, ω, and ϕ are contingent upon the input data and remain fixed without undergoing any tuning operation.

The component of feature aggregation is completed with the assignment of pairs of the valued features in $\mathbf{X}$ to consecutive qubits (q_i, q_{i+1}). Each feature is paired with the one with which has higher value of the Kendall Tau correlation (Kendall 1938). Both features contribute to the state preparation of the operator formulated at the (3). Kendall Tau correlation is a statistical measure used to quantify the degree of association or correlation between two variables. It is particularly useful when dealing with ordinal or ranked data, where the actual numerical differences between values may not be meaningful. Thus, the calculation of such correlation, implemented in this study as classical computing operation, allows us to assess the similarity of the orderings of values between two features, regardless of their actual values.

Finally, we should note that not all the original features are aggregated pairwise by this component, but only the subset of $n - m$ remaining features identified by feature selection (the first m features have been used for data encoding). Also, as anticipated, we consider the even number lower than but closest to $n - m$. This clarifies that feature aggregation embeds as many pooling circuits (operator 3) as the pairs of features are. Feature aggregation does not encode data instance for all the data blocks, but only when changes are detected in the current data block.

Distance estimation and prototype generation

Distance computation is a frequent operation in machine learning, holding significant importance in both unsupervised and supervised methods. In the literature of QC, a measure of distance is the *trace distance* (Nielsen and Chuang 2010), which is related to the maximum probability of distinguishing between two quantum states through

[2] In the notation of operator (3), the superscripts i and $i+1$ refer to the order of the qubits in the input qubit register, hence q_i and q_{i+1}.

measurement. The notion of distance is closely related to the (dis)similarity between quantum states. Conceptually opposite to the distance, there is the *fidelity* (Jozsa 1994) that quantifies the similarity between two quantum states. These two measures offer quite desirable properties, such as non-negativity and symmetry, which enable to work on metric spaces. Indeed, quantum states can be represented as points in a quantum metric space, where the distance between two quantum states captures their distinguishability or dissimilarity which is fundamental for tasks such as quantum clustering and classification (Havlícek et al. 2019).

In the current work, the concept of distance is closely linked to the update of the classification model. Therefore, generally speaking, we assert that quantum distance estimation supports the training process of a classifier constructed using quantum solutions. Going deeper, the distance aids in detecting changes and determining whether the classification model needs updating. Thus, we apply the distance metric to new labeled data instances in comparison to those previously processed to identify potential changes. This enables us to ascertain whether the classification model employed thus far is suitable for these data instances or if it requires adaptation.

To do that, we resort to the notion of the quantum fidelity that measures the similarity, which is complimentary to the distance. The distance between quantum states can be immediately derived from the similarity by using inverse transformation functions. The fidelity acts as an overlap function by returning two results, one representing the case in which the two states are identical and the other one representing the case in which they are orthogonal. It can be implemented through the so-called *Swap test* operator (Barenco et al. 1997) defined on three qubits as here reported:

$$(H^1 \otimes I^2 \otimes I^3) CSWAP^{\{1\}\{2,3\}} (H^1 \otimes I^2 \otimes I^3) \tag{4}$$

The operator consists of single-qubit gates (H and I) and a 3-qubit gate, $CSWAP$,[3] responsible for exchanging the states between the qubits indexed as 2 and 3 under the control of the qubit indexed as 1. As to the formulation (4), the qubits 2 and 3 encode the two quantum states, while qubit 1 serves as an ancillary qubit initialized to $|0\rangle$. The fidelity value is determined by measuring qubit 1, yielding outcome 0 with a probability of 1 if the two quantum states are identical. Otherwise, if the two quantum states are orthogonal or have no overlap, the outcome is 1 with a probability of 0.5 (Kavitha and Kaulgud 2023).

However, such a dichotomy does not fully resolve the problem of detecting changes, as it would restrict to searching for changes only when the outcome is $|0\rangle$ with a probability of 1. Instead, we propose utilizing fidelity in a frequentist approach, which enables us to *i)* tie the similarity of the quantum states to statistics of the resulting basis states and *ii)* quantify the magnitude of changes within a variability range (Acampora et al. 2021). To compute the frequencies, we need to collect the results of measurement outcomes. Therefore, we execute the operator *Swap test* multiple times, with each execution yielding one of several possible measurement outcomes. Consequently, after completing the specified number of executions, we obtain counts for each possible outcome. From these counts, we compute the relative frequency as the ratio of each count to the total number of executions. In essence, the counts represent the distribution of occurrences of the outcomes, and the ratios relative to the total occurrences resemble probability values for the outcomes, with their sum totaling 1. Since the measurements for the *Swap test* concern only the ancillary qubit, we will have the relative frequencies for the two outcomes, that is, 0 and 1. The outcome 0 with probability 1 represents the full overlap of the two quantum states, so we consider the relative frequency computed for the outcome 0 as similarity between the two states. A brief example follows. Suppose the total number of executions be 512, with 400 times turning on the outcome 0, while 112 time turning on the outcome 1. So, the estimation of the similarity is 400/512 and consequently the distance is $\frac{1}{similarity}$.

Once introduced the notion of similarity, we can explain how it is used to support the update of the classification model. This concept of similarity, and specifically its complementary concept of distance, is utilized to generate two prototypes, one for each class of the binary classification problem under consideration. These prototypes serve as representatives for their respective classes and are constructed by synthesizing the properties of the data instances associated with each class. They are computed as *medoids* of the data instances of each class and correspond to the data instance $\mathbf{X}$ whose sum of distances with all the others $\mathbf{Y}$ of the same class C reaches minimum (Aïmeur et al. 2013):

$$\sum_{\mathbf{Y} \in C} dist(\mathbf{X}, \mathbf{Y}) \leq \sum_{\mathbf{Y} \in C} dist(\mathbf{Z}, \mathbf{Y}), \forall \mathbf{Z} \in C \tag{5}$$

Thus, the distance is computed between the medoids and labeled data instances currently processed. Indeed, this allows us to verify whether these data instances adhere to the properties of their respective classes (represented by their medoids) or whether they exhibit significant changes necessitating to update the classification model.

Both when determining the medoids and when computing the distances between medoids and data instances, the calculation concerns quantum states built by the component data encoding. Indeed, the operator at the (2) performs a state

[3] The notation $CSWAP^{\{1\}\{2,3\}}$ in the (4) indicates that the superscript 1 denotes the control qubit, while 2 and 3 denote the two target qubits of the controlled-swap gate, also known as Fredkin gate (Patel et al. 2016)

preparation operation outputting the quantum states $|\phi\rangle$ and $|\psi\rangle$ for the Swap test operator.

Finally, it is worth noting that the search for prototypes described above resembles the construction of centroids in quantum clustering techniques (Benlamine et al. 2020; Kavitha and Kaulgud 2023; Kerenidis et al. 2019). In those works, centroids are determined using classical computing (CC) methods, which involve calculating distances between data instances in their original representation. In contrast, the proposed concept of medoids relies on quantum distance computation between quantum states.

Change detector

Change detection is a topic investigated in machine learning both through unsupervised techniques and those supervised. However, the latter require training data and learning step, which in a scenario of continuous computation is not desirable. This is the reason why we propose an approach inspired to the unsupervised setting. The component change detector exploits the Page-Hinkley statistical test (PH) (Page 1954), originally defined to identify sudden changes on a variable monitored over time or deviations from the expected behavior. The Page-Hinkley test is a non-parametric method and does not require any assumption about the underlying distribution of the data. Specifically, the test evaluates the cumulative sum of the deviations between the observed variable and the expected mean. In this work, the variable observed over time is the *sum of the squared errors* (SSE), which is traditionally utilized in clustering to denote the internal homogeneity of a cluster. It measures the extent to which data instances within a cluster deviate from the prototypes of that cluster. However, unlike its conventional usage, SSE here denotes the homogeneity of the labeled data instances included in the data blocks processed so far, for either classes. Specifically, SSE works on quantum states and uses the notion of quantum distance (explained in Section *Distance estimation & Prototype generation*). The value of SSE computed between the prototypes obtained before processing the data block LD_h and the data instances of LD_h is so formulated:

$$SSE_h^y : \sum_{instance_{h_j}^y \in LD_h} dist(prototype_h^y, instance_{h_j}^y), y \in \{+1, -1\} \quad (6)$$

Thus, after having processed the data block LD_h, we update the mean value $\overline{SSE_h^y}$ including the values of SSE_h^y computed on the previous data blocks, cumulative sum of the deviations Δ_h, and minimum value of the deviations Δ_h^{MIN}:

$$\overline{SSE_h^y} = \frac{1}{h}\sum_{t=1}^{h} SSE_h^y$$

$$\Delta_h = \Delta_{h-1} + (SSE_h^y - \overline{SSE_h^y}), t = 1 \ldots h, \Delta_0 = 0 \quad (7)$$

$$\Delta_h^{MIN} = \min_{t=1,\ldots,h} \Delta_t$$

Once a labeled data block has been processed, the test checks for the discrepancy between the cumulative sum and the minimum value $\Delta_i - \Delta_i^{MIN}$. When this difference exceeds a certain threshold τ, even if it pertains to only one class, it triggers a training session to update the classification model.

Binary classification

The component is in charge of training a classification model in a supervised setting and keeping it update to deal with changes related to the data referred to the two class labels, without possibly forgetting the old data properties of the same two classes. It operates in two alternative modalities, training and prediction. During the training sessions, the classification model learns on the block of labeled data instances currently acquired. The training procedure is not invoked on all the data blocks, but only when changes associated to labeled data instances are being detected. Differently, during the prediction sessions, the model is used to infer the class labels on the unseen data instances.

To train the model, we rely on a *variational quantum circuit* (VQC), that is, a class of hybrid classical-quantum methods based on a quantum circuit with learnable parameters optimized through classical algorithms (Benedetti et al. 2019). Similarly to the classical neural networks, their crucial peculiarity is the capability to approximate any continuous function (Arthur and Date 2022).

From the procedural viewpoint, a VQC works as follows. The quantum circuit prepares a quantum state, guides its evolution, and performs measurements on it. The outcomes of these measurements undergo post-processing by CC techniques to generate estimations or predictions. These estimates are refined by means of a classical optimization algorithm that updates the model parameters, aiming at minimizing a cost function defined on those parameters. The updated parameters are then injected into the circuit, resulting in improved estimations. This entire process operates in a closed loop between classical and quantum algorithms, leading to an improved model represented by a refined-parameters quantum circuit, ready to process data that have not been used in the loop.

