








Detecting Clouds in Multispectral Satellite Images Using Quantum-Kernel Support Vector Machines

Artur Miroszewski , Jakub Mielczarek , Grzegorz Czelusta , Filip Szczepanek , Bartosz Grabowski , Bertrand Le Saux , *Senior Member, IEEE*, and Jakub Nalepa , *Senior Member, IEEE*

Abstract—Support vector machines (SVMs) are well-established classifiers that are effectively deployed in an array of classification tasks. In this article, we consider extending classical SVMs with quantum kernels and applying them to satellite data analysis. The design and implementation of SVMs with quantum kernels (hybrid SVMs) are presented. Here, the pixels are mapped to the Hilbert space using a family of parameterized quantum feature maps (related to quantum kernels). The parameters are optimized to maximize the kernel-target alignment. The quantum kernels have been selected such that they enable the analysis of numerous relevant properties while being able to simulate them with classical computers on a real-life large-scale dataset. Specifically, we approach the problem of cloud detection in the multispectral satellite imagery, which is one of the pivotal steps in both on-the-ground and on-board satellite image analysis processing chains. The experiments performed over the benchmark Landsat-8 multispectral dataset revealed that the simulated hybrid SVM successfully classifies satellite images with accuracy comparable to the classical SVM with the radial basis function kernel for large datasets. Interestingly, for large datasets, the high accuracy was also observed for the simple quantum kernels, lacking quantum entanglement.

Index Terms—Cloud detection, kernel methods, quantum machine learning (QML), remote sensing.

Manuscript received 31 January 2023; revised 12 May 2023; accepted 26 July 2023. Date of publication 10 August 2023; date of current version 24 August 2023. This work was supported in part by the European Space Agency (ESA) and in part by the ESA Φ -lab AI-enhanced Quantum Computing for Earth Observation (QC4EO) initiative (<https://philab.esa.int/>), under ESA contract 4000137725/22/NL/GLC/my. The works of J. Mielczarek, G. Czelusta, and F. Szczepanek, were supported by the Priority Research Area Digworld under the program Excellence Initiative - Research University at the Jagiellonian University in Kraków. The work of J. Nalepa was supported by the Silesian University of Technology grant for maintaining and developing research potential. (Corresponding authors: Artur Miroszewski; Jakub Mielczarek; Jakub Nalepa.)

Artur Miroszewski, Jakub Mielczarek, Grzegorz Czelusta, and Filip Szczepanek are with the Institute of Theoretical Physics, Jagiellonian University, 30-348 Cracow, Poland (e-mail: artur.miroszewski@uj.edu.pl; jakub.mielczarek@uj.edu.pl; grzegorz.czelusta@doctoral.uj.edu.pl; filip.szczepanek@student.uj.edu.pl).

Bartosz Grabowski is with the KP Labs, 44-100 Gliwice, Poland (e-mail: bgrabowski@kplabs.pl).

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

Jakub Nalepa is with the KP Labs, 44-100 Gliwice, Poland, and also with the Department of Algorithmics and Software, Silesian University of Technology, 44-100 Gliwice, Poland (e-mail: jnalepa@ieee.org).

Digital Object Identifier 10.1109/JSTARS.2023.3304122

I. INTRODUCTION

SATELLITE imaging plays an increasingly important role in various aspects of human activity. The spectrum of applications ranges from cartographic purposes [1], [2] through meteorology [3], ecology, and agronomy [4] to security and urban monitoring [5]. Consequently, dozens of terabytes of raw imaging data are generated daily from satellite constellations, such as those built within the European Copernicus Programme. The large volume of multi- or hyperspectral images, which capture the detailed characteristics of the scanned materials, makes them difficult to transfer, store, and ultimately analyze. Therefore, their reduction through the extraction of useful information is a critical issue in real-world applications. An important step in the data analysis chain of optical satellite data is the identification of clouds. The interest is twofold: on the one hand, such cloudy regions can be removed from further processing, as the objects of interest are likely to be obscured. On the other hand, efficient detection of cloud cover on the Earth's surface is important in meteorological and climate research [6]. Since the reduction is performed on a huge amount of raw data, the efficiency of this process is a key factor in practice. Therefore, it is reasonable to search for new methods to analyze such huge datasets, improving image data classification into clear and cloudy areas.

