A Single-Step Multiclass SVM Based on Quantum Annealing for Remote Sensing Data Classification

Amer Delilbasic ⓘ, *Student Member, IEEE*, Bertrand Le Saux ⓘ, *Senior Member, IEEE*, Morris Riedel ⓘ, *Member, IEEE*, Kristel Michielsen ⓘ, and Gabriele Cavallaro ⓘ, *Senior Member, IEEE*

Abstract—In recent years, the development of quantum annealers has enabled experimental demonstrations and has increased research interest in applications of quantum annealing, such as in quantum machine learning and in particular for the popular quantum support vector machine (SVM). Several versions of the quantum SVM have been proposed, and quantum annealing has been shown to be effective in them. Extensions to multiclass problems have also been made, which consist of an ensemble of multiple binary classifiers. This article proposes a novel quantum SVM formulation for direct multiclass classification based on quantum annealing, called quantum multiclass SVM (QMSVM). The multiclass classification problem is formulated as a single quadratic unconstrained binary optimization problem solved with quantum annealing. The main objective of this article is to evaluate the feasibility, accuracy, and time performance of this approach. Experiments have been performed on the D-Wave Advantage quantum annealer for a classification problem on remote sensing data. Results indicate that, despite the memory demands of the quantum annealer, QMSVM can achieve an accuracy that is comparable to standard SVM methods, such as the one-versus-one (OVO), depending on the dataset (compared to OVO: 0.8663 versus 0.8598 on Toulouse, 0.8123 versus 0.8521 on Potsdam). More importantly, it scales much more efficiently with the number of training examples, resulting in nearly constant time (compared to OVO: 85.72 versus 248.02 s on Toulouse, 58.89 versus 580.17 s on Potsdam). This article shows an approach for bringing together classical and quantum computation, solving practical problems in remote sensing with current hardware.

Index Terms—Classification, quantum annealing (QA), quantum computing (QC), remote sensing (RS), support vector machine (SVM).

Nomenclature

AQC	Adiabatic quantum computation.
BDSD	Band-dependent spatial detail.
CS	Crammer–Singer.
DAG	Directed acyclic graph.
DSM	Digital surface model.
EO	Earth observation.
ML	Machine learning.
OVA	One-versus-all.
OVO	One-versus-one.
QA	Quantum annealing.
QC	Quantum computing.
QML	Quantum machine learning.
QMSVM	Quantum multiclass support vector machine.
QSVM	Quantum support vector machine.
QUBO	Quadratic unconstrained binary optimization.
RS	Remote sensing.
SVM	Support vector machine.

Amer Delilbasic is with the Jülich Supercomputing Centre, Forschungszentrum Jülich, 52428 Jülich, Germany, also with the University of Iceland, 107 Reykjavik, Iceland, and also with the Φ-Lab, European Space Research Institute, European Space Agency, IT-00044 Frascati, Italy (e-mail: a.delilbasic@fz-juelich.de).

Bertrand Le Saux is with the Φ-Lab, European Space Research Institute, European Space Agency, IT-00044 Frascati, Italy (e-mail: bertrand.le.saux@esa.int).

Morris Riedel and Gabriele Cavallaro are with the University of Iceland, 107 Reykjavik, Iceland, also with the Jülich Supercomputing Centre, Forschungszentrum Jülich, 52428 Jülich, Germany, and also with the AI, Data Analytics and Scalable Simulation (AIDAS), 52425 Jülich, Germany (e-mail: morris@hi.is; g.cavallaro@fz-juelich.de).

Kristel Michielsen is with the Jülich Supercomputing Centre, Forschungszentrum Jülich, 52428 Jülich, Germany, also with the RWTH Aachen University, D-52056 Aachen, Germany, and also with the AIDAS, 52425 Jülich, Germany (e-mail: k.michielsen@fz-juelich.de).

Digital Object Identifier 10.1109/JSTARS.2023.3336926

I. Introduction

IN THE context of EO [1], there is a growing availability of data acquired by heterogeneous RS sources . Many applications are currently benefitting from RS data, e.g., agriculture, green energy development, and urban monitoring. The devices and software for data processing have to match this trend in order to extract information from the collected data in a timely manner.

QC [2], a computational paradigm based on the postulates and laws of quantum mechanics, has proved the potential to reach an exponential algorithmic speedup with respect to classical computation under certain assumptions [3], [4]. Among the quantum computational models defined in the literature, two broadly employed models can be identified. The quantum circuit model [5], similarly to the classical circuit model, is based on circuits, gates, and measurements applied to qubits (quantum bits). AQC [6], [7] aims at solving optimization problems by exploiting the time evolution of a quantum system satisfying the requirements of the adiabatic theorem [8]. Despite their differences, the two models

have been proven to be computationally equivalent [9]. The focus of this work is QA [10], [11], a heuristic search approach based on AQC, since commercially ready quantum annealers are available for analyzing the disruptive potential of QC.

QML [12], [13] is a research area working on QC algorithms applied to ML tasks, with the purpose of obtaining a computational speedup or a prediction accuracy increase. QML methods based on QA have proven to outperform classical ML in selected applications with limited training examples, for example, in computational biology [14]. Recent studies have analyzed how QML can be integrated into EO tasks. In [15], [16], [17], and [18], circuit-based quantum neural networks have been trained for multispectral image classification. The work of Otgonbaatar and Datcu has covered different aspects of circuit-based QML for EO, e.g., natural data embedding [19], parameterized quantum gates [20], and transfer learning [21]. Circuit-based quantum kernels have been applied to binary [22], [23], [24] and multiclass [25] RS image classification. QA has also found a place in EO for solving specific optimization problems. In synthetic aperture radar imaging, problems related to system design [26] and phase ambiguity [27] have been addressed. In the context of QML, a feature selection method for hyperspectral images has been proposed [28], and a QA-based QSVM method has been successfully used for the binary classification of multispectral images [29], [30].

The SVM is an efficient and theoretically grounded algorithmic approach in statistical learning theory. Different versions and formulations of the SVM can be found in the literature for a variety of tasks and applications, e.g., pattern recognition, computer vision, image analysis, and business intelligence [31]. The SVM has also been proven to be effective in EO pixelwise image classification [32].

Defining an SVM framework for multiclass classification is a nontrivial task. Two different approaches can be followed [33]. The *multiple-step* (or *indirect*) *approach* reframes the problem by defining an ensemble of multiple binary SVM classifiers and multiple classification outputs. The most common methods are the OVO, OVA, and the DAG SVM. In the OVO method, each pair of classes defines an SVM classifier. The outcomes of each classifier are usually combined with a "max wins" strategy, which determines the final prediction. Similarly, in the OVA method, a classifier for each class versus the remaining classes is defined, and the class with highest score is assigned. The QSVM algorithms for multiclass classification available in the literature, e.g., [34], [35], [36], follow the multistep approach. They are defined as ensembles of binary QSVM classifiers, which can be full quantum [37], [38], quantum-kernel-based [39] and QA-based [40] formulations.

The *single-step* (or *direct*) *approach* for multiclass classification defines a single optimization step on the whole training set, which finds boundaries between all classes in one pass. The CS SVM, proposed in [41], is an example of single-step approach. It showed a comparable to better performance on benchmark datasets with respect to OVO, OVA, and the Weston–Watkins [42] multiclass SVM [43]. However, a limitation of this method is the complexity of the training phase, due to the high number of optimization variables, which makes this

approach impractical. In the same work, simplified formulations are proposed, which reduce the problem size and enable better performances, although at the cost of optimality.

The main objective of this article is to propose a novel approach, specifically QMSVM, by reframing the original formulation of the CS SVM, thus enabling the optimization step to be performed using a QA algorithm. This article studies the computational capability offered by the available quantum annealers and assesses the feasibility of a QA-based single-step SVM approach. Experiments are performed on a real quantum annealer, i.e., D-Wave Advantage [44], [45], in order to validate the algorithm on current hardware. As quantum annealers are a rapidly evolving technology, it is important to analyze their current status and their potential, understanding how they can be used to solve real problems in RS. For this purpose, it is reasonable to consider the CS SVM implementation in our work, as it is a computationally intensive task that would benefit from an enhanced performance. The performance is evaluated both in terms of accuracy and execution time, both relevant aspects in practice. The code repository of the algorithm is made available for reproducibility.[1]

The rest of this article is structured as follows. In Section II, the theoretical background related to QA is summarized. In Section III, the mathematical formulation of QMSVM is presented. In Section IV, the algorithm validation setup is described and the results are shown. In Section V, the main findings are discussed. Finally, Section VI concludes this article.