From the structural viewpoint, a VQC is composed of three main constituents: *i)* an encoding circuit designed with quantum gates (typically with fixed parameters) and in charge of representing classical data into quantum states to be managed by the parameterized part; *ii)* a composite gate block (also called as *ansatz*) built upon the parameterized part and in charge of approximating the cost function; *iii) observables* that will be considered as circuit outputs.

As to the encoding circuit, we use the component feature aggregation described above that comprises also angle encoding transformations. As to the ansatz, it is important to say that the design of a specific-by-problem solution is crucial as it impacts the handling of vanishing gradients in classical optimizers and determines the balance between circuit depth, model accuracy, and training time. A promising and straightforward approach seems to be considering "template" quantum circuits that previous research has shown to be effective for training tasks and classification problems. To this regard, the studies of Sim et al. (2019) and Hubregtsen et al. (2021) report experimental evaluations to assess the utility of selected circuit structures based on two quantitative descriptors: *expressibility* and *entangling capability*. The first descriptor denotes the ability to generate pure quantum states capable of fully exploring and representing the Hilbert space. The second descriptor refers to the ability to generate entangled states aimed at exploring the solution space for classification problems while capturing non-trivial correlations. Drawing inspiration from these studies, we identify two circuits that have shown performing relatively. Considering the input qubit register with four qubits, we can formulate these two circuits as follows:

$$\begin{aligned}\mathbf{A}(\Omega,\lambda) = (R_y{}^{i}(\Omega_0)\otimes R_y{}^{i+1}(\Omega_1)\otimes R_y{}^{i+2}(\Omega_2)\otimes R_y{}^{i+3}(\Omega_3))\\ (CX^{i+3,i})\otimes CX^{i+2,i+3}\otimes CX^{i+1,i+2}\otimes CX^{i,i+1})\\ (R_y{}^{i}(\Omega_4)\otimes R_y{}^{i+1}(\Omega_5)\otimes R_y{}^{i+2}(\Omega_6)\otimes R_y{}^{i+3}(\Omega_7))\\ (CX^{i+3,i+2}\otimes CX^{i,i+3}\otimes CX^{i+1,i}\otimes CX^{i+2,i})|0^{\otimes 4}\rangle\lambda\end{aligned} \tag{8}$$

with λ of size 4 denoting a set of four quantum states (resulting from the component feature aggregation on eight selected features), Ω of size 8 denoting the parameter set of the circuit (one parameter for each rotation gate $R_y()$). The term Ω represents the parameters that are tuned by classical algorithm of optimization.

The second circuit has a larger set of parameters and presents two-qubit gates based on rotational gates $R_x()$:

$$\begin{aligned}\mathbf{A}(\Omega,\lambda) = (R_y{}^{i}(\Omega_0)\otimes R_y{}^{i+1}(\Omega_1)\otimes R_y{}^{i+2}(\Omega_2)\otimes R_y{}^{i+3}(\Omega_3))\\ (CRX^{i+3,i}(\Omega_4)\otimes CRX^{i+2,i+3}(\Omega_5)\otimes CRX^{i+1,i+2}(\Omega_6)\otimes CRX^{i,i+1}(\Omega_7))\\ (R_y{}^{i}(\Omega_8)\otimes R_y{}^{i+1}(\Omega_8)\otimes R_y{}^{i+2}(\Omega_{10})\otimes R_y{}^{i+3}(\Omega_{11}))\\ (CRX^{i+3,i+2}(\Omega_{12})\otimes CRX^{i,i+3}(\Omega_{13})\otimes CRX^{i+1,i}(\Omega_{14})\otimes CRX^{i+2,i+1}(\Omega_{15}))|0^{\otimes 4}\rangle\lambda\end{aligned} \tag{9}$$

with $CRX^{control,target}$ denoting the two-qubit controlled gate with a rotation gate $R_x()$ operating on the control qubit, λ is the same as above, Ω of size 16 denoting the parameter set of the circuit (one parameter for each rotation gate $R_y()$ and controlled rotation-x gate $CRX()$).

The operators (8) and (9) are both organized in two main blocks, each containing single and two-qubit gates. The single qubit gates are parameterized rotation gates, denoted as $R_y()$, which, in the representation of the Bloch sphere, induce changes in the magnitudes of the state vectors. A common characteristic is the utilization of qubits in controlled gates, where they serve both as controls and targets. This choice aims to capture various forms of correlation between pairs of selected features encoded by the respective qubits. Additionally, two-qubit gates are employed to establish a cyclic connectivity pattern, wherein nearest-neighbor interactions occur between consecutive qubits, supplemented by non-local interactions to complete the cycle. Contrary, the two circuits differ in the usage of two-qubit gates. In fact, the operator in the formula (9) incorporates parameterized two-qubit gates. These gates potentially enhance the variability of states, leading to a richer representation of input data compared to the fixed two-qubit gates of the circuit in the formula (8). Also, we can note that the circuit in the formula (9) presents a larger number of parameters than that in the formula (8). This could lead to increased complexity in the optimization process and longer computation times.

To operate on a set of qubits larger than four, we extend the circuits while preserving both their original design decisions and the operativity of the gates across the entire block (Schuld et al. 2020). Specifically, we propose to maintain the initial stack of rotational gates $R_y()$ and introduce consecutive controlled gates, with each connected to the previous one. More precisely, once a controlled gate is inserted, the control qubit becomes the target qubit for the subsequent controlled gate. This process is replicated for both blocks within each circuit. This approach aims to propagate the effect of a qubit not only to adjacent qubits but also to those further away, thereby attempting to capture non-local correlations.

Finally, the quantum state provided by the ansatz undergoes measurements and needs to be decoded to obtain the desired output (estimation of the class label). This entails finding an optimal mapping of the outcomes of measurements into the class labels. The usual way is reading an observable of the quantum circuit, that is, measuring the final quantum state of one or more qubits through a Pauli spin operator (a particular axis in the Bloch sphere representation). In this work, we consider the Pauli-Z σ_z operator (namely, the hermitian matrix of σ_z) and use it to calculate the expectation value $\langle\Phi|\sigma_z|\Phi\rangle$ ($|\Phi\rangle$ denotes the final quantum state of the

ansatz, and $\langle\Phi|$ denotes its complex conjugate). The expected value can be approximated by taking the result of $\langle\Phi|\sigma_z|\Phi\rangle$ across multiple runs of the circuit.

The operator σ_z returns either the value +1 or the value -1, representing even or odd parity, respectively. Parity refers to the count of qubits resulting in either $|0\rangle$ or $|1\rangle$. Thus, when calculating the expectation value, it will fall within the range [-1, +1]. We utilize this value to establish the mapping of the circuit output into the class labels. Thus, the expectation value of the circuit quantifies the probability of a data instance x to belong to either class:

$$P(y|x) = \frac{y\mathcal{E}(x,\Omega)+1}{2} \tag{10}$$

where y represents the class labels $-1, +1$, and $\mathcal{E}(x,\Omega)$ denotes the expected value over the qubit register of the ansatz sized as n:

$$|\mathcal{E}(x,\Omega) = \langle\phi(x,\Omega)|\sigma_z^{\otimes n}|\phi(x,\theta)\rangle \tag{11}$$

The probability $P(y|x)$ is utilized in the optimization process of the parameters Ω. The classical optimizer iteratively updates the circuit parameters by minimizing a cost function that is based on the negative log-likelihood of the probabilities $P(y|x)$ computed on the current labeled data block, that is,

$$-\frac{1}{|datablock|}\sum_{i=1}^{|datablock|} log(P(y_i|x_i)) \tag{12}$$

where $|datablock|$ is the number of data instances of the data block, and x_i is the i-th data instance of the data block.

The cost function is minimized by the classical optimizer based on gradient. The derivative concerns the expectation value $\mathcal{E}()$ with respect to the current values of Ω and is computed by means of the parameter shift rule (Wierichs et al. 2022):

$$\frac{d\mathcal{E}}{d\Omega} = \frac{\mathcal{E}(\Omega : k+\epsilon) - \mathcal{E}(\Omega_k - \epsilon)}{2} \tag{13}$$

The gradient value is the difference between the two output values of the circuit: the first value is the output of the circuit with the parameter Ω_k increased by a value ϵ, and the second value is the parameter Ω_k decreased by ϵ.

4 Experiments

We evaluated the proposed framework QUARTA on two public real-world datasets, namely precisely

- *Ozone level detection*.[4] (having 2536 data instances, 73 real-valued features)- *Ozone* for brevity. It contains timestamped data related to the detection of ozone levels in the atmosphere, typically meteorological measurements (e.g., temperature, humidity, wind speed) and other environmental variables. Ozone is used to develop machine learning models that can accurately predict the presence or absence of high ozone levels based on the available features.
- *Spambase* (having 4600 data instances, 57 real-valued features).[5] It contains features extracted from timestamped emails, such as word frequencies, character frequencies, and other attributes, along with labels indicating whether each email is classified as spam or non-spam.

QUARTA has been implemented in Python 3.9 and IBM Qiskit SDK 0.7 (Anis 2021) and tested on the simulator *WashingtonV2* (a fake 127 qubit backend). The following available implementations were used: library Scikit-learn v1.4 (Pedregosa et al. 2011) for the CC component feature selection in the default setting; optimization algorithm ADAM for the QC component binary classification with batch size as 30 and learning rate as 0.01. The number of training epochs fixed to 30.

4.1 Algorithms

Two algorithms have been utilized for comparative analysis. The first algorithm, named *PerceptronMask*, is built on classical computing neural perceptrons (Montiel et al. 2018). The second algorithm, based on a quantum computing approach proposed in Situ et al. (2022), was originally designed for task-incremental learning, where the model is trained on a sequence of distinct classification problems and the learned is signaled when the data instances of a task end and start those of the next task. Therefore, for a fair comparison, we adapted it to the domain-incremental setting.

In addition to these algorithms, we evaluated two variants of QUARTA, each involving different design decisions regarding the quantum components. The first variant, referred to as fm_QUARTA, replaces feature aggregation with a basic feature mapping technique. This technique encodes each qubit using the operator $R_y(\lambda)H|0\rangle$, where λ represents the feature value. The purpose of this variant is to assess the impact of a quantum-based feature compression technique.