One of the most promising directions is the application of quantum algorithms executed on quantum computers. Quantum computing methods hold the potential to improve significantly both the time and accuracy performance of various data analysis tasks. For example, it has been theoretically proven that solving the linear set of equations can be exponentially sped up compared to the best classical algorithms [7]—This task is known to play a significant role in the data analysis. Moreover, quantum algorithms have shown promise in improving the accuracy of data analysis tasks, such as classification problems [8]. In this context, the emerging field of quantum machine learning (QML) [9] may play a crucial role in further advancing the development of quantum computing and its applications in data analysis. The field of QML encompasses a range of approaches that intersect quantum computing with machine learning (ML). Among the most studied methods in QML are the quantum extensions of the support vector machine (SVM) algorithms [10], quantum-kernel methods [11], variational quantum classifiers [8], quantum convolutional neural networks [12], quantum convolutional neural networks [13], and quantum principal component analysis (PCA) [14].

These methods utilize quantum computing to develop more powerful and efficient algorithms for various ML tasks.

In this article, we follow this research pathway and propose to exploit quantum-kernel SVMs to tackle a fundamental task of cloud detection in satellite images, being the “hello, world” problem in Earth observation, which has been already widely researched in [6], [15], and [16]. Yet, building accurate and efficient cloud detection algorithms is of paramount practical importance, as it can help significantly reduce the amount of image data for on-board analysis or downlink by pruning cloudy areas, which may not convey any useful information to the end users, hence it can be interpreted as an on-board smart compression technique [17]. Ultimately, being able to show the usefulness of QML in such a fundamental issue would indicate the importance of such techniques for the whole field of remote sensing, and secondly, pave a way for other, more specialized applications of QML in the field.

A. Contribution

In this article, we investigate the possibility of using QML algorithms in the field of satellite data analysis. As the field of QML is yet much less developed than classical ML, we turn our attention to one of the most fundamental but crucial problems in space-borne remote sensing—cloud detection. The cloud detection problem can be addressed with the most explored QML approaches. As already mentioned, the problem is not only of academic interest but plays an essential role in satellite data reduction. Moreover, not only is cloud detection of great importance in satellite data reduction, but it also serves as a valuable benchmark for quantum computing methods due to the availability of numerous high-quality datasets. Even though it is still in its infancy, the potential of quantum computations might be a game changer for such applications (see [18], [19], [20], and [21]).

Specifically, we compare the classification performance of a classical SVM and its quantum extensions employing quantum kernels.¹ The approach of quantum extensions of SVMs taken in this article belongs to the classical-quantum class [22] of ML algorithms. In this class, algorithms rely on the advantages of quantum computation in order to improve classical ML methods. The specific realization of the classical-quantum routine is called an *implicit* quantum classifier [8], [23]. There are theoretical arguments [8], [24], [25] that some relevant quantum kernels are hard to evaluate on a classical computer. Therefore, if they provide an advantage in classification accuracy, this would advocate a strong use case for quantum computing methods. In this article, a family of quantum kernels has been selected such that both the role of quantum entanglement can be investigated and the quantum kernels can be studied for complex datasets.

Additionally, to get a deeper understanding of the quantum-kernel methods and show their usefulness in practice, it is pivotal

¹Note that the quantum SVM (QSVM)[10] algorithm has theoretically been proven to exhibit exponential speedup over the classical SVM. However, the full application of the algorithm requires many qubits, being of the order of the size of the training set. This is not the case for the hybrid SVM approach (with the quantum kernel) considered here.

to focus on widely adopted image data corresponding to real use cases. Thus, we tackle the cloud detection task in satellite image data, which is one of the most important processing steps for such imagery. Our experimental study was performed over the benchmark multispectral image data acquired by the Landsat-8 satellite revealed that SVMs with quantum kernels offer a classification accuracy at least comparable to classic radial basis function (RBF) kernel SVMs.

B. Structure of This Article

The rest of this article is structured as follows. In Section II-A, we discuss the theory behind SVMs, quantum-kernel methods, and kernel-target alignment. The proposed hybrid SVMs are presented in Section III. In Section IV, we report and discuss the results of our experimental study. Finally, Section V concludes this article.

II. THEORY

This section provides a gentle introduction to SVMs (see Section II-A). Additionally, we present the background behind the quantum-kernel methods (see Section II-B) and kernel-target alignment (see Section II-C)—These concepts are exploited in our hybrid SVMs for multispectral satellite data analysis.