II. BACKGROUND

A. Quantum Annealing and QUBO

To understand the underlying working principles of D-Wave quantum annealers, a brief introduction is needed. In AQC [6], [7], the forces acting in a quantum system are described by a time-varying Hamiltonian $\mathcal{H}(t)$. The time evolution of the state of a quantum system $|\varphi(t)\rangle$ is described by the Schrödinger's equation

$$i\hbar \frac{\partial |\varphi(t)\rangle}{\partial t} = \mathcal{H}(t)|\varphi(t)\rangle \tag{1}$$

where i is the imaginary unit and $\hbar$ is the reduced Planck constant. During the adiabatic evolution, the Hamiltonian gradually transitions from the initial Hamiltonian $\mathcal{H}_I$ to the final Hamiltonian $\mathcal{H}_F$

$$\mathcal{H}(t) = s(t)\mathcal{H}_I + (1 - s(t))\mathcal{H}_F \tag{2}$$

where $s(t)$ is a function modeling the transition, such that $s(0) = 1$ and $s(t_f) = 0$ after a certain elapsed time t_f. Given the assumptions of the adiabatic theorem [8], during the time evolution, the quantum system remains at ground state, i.e., the state with lowest energy associated with the Hamiltonian. The idea in AQC is to encode the desired result as the ground state of the final Hamiltonian $\mathcal{H}_F$.

QA falls into the category of AQC algorithms. More precisely, it is a heuristic approach for solving combinatorial optimization

---

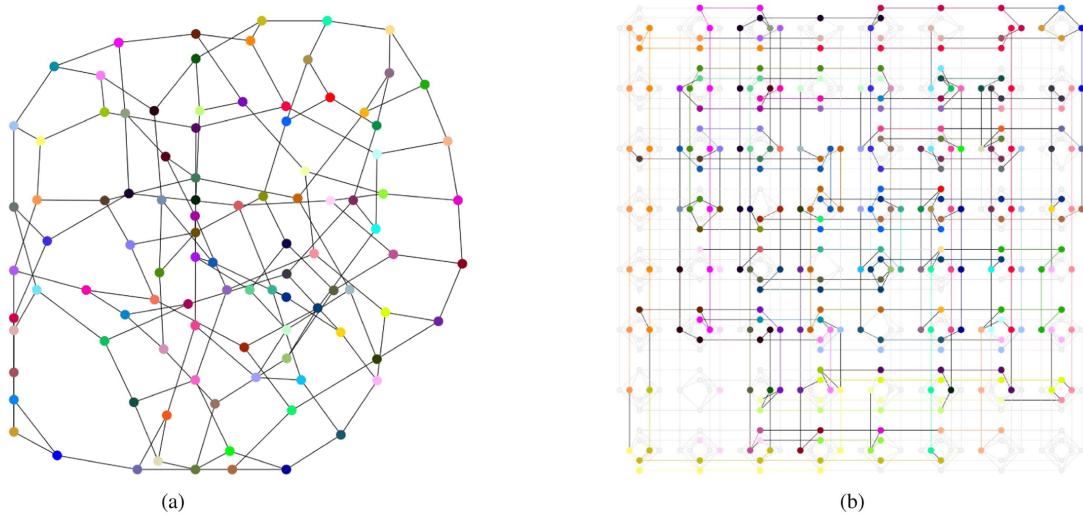[1][Online]. Available: https://gitlab.jsc.fz-juelich.de/sdlrs/qmsvm

Fig. 1. Graphical representation of the minor embedding step. In the graph shown in (a), each node represents a binary variable, and each edge represents a logical connection between two variables of the QUBO problem. In (b), each color represents the corresponding variable embedded into a qubit chain. (a) QUBO graph. (b) Embedding of the QUBO graph on a Pegasus architecture. Source: [6].

problems. In this case, the Hamiltonian of the system is defined as

$$\mathcal{H}(t) = \mathcal{H}_F + \Gamma(t)\mathcal{H}_D \tag{3}$$

where $\mathcal{H}_F$ is the final Hamiltonian, $\Gamma(t)$ is the *transverse field coefficient* as a function of time t, and $\mathcal{H}_D$ is the *transverse field Hamiltonian* (also called *disorder Hamiltonian*). $\mathcal{H}_F$ encodes the objective function, and its ground state is the solution of the optimization problem. $\Gamma(t)$ is a decreasing function, equal to 0 for $t = t_f$. It controls the contribution of $\mathcal{H}_D$, which enables traversability of the solution space, making the optimization process escape local minima. As for this aspect, QA presents a similarity with simulated annealing [46], where the temperature parameter T resembles the role of $\Gamma(t)$. In this framework, the assumptions of the adiabatic theorem are relaxed, i.e., there is no requirement for the quantum system to be closed and to operate in the ground state. The implementation of QA provided by D-Wave quantum annealers enables the solution of a specific type of optimization problems, called QUBO problems. A QUBO problem is defined as

$$\text{minimize} \quad \sum_{i<j} a_i Q_{ij} a_j \tag{4}$$

where $a_i \in \{0, 1\}$ are the binary variables of the problem and Q an upper-triangular matrix called QUBO matrix.

B. Minor Embedding

The hardware architecture of D-Wave quantum annealers poses a limitation on the QUBO problems that can be solved. The most relevant specifications are the number of qubits, the number of couplers (i.e., the physical connections between pairs of qubits), and the qubit connectivity (i.e., the average number of couplers connected to a qubit). D-Wave Advantage is based on the Pegasus architecture and has approximately 5000 qubits,

35 000 couplers, and a qubit connectivity of 15 [45]. When the QUBO problem is submitted to a quantum annealer, a step called *minor embedding* [47] is performed, in which each binary variable of the problem a_i is embedded into a qubit chain. The main requirement is maintaining the logical structure of the problem, described by Q. Each element of the QUBO matrix Q_{ij} represents a logical relation between the variables a_i and a_j. The coefficients Q_{ij} are mapped to the strength of the couplers connecting the qubit chains assigned to the variables a_i and a_j. The existence of such an embedding is a requirement for a problem to be solved by the annealer, i.e., constraints on the dimension and the structure of the QUBO problem need to be satisfied. In Fig. 1, the embedding of a QUBO problem in graph form is shown.

III. QMSVM FORMULATION

In this section, a novel algorithm called QMSVM is described. It is based on a reformulation of the CS SVM [41] as a QUBO problem. The followed steps are adapted from the QSVM proposed in [40], with the addition of a solution combination method. As a starting point, the CS SVM formulation is described in the following.

A. Crammer–Singer Multiclass SVM

In a supervised multiclass classification problem, let N be the number of training examples, C be the number of classes, $X^{\text{tr}} = \{\mathbf{x}_n\}$ be the feature vectors of dimension F, and $Y^{\text{tr}} = \{y_n\}$ be the labels. The training consists in the solution of the following quadratic program:

$$\text{minimize} \quad F(T) = \frac{1}{2} \sum_{n_1, n_2 = 0}^{N-1} K(\mathbf{x}_{n_1}, \mathbf{x}_{n_2}) \sum_{c=0}^{C-1} \tau_{n_1 c} \tau_{n_2 c}$$
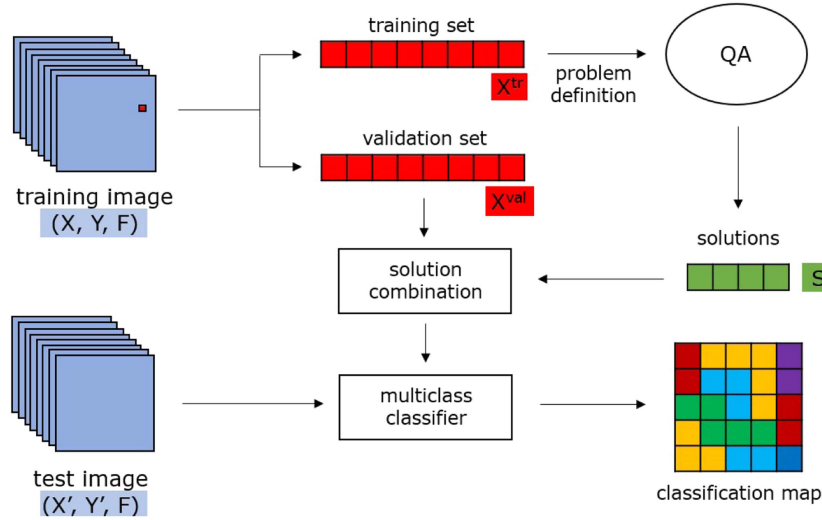
Fig. 2. Flowchart of the QMSVM algorithm. A training set X^{tr} is given as input to the QA step, which obtains a set of S solutions to the training problem. The solutions are then combined with a weighted average based on the accuracy performance on a validation set X^{val} of the classifiers obtained by the single solutions. The final classifier is then used to generate classification maps of test images.

$$- \beta \sum_{n=0}^{N-1} \sum_{c=0}^{C-1} \delta_{cy_n} \tau_{nc} \qquad (5)$$

$$\text{subject to} \quad \sum_{c=0}^{C-1} \tau_{nc} = 0 \quad \forall n, \quad \tau_{nc} \leq 0 \quad \forall n, \forall c \neq y_n \qquad (6)$$

where $T = [\tau_{nc}]$ is the matrix of the NC problem variables, with $n = 0, \ldots, N-1$, $c = 0, \ldots, C-1$, and $\tau_{nc} \in [-1, 1]$, δ_{ij} is the Kronecker delta, and β a regularization parameter.