The second variant, referred to as ce_QUARTA, differs from QUARTA in the design of the Change detector built upon CC techniques only. Specifically, it maintains the orig-

[4] https://archive.ics.uci.edu/ml/datasets/ozone+level+detection.

[5] https://archive.ics.uci.edu/ml/datasets/Spambase.

inal representation of the data instances, computes prototypes as centroids for each class, and uses the CC notion of Euclidean distance. The objective of this variant is to assess the efficacy of a quantum-based technique for prototype generation.

4.2 Experimental setting and performance measures

As to the settings of the competing algorithms, we aligned the input parameters of PerceptronMask with those of QUARTA, maintaining consistency across the comparison. Parameters such as window size and learning rate were set to identical values, while other parameters retained their default settings. As for QC_GEM, we configured the depth of the variational quantum circuit to 1 and conducted training for 50 optimization cycles. The number of "data instances from past experiences" matched the size of the data sample used for data replay of QUARTA. In the QUARTA setting, we initially select the top 50% highest-ranked features from both datasets. Then, we retained the first five for *Ozone* and the first three for *Spambase* as the m-features used for the components data encoding and distance estimation and prototype generation. Subsequently, for both datasets, we considered the remaining feature set from the top 50% (excluding the m-features removed previously) with a cardinality that is a multiple of 8. These additional features constitute set S for the components of feature aggregation and binary classification. Specifically, for both datasets, S comprises either 16 or 8 features, resulting in 8 or 4 as size of the qubit input register for the binary classification. The parameters for the data block size and threshold τ of change detector were adjusted to examine their impact on the results. The data block sizes were chosen to process the flow of data with portions which go from 10% to the 1% of the total length, resulting in the sizes 400, 250, 150, and 50 for Spambase and 250, 150, 100, and 25 for Ozone. The experimental values for the change threshold τ were determined through preliminary experiments and set to 0.1 and 5 as boundary values as to identity two main levels of changes. The threshold τ=0.1 allows to trigger model update in correspondence of changes greater than 0.1, that is, milder and stronger. The threshold τ=5 leads to update the model only when the changes are particularly intense. Additionally, the data sample size for data replay was set to half the data block size. The quantum circuits with measurement operations were run by 512 times.

As evaluation scheme, following the principle of the sequential data where labeled and unlabeled data are not available as a batch, but become available incrementally and without a predefined order, we considered the prequential evaluation (Gama et al. 2013), which is an approach very often used for the evaluation of CC learner performances in the SL, OL, and CL learning scenarios. Prequential evaluation sequentially combines training data (labeled data used during training sessions) and testing data (labeled data with hidden labels during prediction sessions). This approach allows us to assess the reliability and generalizability of the classifier on unseen data that has not yet been acquired. When new labeled data blocks (LD) become available, training sessions begin, during which the classifier is trained on these data. Subsequently, when new unlabeled data blocks (UD) are ready, prediction sessions commence. During these sessions, the classifier estimates labels and the hidden labels are retained for evaluation. Once processed, the testing data are discarded and not used for updating the classifier or subsequent evaluations. This evaluation approach enables us to assess the classifier performance as it encounters new data and ensures that the model performance is evaluated on data that closely reflects its operational environment. For evaluation purposes, we prepared the datasets into two portions: 60% of the labeled data instances are allocated for training sessions, while the remaining 40% are reserved for prediction sessions, where the data instances are unlabeled. As to the performance measures, we collect the *i)* F1-score value computed once each prediction session has been completed for QUARTA and the two competitors. *ii)* Average F1-score as synthesis of the performances for some variants of QUARTA. *iii)* Loss function calculated once every labeled data block has been processed in certain variants of QUARTA. This encompasses data blocks that prompt model updates as well as those falling below the change threshold.

The experimental results have been collected and presented in order to give empirical evidence to the following research questions:

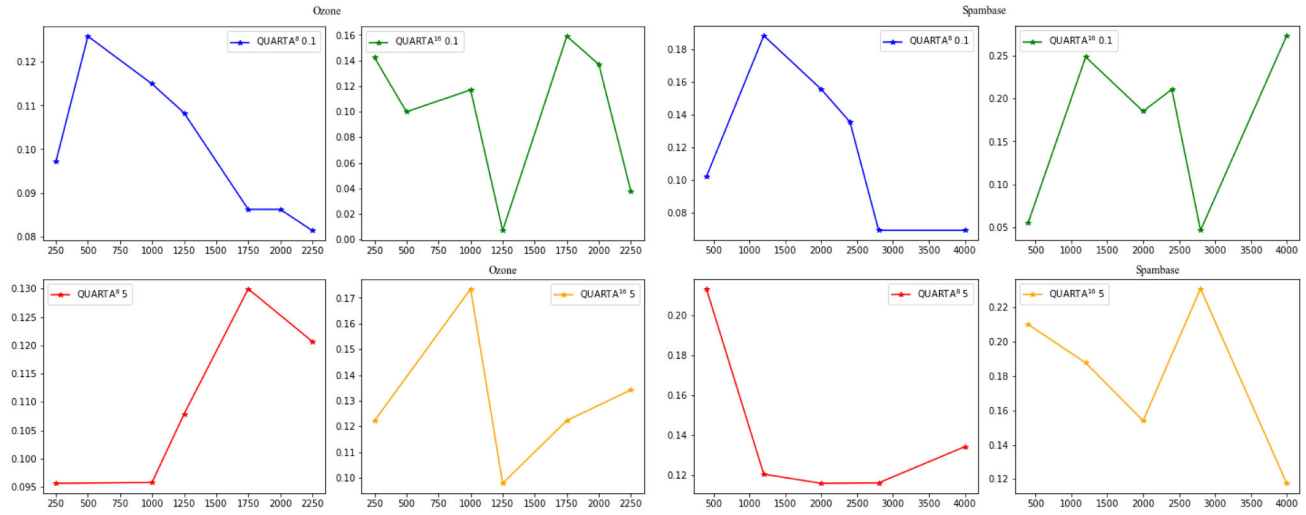
- **Q1)** Evaluate the design of the VQC on the training capability of the binary classification
- **Q2)** Evaluate the performance between prototypes generated using QC versus CC methods
- **Q3)** Compare QUARTA against both the CC solution PerceptronMask and QC competitor QC_GEM
- **Q4)** Responsiveness of QUARTA to the data arrival rate

4.3 Experimental results

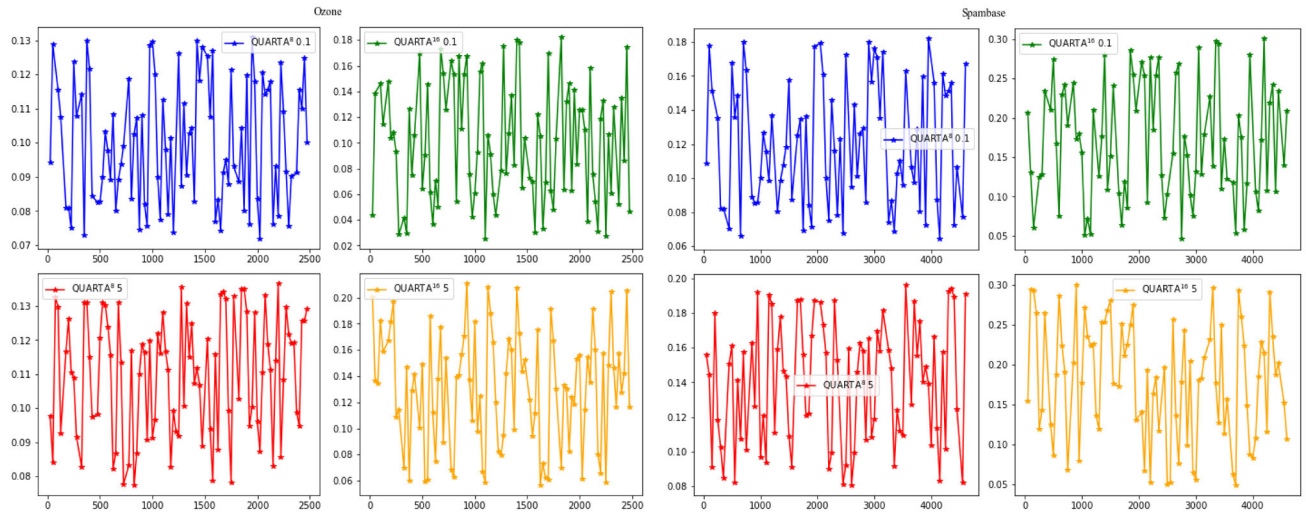
Q1. Evaluate the design of the VQC on the training capability of the binary classification

The first experiment concerns the study of the two alternative operators reported in the (8) and (9) for the component binary classification. We name the two variants as QUARTA^8 (that is, QUARTA executed with the operator at the formula (8)) and QUARTA^{16} (that is, QUARTA executed with operator at the formula (9)). The results will allow us also to identify the variant we can use for the remaining research

questions. In Fig. 1, we present the values of the loss function associated with the cost function defined in (12), projected onto the dataset. Obviously, fewer values (and thus fewer training sessions) are observed when the data block size is larger (Fig. 1a). These values are computed at the conclusion of each training session. We observe that the solution with VQC using 8 parameters (QUARTA^8) exhibits greater stability, as evidenced by fewer fluctuations compared to the model with VQC using 16 parameters (QUARTA^{16}). Specifically, the loss values tend to lie within shorter ranges: [0.08, 0.12] (Ozone), [0.08, 0.18] (Spambase) for larger data blocks, and [0.07, 0.13] (Ozone), [0.06, 0.18] (Spambase) for smaller data blocks. A general decreasing trend at larger data blocks is particularly evident. However, despite the larger fluctuations, QUARTA^{16} occasionally exhibits episodes of very low loss, even surpassing QUARTA^8. This can be attributed to the optimization process of QUARTA^{16}, which struggles with a larger number of parameters compared to QUARTA^8, yet occasionally manages to effectively minimize the cost function. In Fig. 1, we also observe the contribution of the data replay component, which is the first technique employed to address catastrophic forgetting in this work. In the plots at the top, as data instances flow, the loss does not consistently worsen; instead, it is occasionally reduced. This suggests that data replay operates more effectively with a larger pool of data instances, as it can draw from a wider and more diverse range, compared to situations where data blocks are smaller (plots at the bottom), and thus cannot retain pre-



(a)



(b)

Fig. 1 Loss function values of QUARTA^8 and QUARTA^{16} computed on the training sessions with data block size equal to 250 and 400 (**a**), 25 and 50 (**b**). The threshold τ is 0.1 and 5 for both

viously achieved training capabilities. Conversely, the loss ranges are more restricted compared to those in the plots at the top, indicating that working with fewer data instances generally leads to a reduction in loss, although the trend is not consistently decreasing. For the next research questions, we will adopt the variant QUARTA[8], given its superior performance observed in this analysis.