A. Support Vector Machines (SVMs)

In binary classification, we assign one of two labels, conventionally $\{-1, 1\}$, to each datum in a set based on its features. Considering the data in terms of points occupying a feature space, the problem can be thought of as dividing the said feature space so that each of its two parts contains only one class of data points. There are a plethora of supervised ML classifiers for this task, with SVMs being one of the most widely used and well-established in the field, already exploited in an array of pattern recognition and classification tasks [26]. In SVMs, based on training data, a hyperplane is found, defined by its normal vector w and offset b , such that for any training datum x_i and its label y_i , we have

$$y_i(w \cdot x_i - b) \geq 1. \quad (1)$$

In order to decrease the risk of new data being misclassified, one also aims to maximize the margin $2/\|w\|^2$, that is, the distance between the two-class vectors ± 1 . Having found a separating hyperplane, it can be observed that it is defined by a (usually a very small) subset of training vectors, called the *support vectors*, satisfying either $wx - b = 1$ or $wx - b = -1$.

The aforementioned formulation leads to a *hard-margin* SVM, disallowing for any points to fall inside the margin. This makes it impossible to train the classifier on linearly nonseparable data. However, a *soft-margin* SVM can be introduced by allowing each datum x_i to deviate by ξ_i from satisfying the conditions in (1), obtaining a new set of conditions

$$y_i(wx_i + b) \geq 1 - \xi_i. \quad (2)$$

For an N -element training set, the optimization problem at which one arrives is in its dual form given by

$$\begin{aligned} \text{maximize: } & \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i,j=1}^N \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle \\ \text{subject to: } & \sum_i y_i \alpha_i = 0, \quad 0 \leq \alpha_i \leq C \end{aligned} \quad (3)$$

where $C \geq 0$ is the regularization parameter that specifies the impact of values ξ_i on the cost function. Then, the decision function for classifying new data x takes the form of

$$f(x) = \text{sgn} \left(\sum_{i=0}^N y_i \alpha_i \langle x, x_i \rangle + b \right). \quad (4)$$

Observe that both training and test phases do not depend directly on the data points x_i , but on the overlap between points calculated with inner product. If we introduced a different similarity measure between points, the procedure would not change. Therefore, an SVM lends itself to the use of the *kernel trick*. With a nonlinear transformation ϕ chosen, any potential occurrences of $\langle \phi(x_i), \phi(x_j) \rangle$, the inner product of two data points in a higher dimensional space, can instead be replaced with the value of a kernel function $k(x_i, x_j)$. This leads to the objective of the optimization problem, being

$$\sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i,j=1}^N \alpha_i \alpha_j y_i y_j k(x_i, x_j) \quad (5)$$

and the decision function

$$f(x) = \text{sgn} \left(\sum_{i=0}^N y_i \alpha_i k(x, x_i) + b \right) \quad (6)$$

for a specified kernel function k .

Although there are numerous kernels already deployed in SVMs in various applications [26], [27], the RBF kernel is particularly widely used in SVMs [28]. The similarity measure for this kernel is given as

$$k(x_i, x_j) = e^{-\gamma \|x_i - x_j\|^2}. \quad (7)$$

The RBF kernel is known for its extremely high flexibility (its Vapnik–Chervonenkis [29], [30] dimension is infinite) and good generalization properties. Additionally, the RBF kernel is convenient to fine tune, as it has only one parameter (the width of the kernel γ), which is commonly optimized together with the regularization parameter C . This is particularly important due to the high time and memory complexity of the SVM training, depending on the training set size. Hence, grid searching a large solution space may easily become infeasible to optimize the kernel hyperparameters. There are, however, fast approaches toward optimizing the training sets, kernel parameters, and subsets of feature sets for SVMs, which effectively exploit heuristic techniques to accelerate this process [26].

B. Quantum-Kernel Methods

The central motivation for utilizing quantum computational methods in SVM kernels is to take advantage of the exponentially large target space \mathcal{H} . This can lead to better separability of the data. When considering the implementation of quantum-kernel methods, a principal question that quickly arises pertains to the way in which classical input data will be loaded into the quantum circuit. In general, the objective will be to construct a unitary operator for each input datum x , such that applying it to the initial quantum zero state leaves us with a specified representation of x . Considering an example of a 1-qubit quantum circuit

$$U_{\phi(x)}|0\rangle = |\phi(x)\rangle. \quad (8)$$

This process is called *quantum embedding*, while such transformation $U_{\phi(x)}$ induces *quantum feature map* $|\phi(x)\rangle$. In performing quantum embedding of a classical datum on n qubits, we effectively map it into a 2^n -dimensional Hilbert space

$$U_{\phi(x)}|0\rangle^{\otimes n} = |\phi(x)\rangle \in (\mathbb{C}^2)^{\otimes n} = \mathcal{H}. \quad (9)$$