D-Wave Advantage is unable to directly solve (5) and (6). Therefore, a reformulation of (5) and (6) as a QUBO problem is necessary. The followed steps are: choosing a binary encoding (see Section III-B), defining the penalty terms (see Section III-C), deriving the QUBO matrix including the results of the previous steps in the cost function (see Section III-D), and defining a solution combination method (see Section III-E).

B. Binary Encoding

The first step consists in defining the binary variables a_i of the QUBO problem. In the CS formulation, the problem variables τ_{nc} are real numbers. The idea is to discretize the solution space using uniform sampling and represent each value as a set of B binary variables. First, the following intermediate variable is defined:

$$\sigma_{nc} = \sum_{b=0}^{B-1} 2^b a_{nCB+cB+b}. \qquad (7)$$

σ_{nc} is an integer value in $[0, 2^B - 1]$ represented by the binary encoding $\{a_{nCB+cB+b}\}$, $b = 0, \ldots, B-1$. Then, the problem variables τ_{nc} can be defined from σ_{nc} as

$$\tau_{nc} = -1 + \frac{2}{2^B - 1} \sigma_{nc} = -1 + \frac{2}{2^B - 1} \sum_{b=0}^{B-1} 2^b a_{nCB+cB+b}. \qquad (8)$$
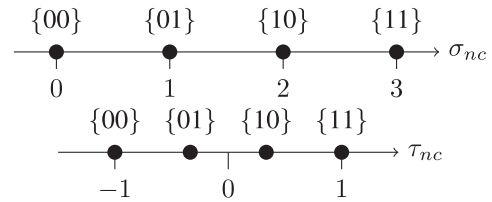

Fig. 3. Representation of the chosen variable sampling and encoding for $B = 2$.

With this definition, it can be proven that each value of τ_{nc} lies in $[-1, 1]$ and the interval is uniformly sampled.

Fig. 3 shows the sampling of (8) for $B = 2$, i.e., in the case, each sample of τ_{nc} is represented by two binary variables, indicated above each sample. Since the total number of problem variables is NC (each variable is associated with an example and a class), the whole optimization space can be represented by a set of NCB binary variables $\{a_0, a_1, \ldots, a_{NCB-1}\}$.

C. Penalty Terms

Another requirement is to include the constraints of (6), as no constraints can be directly enforced in a QUBO problem. A possibility is to add the constraints to the QUBO matrix as weighted positive penalty terms. For the first constraint, the penalty term needs to increase in the case the difference between the value of the sum and 0 increases. In addition, a penalty term needs to be associated with each training example and with the same weight. Since a quadratic polynomial term is required, the following penalty term is chosen:

$$P_n^1 = \left(\sum_{c=0}^{C-1} \tau_{n,c} \right)^2. \qquad (9)$$

For the second constraint, which is an inequality, it is sufficient to directly consider τ_{nc} as the penalty term associated with each training sample and each class. A coefficient $(1 - \delta_{cy_n})$ is attached to account for the case $c = y_n$, in which the penalty is zero

$$P_{nc}^2 = (1 - \delta_{cy_n})\tau_{n,c}. \tag{10}$$

The final penalty term can be written as

$$
\begin{aligned}
P &= \sum_{n=0}^{N-1} P_n^1 + \sum_{n=0}^{N-1} \sum_{c=0}^{C-1} P_{nc}^2 \\
&= \sum_{n=0}^{N-1} \left(\sum_{c=0}^{C-1} \tau_{n,c} \right)^2 + \sum_{n=0}^{N-1} \left(\sum_{c=0}^{C-1} (1 - \delta_{cy_n})\tau_{n,c} \right).
\end{aligned} \tag{11}
$$

Note that P_n^1 and P_{nc}^2 are included with the same weight. The following reasons behind this choice can be listed.
1) Considering two different weights would increase the number of hyperparameters of the optimization problem and the already high complexity of the tuning phase.
2) The two constraints have to be both equally satisfied.
3) The two penalty terms have approximately the same order of magnitude, as $\tau_{nc} \in [-1, 1]$, so there is no imbalance in values.

D. QUBO Matrix

The QUBO problem can now be written by adding to (5) the penalty term in (11) multiplied by a weight μ and substituting $\tau_{i,j}$ with the encoding in (8). The energy function E can be written in the following form:[2]

$$
\begin{aligned}
E = \ & F + \mu P = \sum_{n_1 n_2 c_1 c_2 b_1 b_2} a_{n_1 CB + c_1 B + b_1} \\
& \widetilde{Q}_{n_1 CB + c_1 B + b_1, n_2 CB + c_2 B + b_2} a_{n_2 CB + c_2 B + b_2}.
\end{aligned} \tag{12}
$$

$\widetilde{Q}$ is a symmetric matrix of size $NCB \times NCB$. It can be analytically derived by neglecting the terms not depending on the binary variables and is equal to

$$
\begin{aligned}
& \widetilde{Q}_{n_1 CB + c_1 B + b_1, n_2 CB + c_2 B + b_2} \\
=\ & \delta_{n_1 n_2} \delta_{c_1 c_2} \delta_{b_1 b_2} \frac{2^{b_1 + 1}}{2^B - 1} \left(- \sum_i K(\mathbf{x}_{n_1}, \mathbf{x}_i) \right. \\
& \left. - \delta_{c_1 y_{n_1}}(\beta + \mu) - 2C\mu + \mu \right) \\
& + \delta_{c_1 c_2} \frac{2^{b_1 + b_2 + 1}}{(2^B - 1)^2} K(\mathbf{x}_{n_1}, \mathbf{x}_{n_2}) + \delta_{n_1 n_2} \frac{2^{b_1 + b_2 + 2} \mu}{(2^B - 1)^2}.
\end{aligned} \tag{13}
$$

---

[2]In this formulation, a simplified notation for the sums is used, as the range of the indices is unaltered and redundant.

The upper-triangular QUBO matrix Q can be computed from $\widetilde{Q}$ as

$$
Q_{ij} = \begin{cases} \widetilde{Q}_{ij}, & \text{for } i = j \\ \widetilde{Q}_{ij} + \widetilde{Q}_{ji}, & \text{for } i < j \\ 0, & \text{otherwise} \end{cases}. \tag{14}
$$

E. Solution Combination

Once the QUBO matrix is defined, the problem can submitted to the quantum annealer, assuming the existence of an embedding. As the annealing process is performed multiple times, depending on the value of *num_reads*, the obtained output is a set of *num_reads* solutions. The best S solutions are selected, i.e., $T_i = [\tau_{nc}]_i, i = 0, \ldots, S - 1$, ranked by the value of the energy function $E(T_i)$. During the experiments, it has been noticed that there is no perfect correlation between solutions with lower energy and better classification accuracy of the obtained classifier. Note also that the solution space investigated by the QA algorithm is discrete, due to the variable sampling, so the obtained individual solutions are likely to be suboptimal. For these reasons, a solution combination is performed in order to obtain an optimal final solution. A weighted average is performed, where the weights w_s for each solution s are set according to the prediction accuracy of the obtained classifiers on a validation set $\{X^{\text{val}}, Y^{\text{val}}\}$. In particular, the solutions above a certain threshold accuracy are selected, and their weight is computed applying the softmax function to $multiplier \cdot accuracy_s$, where $multiplier$ is a real value and $accuracy_s$ is the accuracy of the classifier defined by the sth solution on the whole training set. The rest of the weights are set to 0. The combined solution is computed as

$$
\bar{T} = \frac{1}{S} \sum_{s=0}^{S-1} w_s T_s. \tag{15}
$$

The resulting variables $\bar{\tau}_{nc}$ are then used to classify new examples

$$
H(\mathbf{x}) = \arg\max_c \left\{ \sum_{n=0}^{N-1} \bar{\tau}_{nc} K(\mathbf{x}, \mathbf{x}_n) \right\}. \tag{16}
$$

Algorithm 1 summarizes the implemented computational steps required for the training. Fig. 2 visually represents the flowchart of the algorithm for training and testing.

IV. ALGORITHM VALIDATION

A. Experimental Setup

The QMSVM algorithm has been validated on a semantic segmentation problem applied to multispectral RS images. Two different datasets are considered, i.e., SemCity Toulouse [48] (hereafter "Toulouse") and ISPRS Potsdam [49] (hereafter "Potsdam"). Table I describes the selected datasets. While both represent urban areas, the two datasets differ in the features and the ground resolution. Toulouse is based on eight-band Worldview-2 data. We chose the 50-cm multispectral images obtained through pansharpening of the 2-m multispectral data

---

Algorithm 1: Quantum Multiclass SVM.