Q2. Evaluate the performance between prototypes generated using QC versus CC methods

This part of the experiments was conducted with the aim to evaluate how two notions of prototypes (centroid *versus* medoid) used by the change detector, implemented in different computational paradigms, affect the performance of a QC classifier. The assessment of the effectiveness of the change detection to reveal changes is beyond this experiment. QUARTA (based on a QC medoid) and ce_QUARTA (based on a CC centroid) were executed with the same data block sizes used for ***Q1***. Also in this case, the change threshold was set at the lowest value of 0.1 and highest value of 5, and feature aggregation compresses 8 features into 4. The results of the average F1-score are reported in Fig. 2. It can be observed that the use of a dense angle encoding to represent class prototypes (and relative data instances) combined with a quantum-based similarity notion is advantageous on both datasets. Indeed, QUARTA has better performances than ce_QUARTA both when working on 25 and 50 data instances at time (0.82 vs 0.72 and 0.73 vs 0.64 for Ozone, 0.78 vs 0.67 and 0.73 vs 0.67 for Spambase) and when processes a larger set, that is, 400 and 250 (0.78 vs 0.7 and 0.73 vs 0.68 for Ozone, 0.727 vs 0.66 and 0.702 vs 0.66 for Spambase). While there are slight differences, this phenomenon occurs with both stronger changes and those that are less pronounced but still significant. This can be attributed to the different nature of the two distances. The CC Euclidean distance is determined by the original feature values and therefore may suffer from high variability, especially when works on stronger changes (τ=5). Differently, the frequency-based distance, proposed in Section Distance estimation and Prototype generation, is tied to the number of occurrences of the outcomes measured on the Swap test circuit. This value

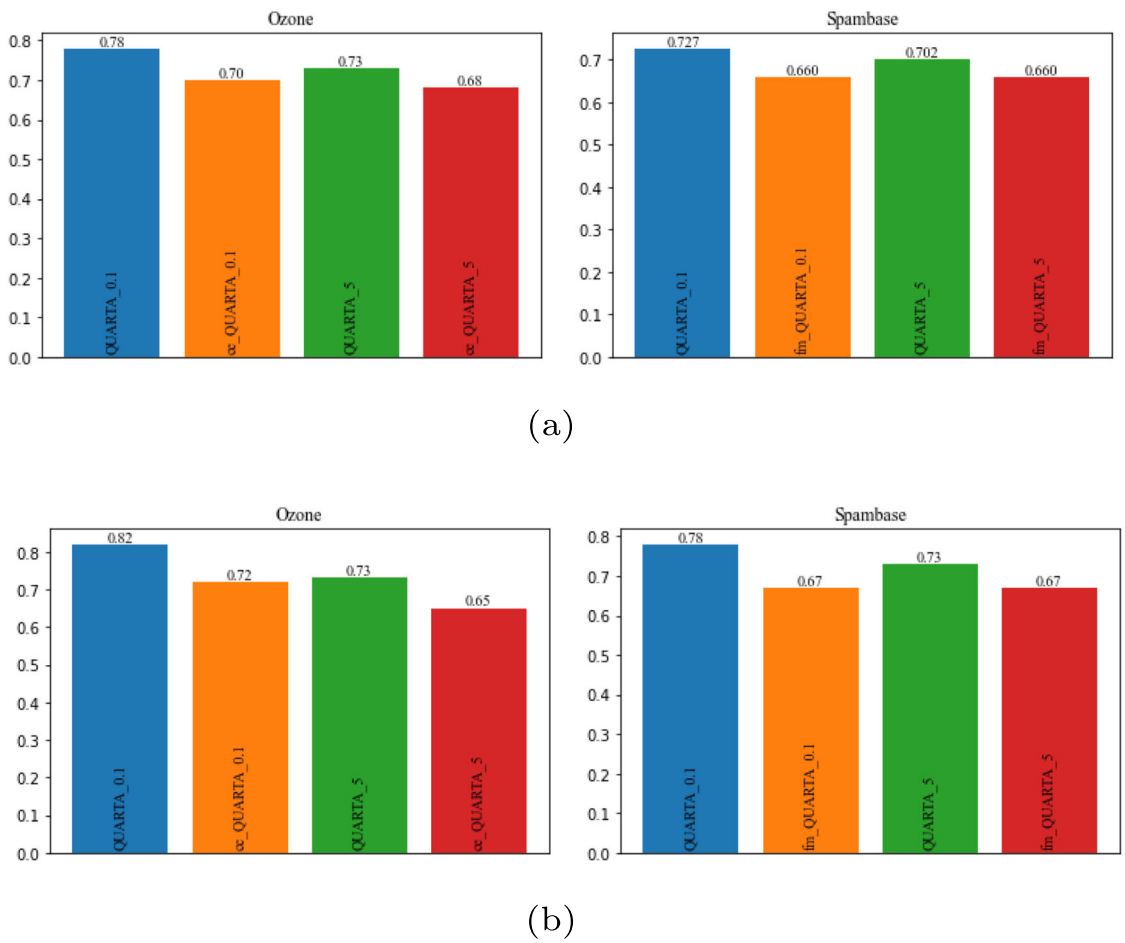


Fig. 2 Average F1-score values of QUARTA and variant ce_QUARTA computed on the prediction sessions with data block size equal to 250 and 400 (**a**), 25 and 50 (**b**). The threshold τ is 0.1 and 5 for both

follows the controlled number of runs of a quantum circuit, providing a more stable measure in the face of data variability.

Q3. Compare QUARTA against both the CC solution PerceptronMask and QC competitor QC_GEM

Similar to the variants of QUARTA, PerceptronMask is designed for binary classification and relies on a learning algorithm that optimizes weights (parameters). Specifically, when new data arrives, it provides an initial estimation by using the classifier so far built and, if the estimate is incorrect, it updates the models by tuning the weights only for the features relevant to the new data. QC_GEM has been built with the variational quantum classifier of the original design (Situ et al. 2023) and the Gradient Episodic Memory technique (GEM) for handling catastrophic forgetting (Lopez-Paz and Ranzato 2017). GEM keeps the classifier remembering previously learned tasks by limiting updates of the classifiers to minimize interference with past experiences. Specifically, before the gradient is used to update the parameters, it is modified to avoid the reduction of the performance on old tasks. When adapted to the domain-incremental scenario, it interprets a labeled data block as a new classification task. Both competitors can process both labeled and unlabeled data blocks but lack mechanisms to update the model in response to changes within these blocks, so their performances have been collected by prediction session. The remaining parameters were kept at their default values. The distribution of labeled and unlabeled data instances is obviously the same for all the algorithms. Figures 3 and 4 present the F1-score values across the flow of data instances, 2536 long for Ozone and 4600 long for Spambase. Each plot represents a distinct block size, namely 250, 150, 100, and 25 for Ozone (Fig. 3) and 400, 250, 100, and 50 for Spambase (Fig. 4). Upon analyzing the performance of QUARTA across the plots, it becomes evident that better results, in terms of *average F1-score*, are achieved when the data are processed by smaller pieces (data block size to 25 for Ozone (Fig. 3d) and 50 for Spambase (Fig. 4d). This suggests that QUARTA may have a greater ability to detect changes at a finer granularity and subsequently adapt the model accordingly. In these situations, the task of the optimizer is facilitated, as it is required to generalize over a smaller subset of data points rather than a larger set. This behavior does not always occur, but only when the classification model is frequently updated, which happens when considering several magnitudes of change, that is, when τ=0.1. Differently, at larger values of data block size (250 and 150 for Ozone; 400 and 250 for Spambase), QUARTA, configured with τ=0.1, takes on the role of the runner-up. In these cases, the CC algorithm PerceptronMask exhibits better performance, largely due to its utilization of all training data. Unlike PerceptronMask, QUARTA follows a "parsimonious" approach to data usage, since it exploits the training data for the model update only in the presence of data drifts; thus, when no drift occurs, the current data block of labeled data is deemed to have the same properties of the one processed previously and therefore it is not used for any alignment.

Another consideration pertains to the F1-score values across the flow of data instances. Even when QUARTA, at τ=0.1, does not exhibit better performance, it demonstrates

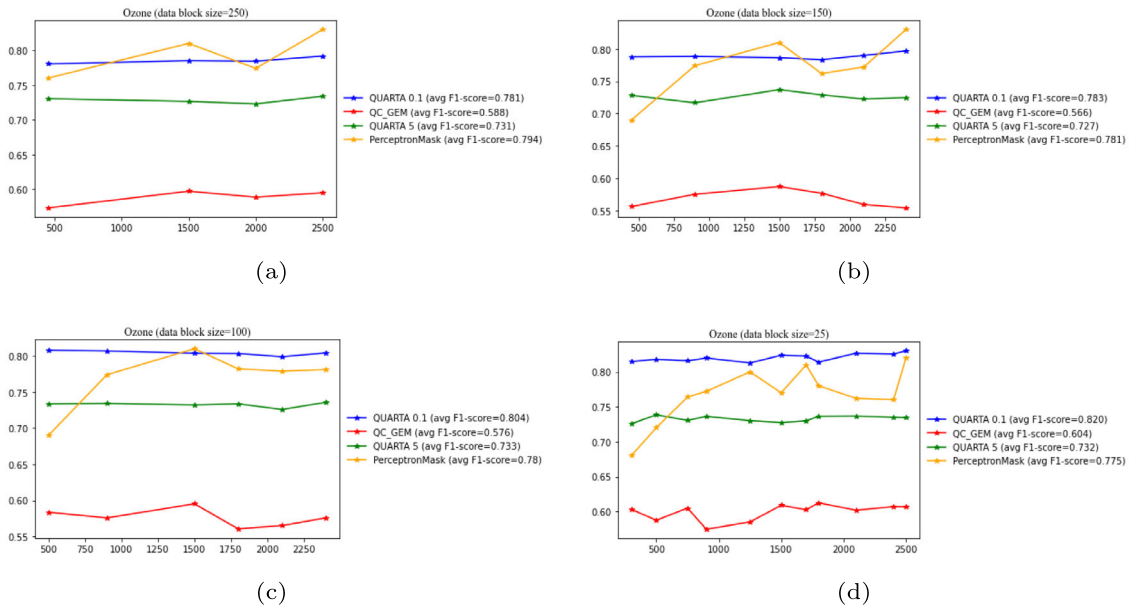


Fig. 3 F1-score values of QUARTA, PerceptronMask, and QC_GEM computed on the prediction sessions with data block size equal to 250, 150, 100, and 25 for Ozone. The threshold τ for QUARTA is 0.1 and 5