Some simpler, dimension-preserving examples of concrete feature maps may be recalled. One such method, which is referred to as the amplitude embedding, results in the quantum state with probability amplitudes corresponding to the components of the normalized input data vector. Let $x \in \mathbb{R}^n$, then its quantum-embedded form will become

$$|\psi\rangle = \frac{1}{\|x\|} \sum_{i=1}^{2^n} x_i |i\rangle \quad (10)$$

where $|i\rangle$ is the i th Z-basis state. Another noteworthy approach is basis embedding, which, in turn, considerably increases the dimension of the data, resulting in a state that is not in superposition. It builds on intuition brought by the analogy between classical binary sequences and corresponding z-basis states

$$\begin{aligned} x &\mapsto |b_1 \dots b_n\rangle \\ x &= [b_1 \dots b_n]^T, \quad b_i \in \{0, 1\}. \end{aligned} \quad (11)$$

However, it is the ability to directly operate on complex high-dimensional data stored in qubits that makes quantum computing promising in the realm of data classification. Therefore, commonly used feature maps aim to increase the dimensionality of input data while also exploring the possibilities provided by quantum entanglement and superposition. Such methods of quantum embedding are introduced and discussed in Section III-A.

Considering a collection of quantum states obtained by means of applying a feature map to different classical input data, it is straightforward to reason about them in terms of kernel methods. Kernel K in regards to any two embedded classical data x_1, x_2 can be defined as the fidelity between the resulting quantum states

$$K(x_i, x_j) = |\langle \phi(x_i) | \phi(x_j) \rangle|^2. \quad (12)$$

Such kernel K is known as a *quantum kernel*, or a *quantum embedding kernel*.

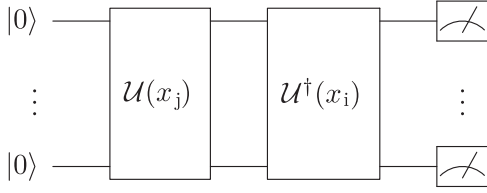


Fig. 1. Quantum circuit for estimating $\langle U(x_i)|U(x_j)\rangle$ [see (14)], with x_i, x_j embedded using an n -qubit U operator, with the use of the circuit inversion method.

Taking into account that for any quantum state $|\psi\rangle \in \mathbb{C}^{2^n}$, only $\langle 0^n|\psi\rangle$ could trivially be estimated with the use of Z-basis measurement, a method for realizing the estimation presented in (12) needs to be selected. A well-known approach would be to employ the *swap test*, which can further be extended to allow fidelity estimation of two n -qubit states [31]. However, this comes with the requirement of having $3n$ qubits available: n qubits for each of the quantum states being compared and n ancilla qubits. In a similar vein, a modification of the *Hadamard test* can be made by extending the circuit with an n -qubit register and preceding the controlled application of $U(x_i)$ with the application of $U(x_j)$ to the new register, conditioned on the ancilla qubit being $|0\rangle$. For such a circuit, the fidelity can be derived from measuring the final state of the ancilla qubit

$$p(|0\rangle) = \frac{2 + 2 \operatorname{Re} \langle U(x_j)|U(x_i)\rangle}{4}. \quad (13)$$

This approach reduces the number of required qubits to $2n + 1$ but, in turn, requires us to be able to construct the controlled version of $U(x)$, the unitary that embeds the classical datum x into n qubits.

Finally, if the state $|\psi\rangle$ is the result of applying $U(x_i)|0^n\rangle$, not unlike the ones in Havlicek's formulation of a hybrid SVM [8] employed in this work, the fidelity between two states $|U(x_i)\rangle, |U(x_j)\rangle$ can be simply derived by concatenating to the existing circuit the Hermitian conjugate of the transformation $U(x_j)$ and performing Z-basis measurement on all qubits (see Fig. 1), yielding

$$\langle U(x_j)|U(x_i)\rangle = P(|0^n\rangle) = \langle 0^n|U^\dagger(x_j)U(x_i)|0^n\rangle. \quad (14)$$

C. Kernel-Target Alignment

Using blindly a huge size of the target Hilbert space \mathcal{H} in order to rigidly embed the classical data can backfire on the resulting classification performance. First, one can easily overfit the model leading to the poor generalization performance. Second, in the high-dimensional space, almost all vectors are orthogonal to each other, causing the vanishing of fidelity based kernels [see (12)] and possible untrainability of the learning models [32]. Therefore, one should look for a tradeoff between the size of the quantum embedded feature space and the aforementioned obstructions. It has been proposed [32], [33] to introduce additional, variational hyperparameters to the quantum feature maps to calibrate them for the specific learning task. Those hyperparameters can be chosen by maximizing a

function called kernel-target alignment, which we introduce in the following.