Input:$X^{\text{tr}}, Y^{\text{tr}}, X^{\text{val}}, Y^{\text{val}}, C, B, \beta, \mu, \gamma, S, \text{multiplier}$
Output:$\bar{T}$
    QUBO matrix initialization, (13), (14)
1:   $Q \leftarrow \text{QUBO\_MATRIX}(X^{tr}, Y^{tr}, C, B, \beta, \mu, \gamma)$
    Run annealing step and sample num_reads solutions
2:   $T \leftarrow \text{QUANTUM\_ANNEALING}(Q)$
    Evaluate classifier on validation set, (16)
3:   **for** $s = 1$ to S **do**
4:     $\hat{Y}^{\text{val}}[s] \leftarrow H(X^{\text{val}}, Y^{\text{val}}, T)$
5:     $\text{accuracy}[s] \leftarrow \text{ACCURACY}(\hat{Y}^{\text{val}}[s], Y^{\text{val}})$
6:   **end for**
    Weights calculation
7:   $\text{threshold} \leftarrow \text{THRESHOLD}(\text{accuracy})$
8:   **for** $s = 1$ to S **do**
9:     **if** $\text{accuracy}[s] < \text{threshold}$ **then**
10:      $\text{accuracy}[s] \leftarrow 0$
11:     **end if**
12: **end for**
13: $W \leftarrow \text{SOFTMAX}(\text{multiplier} \cdot \text{accuracy})$
    Solution combination, eq. (15)
14: $\bar{T} \leftarrow \text{COMBINE}(T, W)$
15: **return** $\bar{T}$

---

with 50-cm panchromatic images using the BDSD algorithm. Potsdam is based on aerial data collection and, on top of four spectral bands, a DSM is provided, which is used as a feature. The complete ground truths are obtained by expert annotators. For each dataset, five training sets of N examples are randomly initialized, from the training tiles indicated in Table I.

The experiments have been performed on a real quantum device, JUPSI [50], a D-Wave Advantage quantum annealer located at Forschungszentrum Jülich. To access the machine, the *Advantage_system5.4* cloud solver has been used. Given the memory and connectivity limitation of the machine, the training set $\{X^{\text{tr}}, Y^{\text{tr}}\}$ defined in Section III-A is initialized as a subset of M examples from the total number of training examples N. The training subset is computed through an example selection step. Two selection methods have been tested.

1) *Random selection:* M random examples are selected from the whole training set. It is a fast and straightforward method, enforcing no selection criterion.
2) *K-means selection:* k-means clustering [51] is applied to each of the C classes, with $k = \frac{M}{C}$, and the obtained M centroids are used as selected examples. It is inspired by undersampling techniques in imbalanced classification [52]. In principle, the method is designed to select meaningful examples, covering the whole feature domain.

The whole training set is then used as the validation set $\{X^{\text{val}}, Y^{\text{val}}\}$ for the solution combination, introduced in Section III-E.[3] The *threshold* accuracy, which determines which

---

[3]Note that the training sets are named differently here. In the formulation, the general terms "training set" and "validation set" are used, coherently with ML literature. Here, the terms "training subset" and "training set" are used, respectively, coherently with our training setup, which is far from being universal.
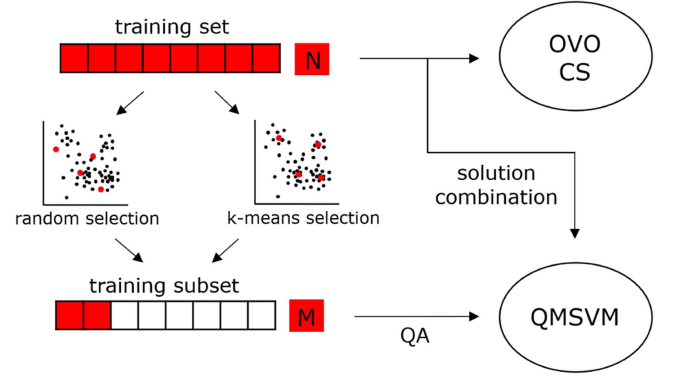

Fig. 4. Training setup. The OVO and CS methods have been trained on a training set of N examples. For QMSVM, a subset of M examples is selected and used by the annealing algorithm, while the solution combination is performed based on the accuracy obtained on the whole training set.

solutions are discarded in the combination, has been computed as

$$\text{threshold} = 0.2 \cdot \min(\text{accuracy}) + 0.8 \cdot \max(\text{accuracy}). \tag{17}$$

The results are compared with three standard implementations of the multiclass SVM, i.e., the OVO and OVA implementation in Scikit-learn [53] and a CS SVM implementation in C++ [54]. A Gaussian kernel with parameter γ is chosen. The training setup is depicted in Fig. 4.

B. Parameters

In Table II, the parameters of the problem are described. The parameters β, μ, and γ are set through a simple grid search optimization, training the model on a smaller training subset of 20 examples and validating it on the whole training set. Different values of N are chosen in order to analyze the performance of the method by varying the number of available examples. The highest tested value is $N = 20000$ for Toulouse and $N = 15000$ for Potsdam. The parameters B, M, and *max_min_ratio* are related to the main limitation of the QA, i.e., the number of qubits and couplers. As previously discussed, finding an embedding on the given qubit architecture is required for solving the QUBO problem. This is achieved in case Q is sufficiently small, sufficiently sparse, or both. Using only the selected training subset, the dimension of Q is MCB. Thus, M is limited, which is the reason why the QA is unable to use an arbitrarily large training set and example selection is performed in the first place. To maximize the number of examples fitting in the QA, the remaining parameters are kept low, i.e., $C = 3$ and $B = 2$. Considering a higher number of classes, i.e., $C > 3$, would require using a lower number of examples M, which degrades the overall performance. Regarding sparsity, a straightforward operation is performed, i.e., pruning the values of Q below the threshold defined by *max_min_ratio*, chosen empirically. This simplification is acceptable, as relatively low values would be mapped to relatively low strengths in the QA, which mildly affect the annealing process. The parameters *num_reads*, *chain_strength*, and *annealing_time* are related to the annealer setup. Their values are chosen empirically, finding a tradeoff

TABLE I
DATASETS USED IN THE EXPERIMENTS

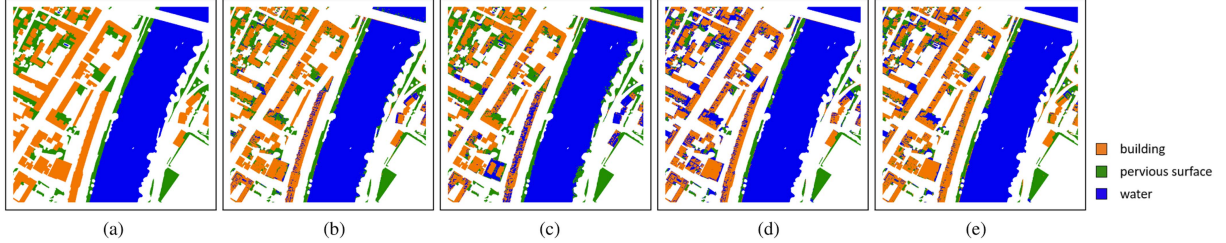| Dataset	Dimension	Resolution	Features	Classes	Training Set	Test Set
SemCity Toulouse [48]	16 tiles, 3504×3452	50 cm	8 bands	$C \leq 7$	N samples, tile 4	800×800 area, tile 8
ISPRS Potsdam [49]	38 tiles, 6000×6000	5 cm	4 bands + DSM	$C \leq 5$	N samples, tile 6_9	1000×1000 area, tile 6_10

Fig. 5. Toulouse—ground truth and predicted land cover maps on an 800×800 selected area from tile 8 for OVO, OVA, CS, and QMSVM using $N = 20000$ training examples. The considered classes are "building" (orange), "pervious surface" (green), and "water" (blue). (a) Ground truth. (b) OVO. (c) OVA. (d) CS. (e) QMSVM.
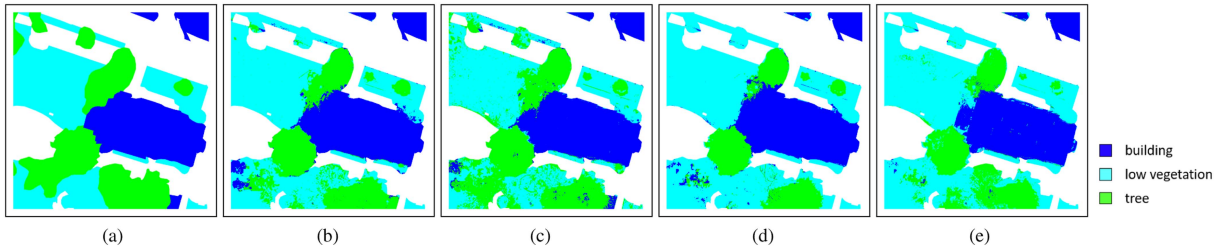