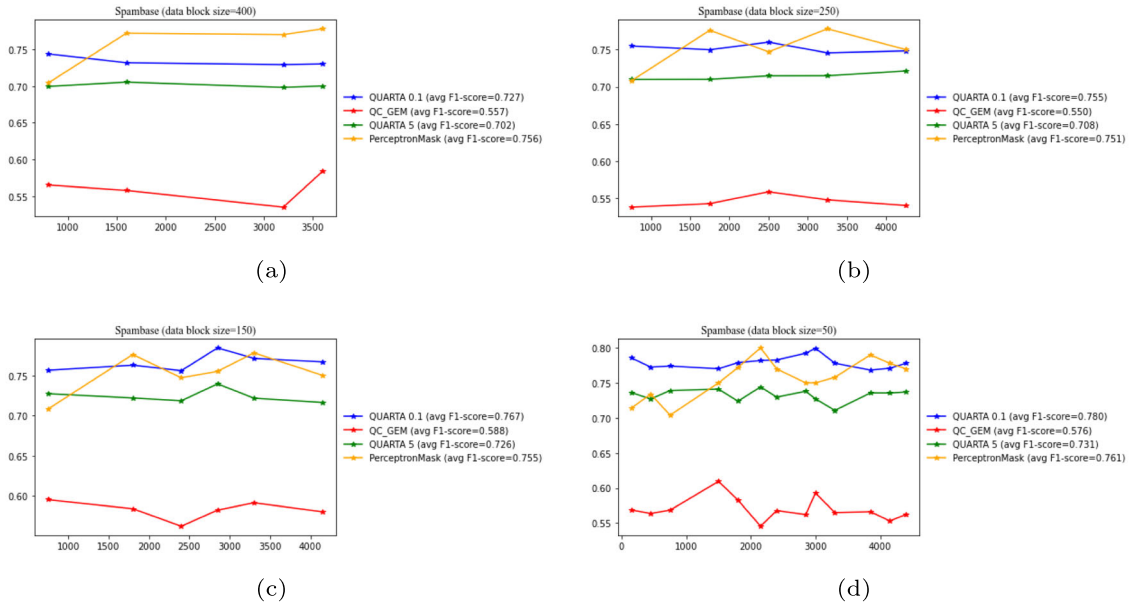


Fig. 4 F1-score values of QUARTA, PerceptronMask, and QC_GEM computed on the prediction sessions with data block size equal to 400, 250, 150, and 50 for Spambase. The threshold τ for QUARTA is 0.1 and 5

greater stability compared to PerceptronMask. PerceptronMask shows more fluctuations when operating on both larger and smaller data pieces.

Q4. Responsiveness of QUARTA to the data arrival rate

In incremental learning, data are typically processed as they arrive without particular stringent timing requirements to provide immediate responses. To maintain control over the rapid availability of data, time-window models are often employed. In this work, we introduce data blocks as a means to regulate the flow and employ change detection techniques to determine whether data exhibit significant changes warranting adaptation. However, the inclusion of time-consuming techniques, such as parameter optimization, and the need to determine an optimal data block size introduce complexities that must be addressed. Therefore, it is essential to investigate these aspects and provide experimental evidence to understand QUARTA capability to complete training sessions effectively, especially when dealing with large data block sizes. To address this, we conducted additional experimental sessions in which QUARTA was required to complete each training session within a specified time limit, during which the data block for the next training session would become available and ready for processing. The results reported in Tables 1 and 2 have been obtained by QUARTA working on Spambase, with τ=0.1 in the configurations of QUARTA^8 and QUARTA^{16} (8 qubits and 4 qubits respectively as input qubit register of the VQC). The value "Y" denotes the completion of QUARTA within the corresponding time limit value, the value "N" when exceeds

Table 1 Responsiveness of QUARTA for Spambase to process labeled data blocks within time limits. Binary classification with 8 qubits

Data block size	Time limit (s) 60	50	40	30	20	$\overline{AvgF1-score}$
100	Y	Y	Y	Y	Y	0.692
150	Y	Y	Y	Y	Y	0.692
250	Y	Y	Y	Y	N	0.677
400	Y	Y	N	N	N	0.622
450	Y	N	N	N	N	0.622
500	N	N	N	N	N	–

Table 2 Responsiveness of QUARTA for Spambase to process labeled data blocks within time limits

Data block size	Time limit (s) 60	50	40	30	20	$\overline{AvgF1-score}$
25	Y	Y	Y	Y	Y	0.776
100	Y	Y	Y	Y	Y	0.762
150	Y	Y	Y	Y	Y	0.758
250	Y	Y	Y	Y	Y	0.727
400	Y	Y	Y	Y	Y	0.721
450	Y	Y	Y	Y	Y	0.721
500	Y	Y	Y	Y	Y	0.721
600	Y	Y	Y	Y	Y	0.705
700	Y	Y	Y	Y	N	0.646
800	Y	Y	Y	Y	N	0.657
900	Y	Y	Y	Y	N	0.645

Binary classification with 4 qubits

that requirement. The column "$\overline{AvgF1-score}$" denotes the mean of the average F1-score metric values computed over the executions of QUARTA which terminate within the time limit. The results obtained shed light on the operational capabilities and limitations of QUARTA. When tasked with processing input qubit registers nearing 8 (QUARTA^{16}), the algorithm struggles to complete the task within a few tens of seconds, even when dealing with portions of data consisting of only a few hundred samples (see Table 1, data block size=400). This challenge likely arises from the significant expansion of the parameter space that the optimization process must explore. However, QUARTA^{16} also demonstrates the ability to achieve good performances, surpassing even QUARTA^{8} in certain scenarios (as seen in Table 1 vs Table 2, data block size=25). This is further confirmed by the results of the loss function analysis (see Fig. 1). We hypothesize that this improvement may be attributed to the incorporation of parameterized entanglements. Conversely, the results presented in Table 2 indicate that a QUARTA configuration utilizing half the number of qubits can successfully process data blocks containing almost one thousand of data instances within a few tens of seconds, while still maintaining satisfactory performance levels.

5 Related works

In the realm of quantum computing research, it is crucial to navigate the challenges and diverse outcomes, especially when revisiting problems previously tackled by classical computing. Our work underscores the significance of utilizing classical routines to build a hybrid quantum framework. Although quantum computing has produced both promising and mixed results, our approach aims to blend classical methodologies with quantum principles. Hybrid quantum-classical machine learning methods have recently gained considerable attention for their ability to combine the strengths of both classical and quantum computing, offering more efficient solutions to computationally complex problems. These methods aim to leverage quantum computing for specific tasks while relying on classical computing for others, such as data preprocessing or postprocessing of results. In a recent study by Houssein et al. (2022), a hybrid quantum-classical convolutional neural network model was proposed for prediction based on X-ray images. The model quantum component involves encoding, random quantum circuits, and decoding phases. This hybrid model demonstrated high accuracy, outperforming previous studies utilizing classical machine learning approaches in sensitivity and F1-measure. Similarly, in Yang et al. (2021), a hybrid quantum CNN model was introduced, employing a federated learning approach to protect models and prevent privacy failures attacks. Experimental results showed that models with additional quantum convolution exhibited slightly improved accuracy compared to baseline classical models. Furthermore, Chen et al. (2022) presented a hybrid quantum-classical model of long short-term memory (LSTM), a type of recurrent neural network (RNN). Their study compared the capability and performance of the hybrid model with its classical counterpart, revealing that the hybrid model converged faster and achieved better accuracy, albeit under conditions of noise and decoherence absence. Moreover, Herr et al. (2021) designed a hybrid quantum-classical approach for generative adversarial learning, aiming to detect anomalies and fraudulent transactions. Their model performance, measured by the F1 score, was comparable to its classical counterpart. Overall, recent research in hybrid quantum-classical machine learning methods underscores their potential to outperform classical machine learning algorithms on specific tasks, particularly those involving large or complex datasets. As quantum computing technology advances, further progress and the development of more powerful hybrid quantum-classical algorithms are anticipated.

[R1A2] There have been proposed hybrid approaches also for the binary classification, the same machine learning task as investigated in this manuscript. Arthur and Date (2022) implements a quantum-classical feed-forward neural network by using VQCs as individual neurons. In contrast to the one proposed here, the architecture has been tested also on real-world and synthetic data in their whole size and proved also on quantum devices. This indicates that they operate in a traditional batch learning setting while we work in continuous computation. A challenging data scenario has been investigated in Schetakis et al. (2022) where existing hybrid solutions have been tested on real quantum machines to learn binary classifiers in the unbalanced and noisy datasets, like those in the financial industry. Contrary, the datasets used in Sect. 4 are balanced and do not present data quality problems.

Incremental learning, often referred to as lifelong learning, continual learning, or sequential learning, constitutes a field of research dedicated to devising methodologies for acquiring knowledge from a perpetually evolving stream of data encompassing multiple tasks. The literature of classical computing on the general task of incremental learning is being a boost with the design of solutions based on neural models. Specific sub-areas, such as domain-incremental learning and continual learning, experience the same evolution, before attention deserved on vectorized-features data (Leo and Kalita 2024 Parisi et al. 2019) and then on complex data, like the images and texts (Wang et al. 2024). All these works are being tested on classical devices and are showing very good performances in terms of time consumption. This is an evaluation which we could not perform on the proposed framework as we considered only simulators. On the other

hand, at the present status of the quantum devices, the comparison in terms of running times with the CC algorithms would not be practicable.