For a given set of data $\{x_1, \dots, x_N\}$, a kernel function can be represented through a Gram matrix

$$K(x_i, x_j) = K_{ij}. \quad (15)$$

Each entry in the aforementioned matrix indicates the mutual similarity for the data points x_i and x_j . Consider a kernel function

$$\bar{K}(x_i, x_j) = \begin{cases} +1, & \text{if } x_i \text{ and } x_j \text{ are in the same class} \\ -1, & \text{if } x_i \text{ and } x_j \text{ are in different classes.} \end{cases}$$

It shows a clear distinction between classes of data points. If one could construct a feature map that gives rise to the aforementioned kernel function, then one would obtain the perfect SVM performance. Therefore, \bar{K} is called the *ideal* kernel. As SVMs are supervised learning models, for a given training data, one can use data point labels to construct the *ideal* kernel matrix

$$\bar{K}_{ij} = y_i y_j \quad (16)$$

where $y_i, y_j \in \{+1, -1\}$ are the labels of the data points x_i, x_j . In general, in almost every situation, one will not be able to find the exact feature map, which gives rise to the ideal kernel. Therefore, parameterized families of feature maps are used to optimize the resulting kernel matrix in such a way that it resembles the ideal kernel as closely as possible.

To compare two kernel matrices, one can use the matrix alignment given as

$$\mathcal{A}(K_1, K_2) = \frac{\langle K_1, K_2 \rangle_F}{\sqrt{\langle K_1, K_1 \rangle_F \langle K_2, K_2 \rangle_F}} \quad (17)$$

where $\langle K_1, K_2 \rangle_F = \operatorname{Tr}\{K_1^T K_2\}$ is a Frobenius inner product. One can utilize the matrix alignment \mathcal{A} to create a smooth function of kernel function parameters, which measures the similarity between the specific and ideal kernel matrices. It is called kernel-target alignment

$$\mathcal{T}(K) = \mathcal{A}(\bar{K}, K) = \frac{\sum_{ij} y_i y_j K_{ij}}{\sqrt{(\sum_{ij} K_{ij}^2) (\sum_{ij} y_i^2 y_j^2)}}. \quad (18)$$

As expected, the kernel-target alignment correlates with the performance of the classifier [34], [35], and it is commonly used in the model selection process. Denoting kernel families obtained from the parameterized feature maps as $K(\theta)$, where θ is a hyperparameter (or a set of hyperparameters), we can express the kernel optimization task as

$$\max_{\theta} \mathcal{T}(K(\theta)). \quad (19)$$

III. METHODS

This section presents the introduced hybrid models, which are exploited for multispectral data analysis (see Section III-A). Additionally, we discuss our approach for reducing the SVM training sets in Section III-B—This step is pivotal to enable us to train SVMs from massively large Earth observation data.

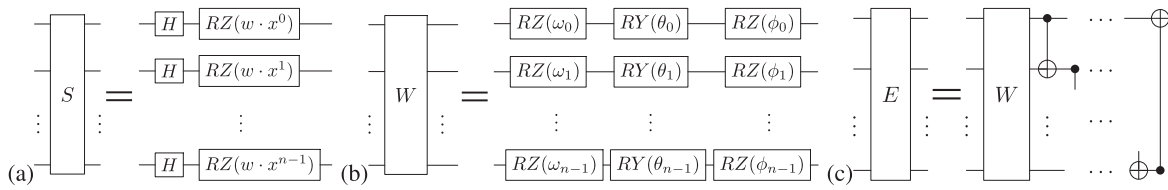


Fig. 2. Layers with which we construct circuit architectures. (a) Data encoding layer S . The data rescaling parameter is set fixed $w = \pi$ throughout this article. (b) Variational layer W . During the kernel-target alignment optimization phase, $3 \cdot n$ independent parameters ($\omega_0, \dots, \omega_{n-1}; \theta_0, \dots, \theta_{n-1}; \phi_0, \dots, \phi_{n-1}$) are varied for each W layer. (c) Entangling variational layer E . It consists of W variational layer followed with entangling gates. Therefore, it inherits $3 \cdot n$ independent variational parameters from W layer. For discussion of the layers, refer to Section III-A.

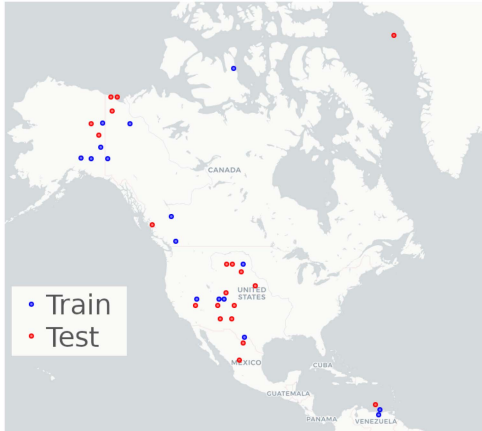


Fig. 3. Location of each 38-Clouds dataset training and test scene.