Fig. 6. Potsdam—ground truth and predicted land cover maps on a 1000×1000 selected area from tile 6_10 for OVO, OVA, CS, and QMSVM using $N = 15000$ training examples. The considered classes are "building" (blue), "low vegetation" (light blue), and "tree" (green). (a) Ground truth. (b) OVO. (c) OVA. (d) CS. (e) QMSVM.
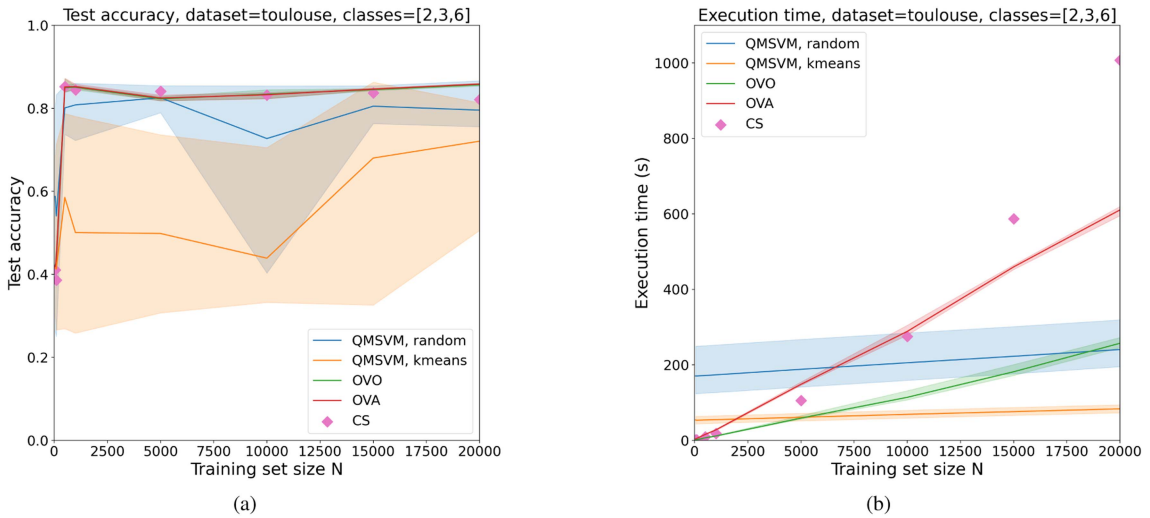

Fig. 7. Toulouse—test accuracy and execution time for QMSVM, OVO, OVA, and CS SVM with respect to training set size N. (a) Test accuracy versus training set size N. (b) Execution time versus training set size N.

between total run time and quality of the solutions on the small validation set, testing a range of values that have been considered valid in previous work, e.g., in [55]. The remaining parameters are set arbitrarily. Table II summarizes the chosen parameter values.

C. Results

In the test phase, the methods are evaluated on a three-class classification problem on a selected area taken from the dataset. For Toulouse, an 800×800 test area from tile 8 has been selected, and the classes "building," "pervious surface," and
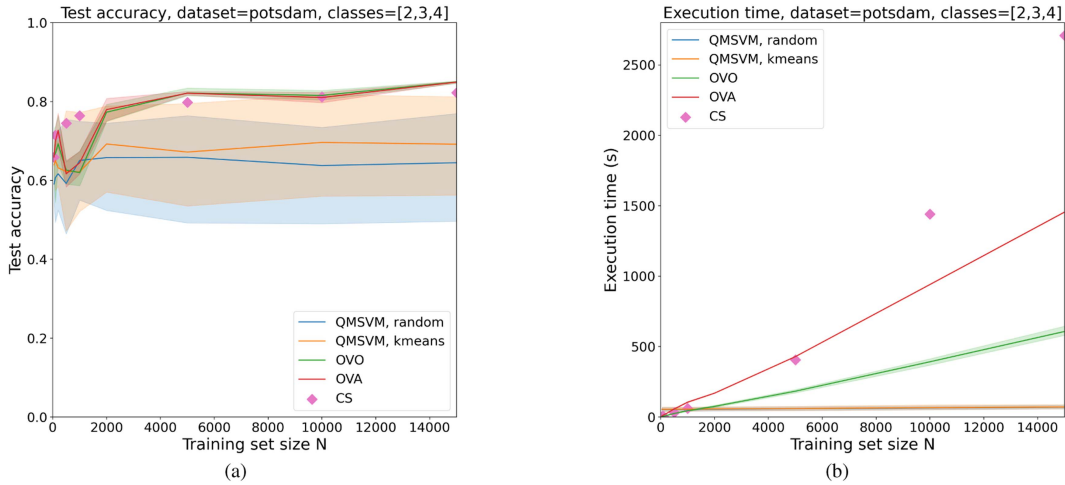
Fig. 8. Potsdam—test accuracy and execution time for QMSVM, OVO, OVA, and CS SVM with respect to training set size N. (a) Test accuracy versus training set size N. (b) Execution time versus training set size N.

TABLE II
PARAMETERS OF THE SETUP

Variable Name	Value	Description
C	3	Number of classes considered in the multiclass classification problem
B	2	Number of binary variables representing each problem variable τ_{nc}
β	0.1	Model related regularization parameter, introduced in Eq. (5)
μ	1	Weight of the penalty term P added to the energy function defined in Eq. (12)
γ	0.5	Gaussian kernel parameter, regulating its radius
N	$[50, 20000]$	Total number of training examples used in the training (multiple values analyzed)
M	60	Number of selected training examples, used to define the QUBO matrix Q submitted to the QA
num_reads	1000	Number of times the annealing schedule is performed and how many solutions are sampled for each run
S	100	Number of solutions selected among the total number *num_reads*, used for the solution combination
multiplier	10	Regulates the balance between higher and lower accuracy values over the combined solutions
max_min_ratio	5	Ratio between the maximum and minimum non-zero absolute value of the QUBO matrix Q
chain_strength	1	Relative coupling strength between qubits that form a chain and qubits in different chains
annealing_time	100	Time (in µs) at which the measurement is performed after starting the annealing schedule

"water" have been considered. For Potsdam, a 1000×1000 test area from tile 6_10 has been selected, and the classes "building," "low vegetation," and "tree" have been considered.

The method is evaluated according to both test accuracy and execution time. For OVO, OVA, and CS, training and inference are considered. For QMSVM, the time measurement includes preprocessing (example selection), training (annealing), postprocessing (solution combination), and inference time. Figs. 5 and 6 show the ground truth of the selected area and the ground maps obtained by OVO, CS, and QMSVM. In Figs. 7–10, the performance of the analyzed methods on the Toulouse and Potsdam datasets is shown in terms of both test accuracy and execution time, and for both example selection methods, i.e., random and k-means selection. Table III summarizes the best obtained results in terms of test accuracy, along with the respective F_1 score. Accuracy and F_1 score are computed as

$$\text{accuracy} = \frac{\text{correct predictions}}{\text{no. of predictions}}, \quad F_1 = \text{average}_c(F_{1c}). \quad (18)$$

TABLE III
RESULTS OF ACCURACY, F_1 SCORE, AND EXECUTION TIME FOR OVO AND QMSVM

Solver	N	M	Accuracy	F1	t (s)
Toulouse - best accuracy, maximum N					
OVO	20000	-	0.8598	0.8612	248.02
OVA	20000	-	0.8606	0.8617	595.50
CS	20000	-	0.8213	0.8279	1007.19
QMSVM (random)	20000	60	0.8663	0.8721	245.39
QMSVM (k-means)	20000	60	0.8133	0.8178	85.72
Potsdam - best accuracy, maximum N					
OVO	15000	-	0.8521	0.8530	580.17
OVA	15000	-	0.8514	0.8523	1317.35
CS	15000	-	0.8226	0.8362	2708.62
QMSVM (random)	15000	60	0.7695	0.7840	84.20
QMSVM (k-means)	15000	60	0.8123	0.8143	58.89

F_{1c} is the F_1 score computed for each class c with respect to the remaining classes

$$F_{1c} = \frac{\text{TP}_c}{\text{TP}_c + \frac{1}{2}(\text{FN}_c + \text{FP}_c)} \quad (19)$$

where TP (true positive), FN (false negative), and FP (false positive) predictions are referred to the class c. For each method, the average results on the five training sets are plotted and the space between the best and the worst obtained results is highlighted. For CS, experiments on only one training set have been performed, due to the slow training.