In the field of quantum machine learning, research on continual learning remains sparse, relatively recent, and frequently in nascent stages. Situ et al. (2023) focuses on continual learning corresponding to a sequence of training sessions each involving a distinct classification tasks. Specifically, the authors center on a sequence of quantum state classification tasks by employing a variational quantum classifier whose parameters undergo optimization via a classical gradient-based optimizer. Drawing inspiration from category replay methods, akin to our approach in this study, the authors advocate for constraining the model updates by projecting the gradient direction onto the region delineated by previous task gradients. This is achieved by retaining a portion of the training data from previous tasks, termed as gradient episodic memory (GEM), upon which gradient descent computations are performed. However, a drawback of this approach is the need to compute gradients of previous tasks at each training iteration. In contrast to our approach, they investigate a multi-class problem (more than two labels) and concern the type of complex data of images. Also, the flow of the images is organized so that some classes can be processed in a data block and do not present anymore in those future. While QUARTA does not work on images but vectorized features and is designed for two class labels.

Still concerning the continual learning, Jiang et al. (2021) observed that as quantum classifiers encounter new classification tasks, their performance on previous tasks may deteriorate. To mitigate this phenomenon, the method of elastic weight consolidation (EWC) was introduced. EWC aims to safeguard the parameters crucial for previous tasks from undergoing significant updates. Numerical experiments have demonstrated that a quantum classifier can continuously learn and adapt to three distinct classification tasks without suffering from significant forgetting. However, it is worth noting that this work primarily deals with classical data (e.g., MNIST images), which are originally unaffected by data distribution drift and class label changes but are characterized by new class labels (tasks). Consequently, changes in the data characteristics of the same labels may only be partially addressed. **[R1A2]** The advantage of that method is that it does not need to store data instances already processed in previously training sessions, which is instead the design decision of the component data replay of QUARTA.

An alternative to EWC is represented by GEM that utilizes previously processed data stored in memory to modify the gradient of current data. Indeed, in Jain (2023), the authors prove that the performance of old data is more likely to improve as these previous data are revisited during parameter updates. The main difference with the current manuscript is that these data instances are selected through classical computing techniques, while we exploit the representation of Hilbert space and a quantum-based distance to search for data expressing changes. However, despite the growing interest in applying quantum to handle continuous processing, our results caution that, to provide a wide range of applications, we need to take the tradeoffs of accuracy, speed, and the cost of computing into consideration, as well as the kind of data the system works with.

6 Conclusions

As the production of actual quantum processors increases, quantum computing is gradually transcending its status as a purely theoretical field within computer science. The foundational principles governing QC are now firmly established, and initial applications underscore the practical relevance of quantum computational paradigms.

In light of these advancements, much of the current focus in QC revolves around pinpointing problems that offer tangible benefits from quantum computation over classical methods. This is witnessed not only by the attention on areas where quantum algorithms can achieve a significant speedup compared to the best-known classical algorithms, but also to a significant range of applications where quantum methods to generalize machine learning models efficiently from limited training data (Caro et al. 2022) and achieve superior prediction accuracies compared to the classical counterparts (Mishra et al. 2020). These advancements have ignited optimism regarding the prospect of attaining a tangible practical advantage in the form of quantum utility (Herrmann et al. 2023).

Incremental learning, or more generally lifelong learning, represents a category of machine learning tasks focused on the ability to learn, keep perpetually model capabilities, and eventually adapt them whether the learning context changes. This a study which does not demand exceptional computational properties of the quantum devices and that, theoretically, may work on NISQ devices. To our knowledge, there are very few works focused on this machine learning paradigm, whose main challenges include the unavailability of the data overall at one time and time-variability of the data properties.

In this paper, we studied this problem through a computational solution, called QUARTA, able to learn and adapt the capabilities of a predictive model represented by a quantum model and built by means of a quantum circuit. QUARTA enables the model to work in incremental learning, recognizing changing environments and reacting to drifting data scenarios while retaining previously assimilated knowledge.

The results demonstrate QUARTA ability to compete with and, in some conditions, surpass competing models, both quantum-based and classical-based. They also highlight the

importance of the number of qubits in the design of the quantum circuit, which can determine predictive performance improvements and the *utility* of the method. Additionally, the experiments underscore the significance of the number of data instances processed each time for QUARTA to realistically function in an incremental learning setting.

[R1A1] Experiments also provide arguments on the viability of QUARTA in different practical scenarios For instance, we explored the case study of a spam detection system for an email service, where QUARTA may support the system in classifying incoming emails as either "spam" or "not spam" based on a variety of features, such as the frequency of certain keywords, the presence of suspicious links, sender reputation. Unlike traditional machine learning, where the model is trained on a fixed dataset, emails arrive continuously, and the model must update in real-time with availability of access to few historical data. Incoming emails are accumulated in a container and time-regularly processed by blocks. Thus, the model is initialized with the first training data block. Over time, spammers develop new techniques to bypass traditional spam filters. These might involve using new types of words or new domains for sending spam emails. Therefore, as new spam emails with different characteristics emerge, the model is retrained incrementally, while including a representative subset of older spam data. This guarantees the model is continually adapted to these new spam strategies without forgetting how to detect older types of spam. Among the several quantum properties, the spam detection system can benefit from the use of the exponentially growing n-sized Hilbert space as the addressable feature space.

Future works include exploring machine learning paradigms such as few-shot learning to facilitate the construction of a parsimonious set of labeled data, thereby alleviating the encoding workload. Another direction involves designing quantum routines for change detection, potentially leveraging quantum parallelism to compute distances among multiple data instances simultaneously. Finally, considering more complex classical data may necessitate re-designing the feature aggregation component to encode, for instance, the flow of images.

Appendix 1. Quantum circuits employed in QUARTA

[R2A3] In this section, we provide a summary of the QC components implemented in the framework QUARTA. Basically, these components are involved in either quantum unsu-

Table 3 Quantum operators implemented in the QC components of QUARTA

QC component	Formulation	Explanation
Data encoding	$\mathbf{U}_\Phi(\mathbf{X}) = \left(R_z(\psi^m)R_y(\psi^m)\right)\ldots\left(R_z(\psi^j)R_y(\psi^j)\right)\ldots\left(R_z(\psi^1)R_y(\psi^1)\right)H\|0\rangle$	A dense data encoding technique to build quantum states from classical data described by the m first selected features finalized to the component Distance estimation & Prototype generation
Feature aggregation	$\mathbf{U}_{\Delta(x_i,x_{i+1})} = (I^i \otimes {R_y}^{i+1}(\lambda))\ CX^{i,i+1}\ (R_z^i(\omega) \otimes R_y^{i+1}(\phi))\ CX^{i+1,i}\ (I^i \otimes {R_z}^{i+1}(-\frac{\pi}{2}))\|0^2\rangle$	Construction of new features by aggregation of pairs of features. The operator compresses the pair of features on two qubits into one qubit, so, the information of a pair of quantum states is conveyed on one of the two qubits
Distance estimation & Prototype generation	$(H^1 \otimes I^2 \otimes I^3)CSWAP^{\{1\}\{2,3\}}(H^1 \otimes I^2 \otimes I^3)$	Quantum fidelity is nested in the calculation of the distance between quantum states encoding classical data when deciding whether data blocks exhibit changes
Binary classification	$\mathbf{A}(\Omega,\lambda) = ({R_y}^i(\Omega_0) \otimes {R_y}^{i+1}(\Omega_1) \otimes {R_y}^{i+2}(\Omega_2) \otimes {R_y}^{i+3}(\Omega_3))(CX^{i+3,i}) \otimes CX^{i+2,i+3} \otimes CX^{i+1,i+2} \otimes CX^{i,i+1})({R_y}^i(\Omega_4) \otimes {R_y}^{i+1}(\Omega_5) \otimes {R_y}^{i+2}(\Omega_6) \otimes {R_y}^{i+3}(\Omega_7))(CX^{i+3,i+2} \otimes CX^{i,i+3} \otimes CX^{i+1,i} \otimes CX^{i+2,i})\|0^{\otimes 4}\rangle\lambda$	Template circuit with four-sized qubits and eight parameters considered to design the VQC finalized to learn the classification model. It is organized by two main blocks with single qubit and parameterized two qubit gates. The whole VQC has a number of qubit determined by Feature aggregation
Binary classification	$\mathbf{A}(\Omega,\lambda) = ({R_y}^i(\Omega_0) \otimes {R_y}^{i+1}(\Omega_1) \otimes {R_y}^{i+2}(\Omega_2) \otimes {R_y}^{i+3}(\Omega_3))(CRX^{i+3,i}(\Omega_4) \otimes CRX^{i+2,i+3}(\Omega_5) \otimes CRX^{i+1,i+2}(\Omega_6) \otimes CRX^{i,i+1}(\Omega_7))({R_y}^i(\Omega_8) \otimes {R_y}^{i+1}(\Omega_8) \otimes {R_y}^{i+2}(\Omega_{10}) \otimes {R_y}^{i+3}(\Omega_{11}))(CRX^{i+3,i+2}(\Omega_{12}) \otimes CRX^{i,i+3}(\Omega_{13}) \otimes CRX^{i+1,i}(\Omega_{14}) \otimes CRX^{i+2,i+1}(\Omega_{15}))\|0^{\otimes 4}\rangle\lambda$	Template circuit with four-sized qubits and sixteen parameters considered to design the VQC finalized to learn the classification model. It is organized by two main blocks with single qubit and parameterized two qubit gates. The whole VQC has a number of qubit determined by Feature aggregation

pervised learning process or quantum supervised learning process. We report the formulation of each quantum operator and explain the purpose in Table 3.

Appendix 2. Results on the effectiveness of feature aggregation

[R2A1]. In addition to the experiments above presented, we evaluated another variant of QUARTA, referred to as fm_QUARTA, designed by replacing feature aggregation with a basic feature mapping technique. This technique encodes each qubit using the operator $R_y(\lambda)H|0\rangle$, where λ represents the feature value. The purpose of this variant is to assess the impact of a quantum-based feature compression technique.