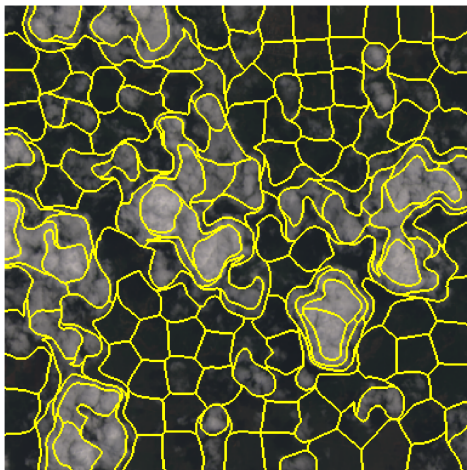


Fig. 4. SLIC superpixel segmentation applied to the 38-Cloud training patch. Yellow lines indicate the borders of the segment.

A. Hybrid Models

In this work, we introduce the circuit architectures for kernel estimation in the cloud classification task. These circuits are designed with three types of gate layers (see Fig. 2).

- 1) *Data encoding layer S* [see Fig. 2(a)]—The rotations by the w -rescaled value of the specific feature performed on the corresponding qubits. We keep the same scaling factor for each feature $w = \pi$. The initial layer of Hadamard gates is introduced in order to utilize the superposition of states by abandoning the computational Z-basis. With

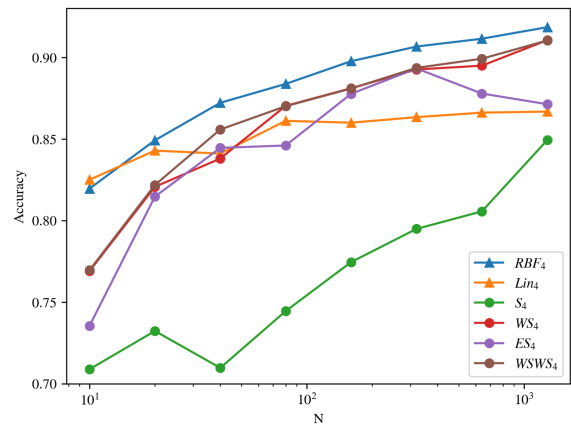


Fig. 5. Overall test accuracy (Acc) of learning models for different training sample sizes N . The classical SVM models are shown with triangular markers, while hybrid models are shown with circular markers.

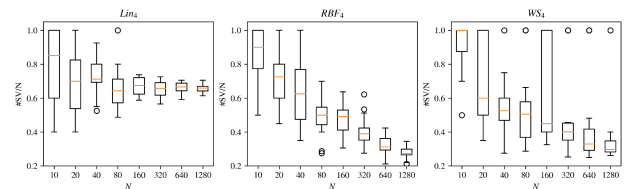


Fig. 6. Box plots of the ratio of the number of support vectors ($\#SV$) and the size of the training set (N) obtained for the Lin_4 , RBF_4 , and WS_4 SVM models over all 20 independent executions for each N . Each rectangular box shows IQR of data points, with median indicated by the orange horizontal bar. The whiskers extend to the extreme data points within 1.5 IQR from the rectangular box. The outliers are marked with circles.

such a map, each feature is encoded into different quantum register, therefore, the number of features ultimately equals the number of qubit registers $n = m$.

- 2) *Variational layer W* [see Fig. 2(b)]—The parameterized arbitrary rotations of each qubit. Each W layer introduces $3 \cdot n$ variational hyperparameters. The layer is implemented by broadcasting PennyLane's Rot operation. Both S and W layers consist of one-qubit gates, hence, they do not introduce entanglement to the system. There is a perfect separation of the qubit registers in the circuit.
- 3) *Entangling variational layer E* [see Fig. 2(c)]—The W layer followed by strong entangling of qubits done by CNOT two-qubit gates. E layer inherits $3 \cdot n$ independent variational parameters from the W layer used in its definition. The layer is implemented by the PennyLane's StronglyEntanglingLayers operation.