V. DISCUSSION

A. QMSVM on D-Wave Advantage

Running the proposed algorithm on the D-Wave Advantage quantum annealer requires particular attention to its functioning. First of all, a QUBO formulation needs to be derived from the original problem, which is not always possible. The analytical derivation we provided successfully reframed the problem, although with some inevitable changes and arbitrary choices, e.g., on the optimization variable domain and the constraint satisfaction. Then, as previously mentioned, the main limitation of quantum annealers is in the maximum problem size that can be submitted. We tackle this limitation by using only a small subset of the training set and then refining the obtained solutions using the whole training set. Understanding how many examples we should consider and how sparse the QUBO matrix should be requires some experience given by trial and error. The parameters provided can be used as a reference for future work.

B. Test Accuracy

The results show that QMSVM can reach a comparable or higher classification accuracy with respect to its classical counterpart, i.e., CS SVM. This is a promising result, given the limitations discussed in Section V-A, in particular the small number of training examples M that QMSVM can handle in the optimization step. The reason is that the solution combination using N examples is able to improve the quality of the final solution on average by increasing N, as seen in the accuracy plots. However, the prediction accuracy of OVO and OVA is slightly higher on average than both CS and QMSVM for higher N. An even worse performance of QMSVM can be clearly seen on the more complex Potsdam dataset. The random selection method consistently outperforms k-means selection on Toulouse, while performing slightly better on Potsdam. This shows that k-means is inconsistent in practice and leaves an open question on the path to follow for an optimal example selection method. A high variance in accuracy with respect to the chosen training set should also be mentioned, which makes QMSVM more unstable than the compared standard SVM methods.

C. Time Complexity

The most remarkable result is that the execution time of QMSVM for a high number of training examples N is lower than the execution time of the classical methods considered in the comparison. The different steps included in the measurement of the execution time are shown in detail in Figs. 9 and 10. It can be seen that the most demanding step in the QMSVM is the annealing step. The main reason is the high time complexity of the minor embedding algorithm [47] required to run the QUBO
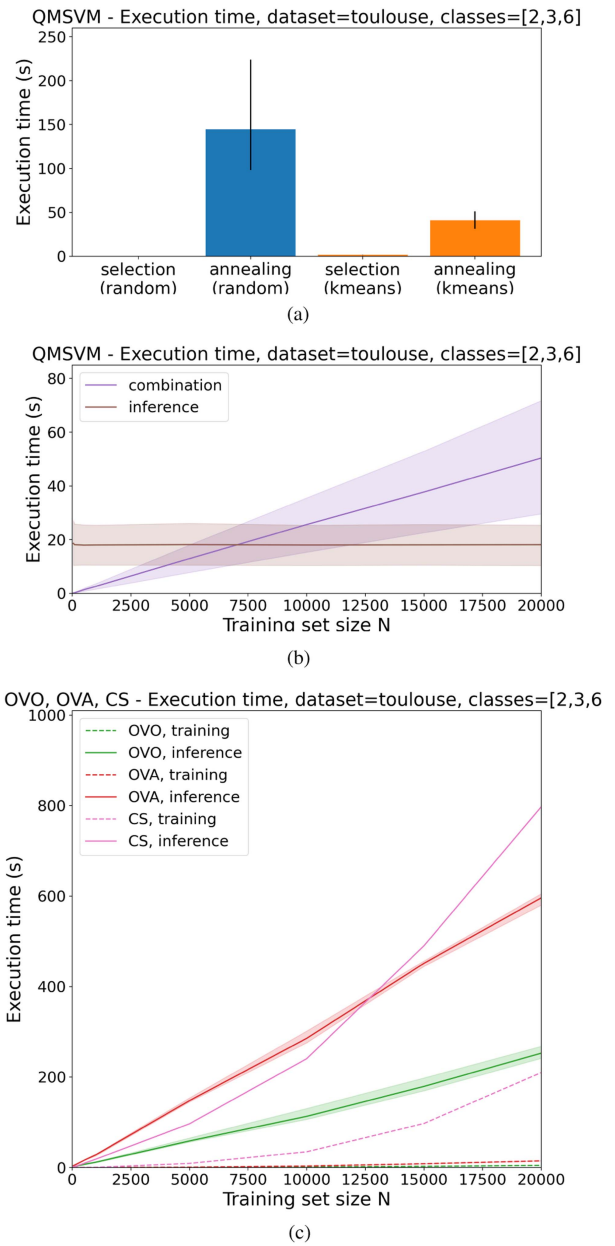

Fig. 9. Toulouse—execution time of each performed step, for QMSVM, OVO, OVA, and CS SVM, with respect to training set size N. (a) QMSVM (selection, annealing)—execution time versus training set size N. (b) QMSVM (combination, inference)—execution time versus training set size N. (c) OVO, OVA, CS—execution time versus training set size N.

problem on the quantum annealer. The interesting results are the linear time increase for the solution combination and the constant time for inference with respect to N. A theoretical explanation supporting the experimental results is summarized as follows.

In the solution combination step, the single solutions obtained by the annealer define S decision functions as in (16). The decision functions are evaluated on N examples and require the computation of the kernel values $K(x, x_m)$ with the M examples of the training subset. The evaluation is repeated for C classes and the class with highest value is chosen. For evaluating all the S classifiers, $O(SCMN)$ operations are needed, leading
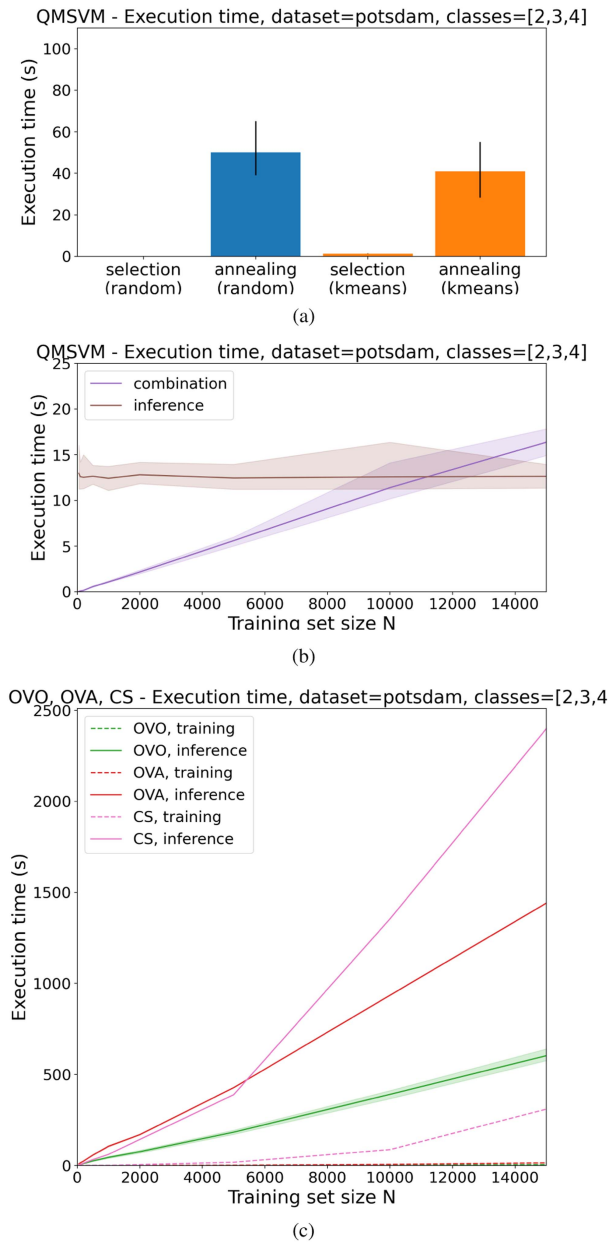
Fig. 10. Potsdam—execution time of each performed step, for QMSVM, OVO, OVA, and CS SVM, with respect to training set size N. (a) QMSVM (selection, annealing)—execution time versus training set size N. (b) QMSVM (combination, inference)—execution time versus training set size N. (c) OVO, OVA, CS—execution time versus training set size N.

to a linear time complexity with respect to the training set size. Similarly, the classifier used in the inference computes the kernel distances between the I test examples and the M training subset examples and it requires $O(CMI)$ operations, i.e., it is independent from N. It is worth mentioning that the speedup of QMSVM is not strictly a quantum speedup. Instead, it is an algorithmic speedup related to the postprocessing of the annealing solutions. The outcomes for OVO and OVA are also coherent with theory: the training complexity of the Scikit-learn SVM implementation is $O(N^3)$, whereas the inference time is linearly dependent on N [56].

These results clearly show that QMSVM is a much more scalable algorithm with respect to the considered training examples N compared to standard multiclass SVM methods. Given that the annealing step, which is a fixed step unrelated to N, has the largest impact in terms of time, the overall execution time can be regarded as near constant.