QUARTA and fm_QUARTA were executed with two different data block sizes per dataset, specifically 250 and 25 for Ozone and 400 and 50 for Spambase. The change threshold (τ) was set at the lowest value of 0.1 and highest value of 5, one aiming at facilitating model updates, the other one preventing frequent updates. Feature aggregation compresses 8 features into 4. The results of the average F1-score are reported in Fig. 5. The plots demonstrate the effectiveness of the proposed solution for pairwise feature encoding, particularly in scenarios where the threshold $\tau = 0.1$ (0.78 and 0.82 Ozone, 0.727 and 0.78 Spambase): low values of τ allow to capture also mild changes, resulting in frequent updates of the model. This happen regardless of whether the data blocks are large or small. Moreover, the discrepancy between QUARTA and fm_QUARTA is a bit larger when $\tau = 0.1$ (0.78 vs 0.73 and 0.82 vs 0.75 for Ozone, 0.727 vs 0.700 and 0.78 vs 0.73 for Spambase) than when $\tau = 5$, meaning that fm_QUARTA is less sensitive to mild changes. Both approaches initially utilize identical feature sets, but the potential advantage of fm_QUARTA, that is, a broader representational space for its VQC, transforms into a disadvantage because the optimizer must minimize the loss function while dealing with a decoupled number of variational parameters, unlike QUARTA. Notably, pairing quantum states offers sev-

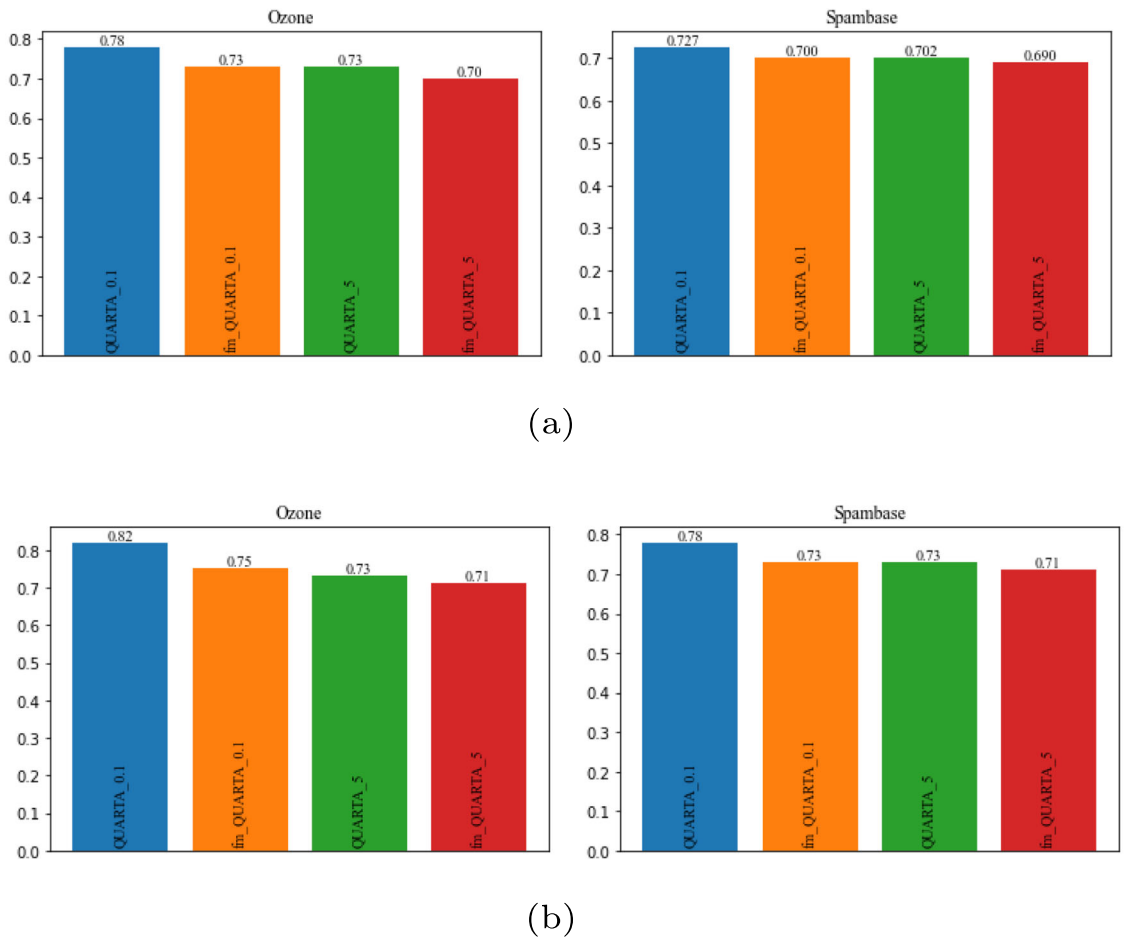


Fig. 5 Average F1-score values of QUARTA and variant fm_QUARTA computed on the prediction sessions with data block size equal to 250 and 400 (**a**), 25 and 50 (**b**). The threshold τ is 0.1 and 5 for both

eral advantages: (i) it allows for conveying more information compared to a "flat" feature mapping approach that considers fewer features, while using the same number of resulting qubits; (ii) it enables the synthesis of quantum states while maintaining a high level of discriminative power among the data; (iii) it exploits better the Hilbert space to build encoding able to discern features with both high variability from those with marginal drifts. This result confirms the adequacy of the operator 3 in halving the dimensionality and retaining of important information, in addition to the one of supporting convolutions in quantum neural networks Hur et al. (2022). Clearly, this ability tends to diminish when τ values are high, as the change detector reveals less changes and the data blocks contain high variability data, requiring QUARTA to encode and join features with several degrees of drifting.

Acknowledgements Corrado Loglisci and Saverio Pascazio acknowledge the support by PNRR MUR project PE0000023-NQSTI. Donato Malerba is partially supported by InnoAgroECoS - Innovazioni tecnologiche e organizzative per la transizione agroecologica dei sistemi agroalimentari locali verso un modello di economia circolare e solidale", Horizon Europe Seeds 2021, CUP H91I21001630006.

Author Contributions C.L. contributed to methodology, software, validation, and writing. D.M. contributed to methodology, writing, and supervision. S.P. contributed to methodology, writing, supervision, and funding acquisition

Funding Open access funding provided by Università degli Studi di Bari Aldo Moro within the CRUI-CARE Agreement.

Data Availability The experimental datasets are available at the links https://archive.ics.uci.edu/ml/datasets/ozone+level+detection and https://archive.ics.uci.edu/ml/datasets/Spambase.

Declarations

Conflict of interest The authors declare no Conflict of interest.

References

Preskill, J (2018) Quantum computing in the NISQ era and beyond. Quantum. 2:79 https://doi.org/10.22331/q-2018-08-06-79

Callison, A., Chancellor, N (2022) Hybrid quantum-classical algorithms in the noisy intermediate-scale quantum era and beyond. Phys Rev A 106:010101 https://doi.org/10.1103/PhysRevA.106.010101

Bova Francesco, Goldfarb Avi, Melko Roger G (2021) Commercial applications of quantum computing. EPJ Quantum Technol. 8(1):2. https://doi.org/10.1140/epjqt/s40507-021-00091-1

Gunasekara N, Pfahringer B, Gomes HM, Bifet A (2023) Survey on online streaming continual learning. In: IJCAI, pp 6628–6637

Halstead B, Koh YS, Riddle P, Pears R, Pechenizkiy M, Bifet A, Olivares G, Coulson G (2022) Analyzing and repairing concept drift adaptation in data stream classification. Mach Learn 111(10):3489–3523

Mai Z, Li R, Jeong J, Quispe D, Kim H, Sanner S (2022) Online continual learning in image classification: an empirical survey. Neurocomputing 469:28–51

Gunasekara N, Gomes HM, Bifet A, Pfahringer B (2022) Adaptive neural networks for online domain incremental continual learning. In: Poncelet P, Ienco D (eds.) Discovery Science - 25th International Conference, DS 2022, Montpellier, France, October 10-12, 2022, Proceedings. Lecture Notes in Computer Science, vol 13601, pp 89–103 . https://doi.org/10.1007/978-3-031-18840-4_7

Peng J, Tang B, Jiang H, Li Z, Lei Y, Lin T, Li H (2022) Overcoming long-term catastrophic forgetting through adversarial neural pruning and synaptic consolidation. IEEE Trans Neural Networks Learn Syst 33(9):4243–4256. https://doi.org/10.1109/TNNLS.2021.3056201

Jiang W, Lu Z, Deng D-L (2022) Quantum continual learning overcoming catastrophic forgetting. Chin Phys Lett 39(5). https://doi.org/10.1088/0256-307x/39/5/050303

Dunjko V, Briegel HJ (2018) Machine learning and artificial intelligence in the quantum domain: a review of recent progress. Rep Prog Phys 81(7). https://doi.org/10.1088/1361-6633/aab406

Hur T, Kim L, Park DK (2022) Quantum convolutional neural network for classical data classification. Quantum Mach Intell 4(1) https://doi.org/10.1007/s42484-021-00061-x

Benedetti M, Lloyd E, Sack SH, Fiorentini M (2019) Erratum: Parameterized quantum circuits as machine learning models (2019 quant sci tech 4 043001). Quantum Science and Technology. 5

Sharma D, Singh P, Kumar A (2022) The role of entanglement for enhancing the efficiency of quantum kernels towards classification. Statistical Mechanics and its Applications, Physica A

Liu B (2017) Lifelong machine learning: a paradigm for continuous learning. Frontiers Comput Sci 11(3):359–361

Herrmann N, Arya D, Doherty MW, Mingare A, Pillay JC, Preis F, Prestel S (2023) Quantum utility - definition and assessment of a practical quantum advantage. In: Ali S, Ardagna CA, Atukorala NL, Barzen J, Chang CK, Chang RN, Fan J, Faro I, Feld S, Fox G, Jin Z, Leymann F, Neukart F, Puente S, Wimmer M (eds.) IEEE International Conference on Quantum Software, QSW 2023, Chicago, IL, USA, 2-8 July 2023, pp 162–174 . https://doi.org/10.1109/QSW59989.2023.00028

Zhao J, Xie X, Xu X, Sun S (2017) Multi-view learning overview: recent progress and new challenges. Inf Fusion 38:43–54

Sierra-Sosa D, Pal S, Telahun M (2023) Data rotation and its influence on quantum encoding. Quantum Inf Process 22(1):89

Gama J, Gaber MM (2007) Learning from data streams: processing techniques in sensor networks, 1st edn