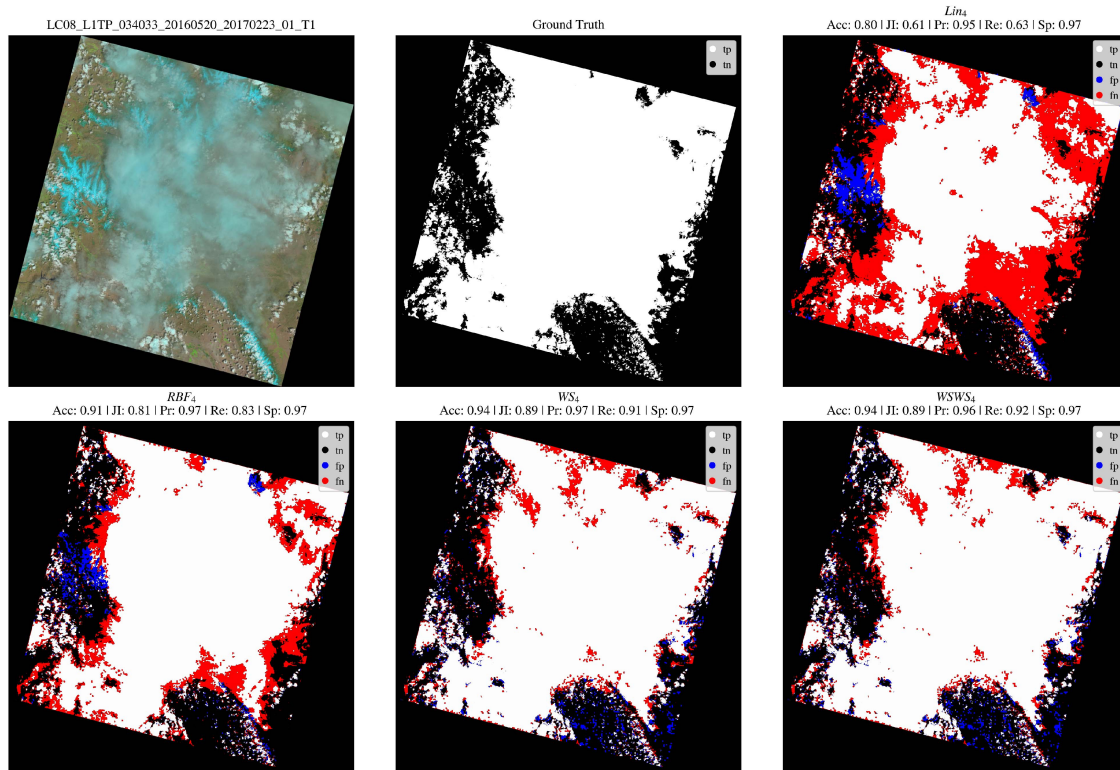


Fig. 7. Visualization of predictions of different models (Lin₄, RBF₄, WS₄, WWS₄), alongside the quantitative metrics. All models were trained on one of the training samples of size $N = 1280$ (the same for all classifiers), together with the natural false color scene image (here: LC08_L1TP_034033_20160520_20170223_01_T1), and the ground truth corresponding to this scene.

Our circuit architectures are recognized by their layer composition. By using the WS_n symbols, we mean that the data embedding map first transforms the initial state $|0\rangle^{\otimes n}$ by the arbitrary rotations layer W , then the data are encoded with the S layer. To estimate a quantum-kernel entry, the conjugate embedding map with respect to a different data point is concatenated to the WS_n circuit, as explained in Section II-B. Other architecture symbols follow the same rule.

In this study, we investigate the S , WS , ES , and WWS circuit architectures. This choice enables us to analyze the significance of hyperparameter tuning, superposition, entanglement, and expressivity in quantum feature maps, while being able to simulate them with classical computers on a real-life dataset. It is worth emphasizing that for the S and W layers-based maps, the quantum-kernel complexity is expected to be low. Such circuits are, therefore, easy to simulate on classical computers, and the application of quantum computers does not provide an advantage here. However, precisely thanks to this property, we were able to perform studies for real-world large-scale datasets, which would be much more difficult to do in the case of more complex quantum kernels. This especially concerns the ZZ map discussed in [8] and [24], for which the computations are $\#P$ —hard for classical computers. The case of the ZZ map was beyond the reach of our computational abilities for the complex dataset under investigation. However, we managed to analyze intermediate complexity kernels involving the entangling E layer. Hence, in this article, we focus on applying the quantum-kernel methods

to huge amounts of real Earth observation data captured in orbit. However, this was achieved by the cost of reducing the kernels' complexity.