VI. CONCLUSION

QMSVM serves as a preliminary framework for applying QA to a single-step multiclass SVM algorithm, successfully leveraging the D-Wave Advantage quantum annealer in the training step. Although the results show that the prediction accuracy is not always higher than standard multiclass SVM algorithms trained on the same training set, especially using the more complex Potsdam dataset (compared to OVO: 0.8663 versus 0.8598 on Toulouse, 0.8123 versus 0.8521 on Potsdam), the improved scalability allows the usage of large-scale datasets, due to the lower total execution time (compared to OVO: 85.72 versus 248.02 s on Toulouse, 58.89 versus 580.17 s on Potsdam). Further research has to be conducted, in light of the promising achieved results. Time and accuracy analysis can be performed on different datasets, better assessing the impact of N and the model parameters on the prediction accuracy. A deeper analysis of different selection methods based on dataset representativeness, on top of the k-means method, can improve the quality of the solutions obtained by the QA. An improvement in performance for QMSVM is expected with the future development of QA, as a higher memory and qubit connectivity allows the usage of a larger training subset and enhances the quality of the obtained solutions.

REFERENCES

[1] M. Chi, A. Plaza, J. A. Benediktsson, Z. Sun, J. Shen, and Y. Zhu, "Big data for remote sensing: Challenges and opportunities," *Proc. IEEE*, vol. 104, no. 11, pp. 2207–2219, Nov. 2016.

[2] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*, 10th Anniversary ed. Cambridge, U.K.: Cambridge Univ. Press, Dec. 2010.

[3] R. Jozsa and N. Linden, "On the role of entanglement in quantum-computational speed-up," *Proc. Roy. Soc. London. Ser. A: Math., Phys. Eng. Sci.*, vol. 459, pp. 2011–2032, Aug. 2003.

[4] T. F. Rønnow et al., "Defining and detecting quantum speedup," *Science*, vol. 345, no. 6195, pp. 420–424, 2014.

[5] D. Aharonov, A. Kitaev, and N. Nisan, "Quantum circuits with mixed states," in *Proc. 30th Annu. ACM Symp. Theory Comput.*, 1998, pp. 20–30.

[6] C. C. McGeoch, *Adiabatic Quantum Computation and Quantum Annealing: Theory and Practice*. San Rafael, CA, USA: Morgan & Claypool, 2014.

[7] T. Albash and D. A. Lidar, "Adiabatic quantum computation," *Rev. Mod. Phys.*, vol. 90, Jan. 2018, Art. no. 015002.

[8] M. Born and V. Fock, "Beweis des adiabatensatzes," *Zeitschrift für Physik*, vol. 51, no. 3, pp. 165–180, Mar. 1928.

[9] D. Aharonov, W. van Dam, J. Kempe, Z. Landau, S. Lloyd, and O. Regev, "Adiabatic quantum computation is equivalent to standard quantum computation," *SIAM Rev.*, vol. 50, no. 4, pp. 755–787, 2008.

[10] T. Kadowaki and H. Nishimori, "Quantum annealing in the transverse ising model," *Phys. Rev. E*, vol. 58, no. 5, pp. 5355–5363, 1998.

[11] A. Finnila, M. Gomez, C. Sebenik, C. Stenson, and J. Doll, "Quantum annealing: A new method for minimizing multidimensional functions," *Chem. Phys. Lett.*, vol. 219, no. 5, pp. 343–348, 1994.

[12] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, "Quantum machine learning," *Nature*, vol. 549, no. 7671, pp. 195–202, 2017.

[13] V. Dunjko and H. J. Briegel, "Machine learning and artificial intelligence in the quantum domain: A review of recent progress," *Rep. Prog. Phys.*, vol. 81, no. 7, Art. no. 074001, Jun. 2018.

[14] R. Y. Li, R. D. Felice, R. Rohs, and D. A. Lidar, "Quantum annealing versus classical machine learning applied to a simplified computational biology problem," *NPJ Quantum Inf.*, vol. 4, no. 1, pp. 1–10, Feb. 2018.

[15] P. Gawron and S. Lewiński, "Multi-spectral image classification with quantum neural network," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2020, pp. 3513–3516.

[16] A. Sebastianelli, D. A. Zaidenberg, D. Spiller, B. L. Saux, and S. L. Ullo, "On circuit-based hybrid quantum neural networks for remote sensing imagery classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 15, pp. 565–580, 2022.

[17] F. Fan, Y. Shi, and X. X. Zhu, "Urban land cover classification from Sentinel-2 images with quantum-classical network," in *Proc. Joint Urban Remote Sens. Event*, 2023, pp. 1–4.

[18] M. van Waveren et al., "Comparison of quantum neural network algorithms for Earth observation data classification," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2023, pp. 780–783.

[19] S. Otgonbaatar and M. Datcu, "A quantum annealer for subset feature selection and the classification of hyperspectral images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 7057–7065, 2021.

[20] S. Otgonbaatar and M. Datcu, "Classification of remote sensing images with parameterized quantum gates," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, 2022, Art. no. 8020105.

[21] S. Otgonbaatar, G. Schwarz, M. Datcu, and D. Kranzlmüller, "Quantum transfer learning for real-world, small, and high-dimensional remotely sensed datasets," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sensing*, vol. 16, pp. 9223–9230, 2023, doi: 10.1109/JSTARS.2023.3316306.

[22] R. U. Shaik and S. Periasamy, "Accuracy and processing speed trade-offs in classical and quantum SVM classifier exploiting PRISMA hyperspectral imagery," *Int. J. Remote Sens.*, vol. 43, nos. 15/16, pp. 6176–6194, 2022.

[23] M. K. Gupta, M. Romaszewski, and P. Gawron, "Potential of quantum machine learning for processing multispectral Earth observation data," 2023, *TechRxiv*.

[24] A. Miroszewski et al., "Detecting clouds in multispectral satellite images using quantum-kernel support vector machines," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 16, pp. 7601–7613, 2023.

[25] A. G. Pai, K. M. Buddhiraju, and S. S. Durbha, "Multiclass classification of hyperspectral remote sensed data using QSVC," *Proc. SPIE*, vol. 12262, 2022, Art. no. 122620P.

[26] S. Huber, K. Glatting, G. Krieger, and A. Moreira, "Quantum annealing for SAR system design and processing," in *Proc. 14th Eur. Conf. Synthetic Aperture Radar*, 2022, pp. 1–6.

[27] S. Otgonbaatar and M. Datcu, "Quantum annealer for network flow minimization in InSAR images," in *Proc. 13th Eur. Conf. Synthetic Aperture Radar*, 2021, pp. 1–4.

[28] S. Otgonbaatar and M. Datcu, "Natural embedding of the stokes parameters of polarimetric synthetic aperture radar images in a gate-based quantum computer," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 4704008.

[29] G. Cavallaro, D. Willsch, M. Willsch, K. Michielsen, and M. Riedel, "Approaching remote sensing image classification with ensembles of support vector machines on the D-wave quantum annealer," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2020, pp. 1973–1976.

[30] A. Delilbasic, G. Cavallaro, M. Willsch, F. Melgani, M. Riedel, and K. Michielsen, "Quantum support vector machine algorithms for remote sensing data classification," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2021, pp. 2608–2611.

[31] Y. Ma and G. Guo, *Support Vector Machines Applications*, vol. 649. New York, NY, USA: Springer, 2014.

[32] F. Melgani and L. Bruzzone, "Classification of hyperspectral remote sensing images with support vector machines," *IEEE Trans. Geosci. Remote Sens.*, vol. 42, no. 8, pp. 1778–1790, Aug. 2004.

[33] C.-W. Hsu and C.-J. Lin, "A comparison of methods for multiclass support vector machines," *IEEE Trans. Neural Netw.*, vol. 13, no. 2, pp. 415–425, Mar. 2002.

[34] A. K. Bishwas, A. Mani, and V. Palade, "An all-pair quantum SVM approach for big data multiclass classification," *Quantum Inf. Process.*, vol. 17, pp. 1–16, Oct. 2018.

[35] B. A. Dema, J. Arai, and K. Horikawa, "Support vector machine for multiclass classification using quantum annealers," in *Proc. DEIM Forum*, 2020.

[36] X.-J. Yuan et al., "Quantum support vector machines for aerodynamic classification," *Intell. Comput.*, vol. 2, 2023, Art. no. 0057, doi: 10.34133/icomputing.0057.

[37] P. Rebentrost, M. Mohseni, and S. Lloyd, "Quantum support vector machine for big data classification," *Phys. Rev. Lett.*, vol. 113, no. 13, 2014, Art. no. 130503.

[38] R. Zhang, J. Wang, N. Jiang, H. Li, and Z. Wang, "Quantum support vector machine based on regularized newton method," *Neural Netw.*, vol. 151, pp. 376–384, 2022.