LaRose R, Coyle B (2020) Robust data encodings for quantum classifiers. Phys Rev A 102:032420 https://doi.org/10.1103/PhysRevA.102.032420

Lange MD, Aljundi R, Masana M, Parisot S, Jia X, Leonardis A, Slabaugh GG, Tuytelaars T (2022) A continual learning survey: defying forgetting in classification tasks. IEEE Trans Pattern Anal Mach Intell 44(7):3366–3385. https://doi.org/10.1109/TPAMI.2021.3057446

Chandrashekar G, Sahin F (2014) A survey on feature selection methods. Comput Electr Eng 40(1):16–28. https://doi.org/10.1016/j.compeleceng.2013.11.024

Ross BC (2014) Mutual information between discrete and continuous data sets. PLoS ONE 9(2):87357. https://doi.org/10.1371/journal.pone.0087357

Lloyd S, Schuld M, Ijaz A, Izaac J, Killoran N (2020) Quantum embeddings for machine learning. arXiv:2001.03622

Schuld M, Killoran N (2019) Quantum machine learning in feature Hilbert spaces. Phys Rev Lett 122(4)

Pérez-Salinas A, Cervera-Lierta A, Gil-Fuster E, Latorre JI (2020) Data re-uploading for a universal quantum classifier. Quantum. 4:226 https://doi.org/10.22331/q-2020-02-06-226

Nielsen MA, Chuang IL (2011) Quantum computation and quantum information: 10th anniversary edition, 10th edn. Cambridge University Press, USA

Guo S, Zhao H, Yang W (2021) Hierarchical feature selection with multi-granularity clustering structure. Inf Sci 568:448–462 https://doi.org/10.1016/j.ins.2021.04.046

Yu K, Lin S, Guo G-D (2023) Quantum dimensionality reduction by linear discriminant analysis. Physica A: Statis Mechanic Appl 614:128554 https://doi.org/10.1016/j.physa.2023.128554

Lloyd S, Mohseni M, Rebentrost P (2014) Quantum principal component analysis. Nat Phys 10(9):631–633. https://doi.org/10.1038/nphys3029

Galanis N-I, Vafiadis P, Mirzaev K-G, Papakostas GA (2022) Convolutional neural networks: a roundup and benchmark of their pooling layer variants. Algorithms. 15(11) https://doi.org/10.3390/a15110391

Vatan F, Williams C (2004) Optimal quantum circuits for general two-qubit gates. Phys Rev A 69(3)

Madden L, Simonetto A (2022) Best approximate quantum compiling problems. ACM Transactions on Quantum Computing. 3(2) https://doi.org/10.1145/3505181

Kendall MG (1938) A new measure of rank correlation. Biometrika. 30(1/2):81–93. Accessed 2024-03-14

Nielsen MA Chuang IL (2010) Distance measures for quantum information. In: Quantum Computation and Quantum Information, 2nd edn. Cambridge University Press, Cambridge

Jozsa R (1994) Fidelity for mixed quantum states. Journal of Modern Optics. 41(12):2315–2323 https://doi.org/10.1080/09500349414552171

Havlícek V, Córcoles AD, Temme K, Harrow AW, Kandala A, Chow JM, Gambetta JM (2019) Supervised learning with quantum-enhanced feature spaces. Nat 567(7747):209–212. https://doi.org/10.1038/s41586-019-0980-2

Barenco A, Berthiaume A, Deutsch D, Ekert A, Jozsa R, Macchiavello C (1997) Stabilization of quantum computations by symmetrization. SIAM Journal on Computing. 26(5):1541–1557 https://doi.org/10.1137/S0097539796302452

Patel RB, Ho J, Ferreyrol F, Ralph TC, Pryde GJ (2016) A quantum Fredkin gate. Science. Advances 2(3):1501531. https://doi.org/10.1126/sciadv.1501531

Kavitha SS, Kaulgud N (2023) Quantum k-means clustering method for detecting heart disease using quantum circuit approach. Soft Comput 27(18):13255–13268

Acampora G, Di Martino F, Schiattarella R, Vitiello A (2021) Measuring distance between quantum states by fuzzy similarity operators. In: 2021 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), pp 1–6. https://doi.org/10.1109/FUZZ45933.2021.9494403

Aïmeur E, Brassard G, Gambs S (2013) Quantum speed-up for unsupervised learning. Mach Learn 90(2):261–287

Benlamine K, Bennani Y, Grozavu N, Matei B (2020) Quantum collaborative k-means. In: 2020 International Joint Conference on Neural Networks, IJCNN 2020, Glasgow, United Kingdom, 19-24 July 2020, pp 1–7 . https://doi.org/10.1109/IJCNN48605.2020.9207334

Kerenidis I, Landman J, Luongo A, Prakash A (2019) q-means: a quantum algorithm for unsupervised machine learning. In: Wallach HM, Larochelle H, Beygelzimer A, d'Alché-Buc F, Fox EB, Garnett R (eds.) Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada, pp 4136–4146

Page ES (1954) Continuous inspection schemes. Biometrika 41(1/2):100–115 . Accessed 2023-06-04

Benedetti M, Lloyd E, Sack S, Fiorentini M (2019) Parameterized quantum circuits as machine learning models. Quantum Sci Technol 4(4). https://doi.org/10.1088/2058-9565/ab4eb5

Arthur D, Date P (2022) Hybrid quantum-classical neural networks. In: IEEE International Conference on Quantum Computing and Engineering, QCE 2022, Broomfield, CO, USA, 18-23 September 2022, pp 49–55. https://doi.org/10.1109/QCE53715.2022.00023

Sim S, Johnson PD, Aspuru-Guzik A (2019) Expressibility and entangling capability of parameterized quantum circuits for hybrid quantum-classical algorithms. Adv Quantum Technol 2(12):1900070

Hubregtsen T, Pichlmeier J, Stecher P, Bertels K (2021) Evaluation of parameterized quantum circuits: on the relation between classification accuracy, expressibility, and entangling capability. Quantum Mach Intell 3(1):1–19

Schuld M, Bocharov A, Svore KM, Wiebe N (2020) Circuit-centric quantum classifiers. Phys Rev A 101(3) https://doi.org/10.1103/physreva.101.032308

Wierichs D, Izaac J, Wang C, Lin CY-Y (2022) General parameter-shift rules for quantum gradients. Quantum 6:677 https://doi.org/10.22331/q-2022-03-30-677

Anis MS, al (2021) Qiskit: an open-source framework for quantum computing. https://doi.org/10.5281/zenodo.2573505

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

Montiel J, Read J, Bifet A, Abdessalem T (2018) Scikit-multiflow: a multi-output streaming framework. J Mach Learn Res 19(72):1–5

Situ H, Lu T, Pan M, Li L (2022) Quantum continual learning of quantum data realizing knowledge backward transfer. arXiv:2203.14032 [quant-ph]

Gama J, Sebastião R, Rodrigues PP (2013) On evaluating stream learning algorithms. Mach Learn 90(3):317–346

Situ H, Lu T, Pan M, Li L (2023) Quantum continual learning of quantum data realizing knowledge backward transfer. Physica A: Statistic Mechanic Appl 620:128779. https://doi.org/10.1016/j.physa.2023.128779

Lopez-Paz D, Ranzato M (2017) Gradient episodic memory for continual learning. Adv Neural Inf Process Syst 30

Houssein EH, Abohashima Z, Elhoseny M, Mohamed WM (2022) Hybrid quantum-classical convolutional neural network model for COVID-19 prediction using chest X-ray images. J Comput Design and Eng 9(2):343–363. https://doi.org/10.1093/jcde/qwac003

Yang C-HH, Qi J, Chen SY-C, Chen P-Y, Siniscalchi SM, Ma X, Lee C-H (2021) Decentralizing feature extraction with quantum convolutional neural network for automatic speech recognition. In: ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp 6523–6527 . https://doi.org/10.1109/ICASSP39728.2021.9413453

Chen SY, Yoo S, Fang YL (2022) Quantum long short-term memory. In: IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2022, Virtual and Singapore, 23-27 May

2022, pp 8622–8626 . https://doi.org/10.1109/ICASSP43922.2022.9747369

Herr D, Obert B, Rosenkranz M (2021) Anomaly detection with variational quantum generative adversarial networks. Quantum Sci Technol 6(4). https://doi.org/10.1088/2058-9565/ac0d4d

Schetakis N, Aghamalyan D, Griffin P, Boguslavsky M (2022) Review of some existing QML frameworks and novel hybrid classical–quantum neural networks realising binary classification for the noisy datasets. Sci Rep 12(1):11927. https://doi.org/10.1038/s41598-022-14876-6

Leo J, Kalita J (2024) Survey of continuous deep learning methods and techniques used for incremental learning. Neurocomputing. 582:127545 https://doi.org/10.1016/j.neucom.2024.127545

Parisi GI, Kemker R, Part JL, Kanan C, Wermter S (2019) Continual lifelong learning with neural networks: A review. Neural Netw 113:54–71. https://doi.org/10.1016/j.neunet.2019.01.012

Wang K, Zhang G, Yue H, Liu A, Zhang G, Feng H, Han J, Ding E, Wang J (2024) Multi-domain incremental learning for face presentation attack detection. Proceed AAAI Conf Art Intell 38(6):5499–5507. https://doi.org/10.1609/aaai.v38i6.28359

Jiang W, Lu Z, Deng D (2021) Quantum continual learning overcoming catastrophic forgetting. CoRR. abs/2108.02786 arXiv:2108.02786

Jain S (2023) Cqural: A novel CNN based hybrid architecture for quantum continual machine learning. CoRR. abs/2305.09738 https://doi.org/10.48550/arXiv.2305.097382305.09738

Caro MC, Huang H-Y, Cerezo M, Sharma K, Sornborger A, Cincio L, Coles PJ (2022) Generalization in quantum machine learning from few training data. Nat Commu 13(1) . https://doi.org/10.1038/s41467-022-32550-3

Mishra N, Kapil M, Rakesh H, Anand A, Mishra N, Warke A, Sarkar S, Dutta S, Gupta S, Dash AP, Gharat RM, Chatterjee Y, Roy S, Raj S, Jain VK, Bagaria S, Chaudhary S, Singh V, Maji R, Dalei P, Behera BK, Mukhopadhyay S, Panigrahi PK (2020) Quantum machine learning: a review and current status. Data Management, Analytics and Innovation