[39] V. Havlíček et al., "Supervised learning with quantum-enhanced feature spaces," *Nature*, vol. 567, no. 7747, pp. 209–212, 2019.

[40] D. Willsch, M. Willsch, H. D. Raedt, and K. Michielsen, "Support vector machines on the D-wave quantum annealer," *Comput. Phys. Commun.*, vol. 248, Jun. 2019, Art. no. 107006.

[41] K. Crammer and Y. Singer, "On the algorithmic implementation of multiclass kernel-based vector machines," *J. Mach. Learn. Res.*, vol. 2, pp. 265–292, Mar. 2002.

[42] J. Weston and C. Watkins, "Support Vector Machines for Multi-class Pattern Recognition," in *Proc. 7th Eur. Symp. Artif. Neural Netw.*, Apr. 1999, pp. 219–224.

[43] Z. Wang and X. Xue, "Multi-class support vector machine," in *Support Vector Machines Applications*. New York, NY, USA: Springer, 2014, pp. 23–48.

[44] "D-Wave systems." Accessed: Aug. 1, 2023. [Online]. Available: https://www.dwavesys.com/

[45] C. McGeoch and P. Farré, "The advantage system: Performance update," D-Wave Syst., Burnaby, BC, Canada, Tech. Rep. 14-1054A-A, 2021.

[46] P. J. M. van Laarhoven and E. H. L. Aarts, *Simulated Annealing*. Dordrecht, The Netherlands:Springer, 1987, pp. 7–15.

[47] J. Cai, W. G. Macready, and A. Roy, "A practical heuristic for finding graph minors," 2014, *arXiv:1406.2741*.

[48] R. Roscher, M. Volpi, C. Mallet, L. Drees, and J. D. Wegner, "SemCity Toulouse: A benchmark for building instance segmentation in satellite images," *ISPRS Ann. Photogrammetry, Remote Sens. Spatial Inf. Sci.*, vol. V-5, pp. 109–116, 2020.

[49] ISPRS, "2D semantic labeling contest—Potsdam." Accessed: Aug. 1, 2023. [Online]. Available: https://www.isprs.org/education/benchmarks/UrbanSemLab/2d-sem-label-potsdam.aspx

[50] "Forschungszentrum Jülich." Accessed: Aug. 1, 2023. [Online]. Available: https://www.fz-juelich.de/en/ias/jsc/systems/quantum-computing/juniq-facility/juniq/d-wave-advantagetm-system-jupsi

[51] J. A. Hartigan and M. A. Wong, "Algorithm AS 136: A K-means clustering algorithm," *J. Roy. Stat. Soc. Ser. C (Appl. Statist.)*, vol. 28, no. 1, pp. 100–108, 1979.

[52] D.-S. Huang, K. Li, G. W. Irwin, S.-J. Yen, and Y.-S. Lee, "Under-sampling approaches for improving prediction of the minority class in an imbalanced dataset," *Intell. Control Autom.*, vol. 344, pp. 731–740, 2006.

[53] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.

[54] T. Joachims, "SVM-Multiclass: Multi-class support vector machine." Accessed: Aug. 1, 2023. [Online]. Available: https://www.cs.cornell.edu/people/tj/svm_light/svm_multiclass.html

[55] D. Willsch et al., "Benchmarking advantage and D-wave 2000Q quantum annealers with exact cover problems," *Quantum Inf. Process.*, vol. 21, pp. 1–22, 2022.

[56] A. Ns and R. Wardoyo, "Time complexity analysis of support vector machines (SVM) in LibSVM," *Int. J. Comput. Appl.*, vol. 128, pp. 975–8887, Oct. 2015.
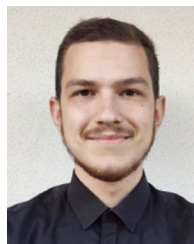
Amer Delilbasic (Student Member, IEEE) received the B.Sc. and M.Sc. degrees in information and communication engineering from the University of Trento, Trento, Italy, in 2019 and 2021, respectively. He is currently working toward the Ph.D. degree in computational engineering with the University of Iceland, Reykjavík, Iceland.

He is currently a Member of the "AI and ML for Remote Sensing" Simulation and Data Lab, Jülich Supercomputing Centre, Forschungszentrum Jülich, Jülich, Germany. He is an External Researcher with the Φ-Lab, European Space Research Institute, European Space Agency, Frascati, Italy. His research interests include machine learning methods for remote sensing applications, with a particular focus on quantum computing and high-performance computing.

Bertrand Le Saux (Senior Member, IEEE) received the Ms.Eng. and M.Sc. degrees in signal processing from the Institut Polytechnique de Grenoble, Grenoble, France, in 1999, the Ph.D. degree in computer science from the University of Versailles/Inria, Versailles, France, in 2003, and the Dr. Habilitation degree in physics from the University of Paris-Saclay, Saclay, France, in 2019.

He is currently a Senior Scientist with the Φ-Lab, European Space Research Institute, European Space Agency, Frascati, Italy. He is interested in tackling practical problems that arise in earth observation to bring solutions to current environment and population challenges. His research interests include visual understanding of the environment by data-driven techniques, including artificial intelligence and (quantum) machine learning.

Dr. Le Saux is an Associate Editor for IEEE GEOSCIENCE AND REMOTE SENSING LETTERS. He was Co-Chair (2015–2017) and Chair (2017–2019) for the Technical Committee on Image Analysis and Data Fusion of the IEEE Geoscience and Remote Sensing Society.

Kristel Michielsen received the Ph.D. degree in simulation of strongly correlated electron systems from the University of Groningen, Groningen, The Netherlands, in 1993.

Since 2009, she has been a Group Leader of the Research Group Quantum Information Processing with the Jülich Supercomputing Centre, Forschungszentrum Jülich, Jülich, Germany. She is also a Professor of Quantum Information Processing with RWTH Aachen University, Aachen, Germany. Her current research interests include quantum computation, quantum annealing, quantum statistical physics, event-based simulation methods of quantum phenomena, logical inference approach to quantum mechanics, and computational electrodynamics.

Morris Riedel (Member, IEEE) received the Ph.D. degree in parallel and distributed systems from the Karlsruhe Institute of Technology, Karlsruhe, Germany, in 2012.

Since 2004, he has been working in data-intensive parallel and distributed systems. Since 2004, he has held various positions with the Jülich Supercomputing Centre, Forschungszentrum Jülich, Jülich, Germany. Since 2020, he has also been the EuroHPC Joint Undertaking Governing Board Member for Iceland. He is currently a Full Professor of high-performance computing with an emphasis on parallel and scalable machine learning with the School of Natural Sciences and Engineering, University of Iceland, Reykjavik, Iceland. He is also the Head of the joint High Productivity Data Processing Research Group between the Jülich Supercomputing Centre and the University of Iceland. His online YouTube and university lectures include High-Performance Computing—Advanced Scientific Computing, Cloud Computing and Big Data—Parallel and Scalable Machine and Deep Learning, as well as Statistical Data Mining. In addition, he has performed numerous hands-on training events in parallel and scalable machine and deep learning techniques on cutting-edge high-performance computing systems. He has authored extensively in his research areas. His research interests include high-performance computing, remote sensing applications, medicine and health applications, pattern recognition, image processing, and data sciences.

Gabriele Cavallaro (Senior Member, IEEE) received the B.Sc. and M.Sc. degrees in telecommunications engineering from the University of Trento, Trento, Italy, in 2011 and 2013, respectively, and the Ph.D. degree in electrical and computer engineering from the University of Iceland, Reykjavik, Iceland, in 2016.

From 2016 to 2021, he was the Deputy Head of the "High Productivity Data Processing" Research Group, Jülich Supercomputing Centre (JSC), Forschungszentrum Jülich, Jülich, Germany. Since 2022, he has been the Head of the "AI and ML for Remote Sensing" Simulation and Data Lab, JSC, and an Adjunct Associate Professor with the School of Natural Sciences and Engineering, University of Iceland. He is a Visiting Professor with the Φ-Lab, European Space Agency, where he contributes to the Quantum Computing for Earth Observation initiative. His research interests include remote sensing data processing with parallel machine learning algorithms that scale on distributed computing systems and innovative computing technologies.

Dr. Cavallaro was the recipient of the IEEE Geoscience and Remote Sensing Society (IEEE GRSS) Third Prize in the Student Paper Competition of IEEE International Geoscience and Remote Sensing Symposium 2015 in Milan, Italy. From 2020 to 2023, he was the Chair for High-Performance and Disruptive Computing in Remote Sensing Working Group of the Earth Science Informatics Technical Committee (ESI TC) of the IEEE GRSS. In 2023, he took on the role of Co-Chair for the ESI TC. He has been an Associate Editor for IEEE TRANSACTIONS ON IMAGE PROCESSING since October 2022.