B. Training Data Reduction

SVMs suffer from their high time and memory training complexity, which depend on the size of the training set. The 38-Clouds training data consists of approximately 1.24 billion of pixels (the dataset is presented in more detail in Section IV-A), the use of all of them is implausible as this size significantly exceeds the computational capabilities of modern computers in the context of the SVM training. Since only a subset of all training vectors is annotated as support vectors during the process of SVM training, we can effectively exploit only a subset of the most important examples or create the *prototype vectors*, which are a good representation of similar examples [26] (e.g., combining the information captured by several neighboring vectors in the feature space). In this work, we follow the later approach by utilizing the superpixel segmentation techniques [36]. Here, we create coherent pixel groupings by considering similarity measures defined using perceptual features—We build upon the famous simple linear iterative clustering (SLIC) [37], which performs the segmentation based on color and proximity distance (see an example result of SLIC obtained for the 38-Cloud data sample rendered in Fig. 4). For each multispectral training patch, we do the following steps.

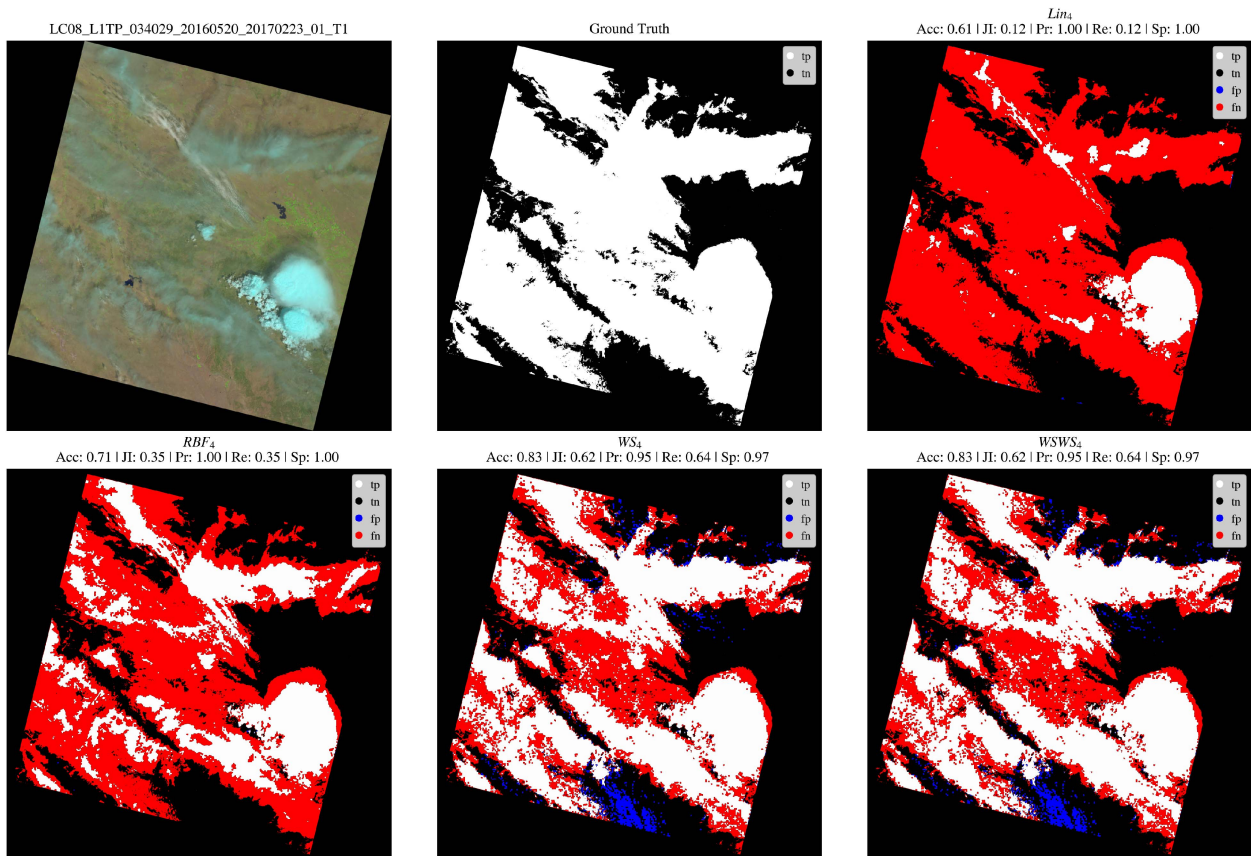


Fig. 8. Visualization of predictions of different models (Lin_4 , RBF_4 , WS_4 , $WSWS_4$), alongside the quantitative metrics. All models were trained on one of the training samples of size $N = 1280$ (the same for all classifiers), together with the natural false color scene image (here: $LC08_L1TP_034029_20160520_20170223_01_T1$), and the ground truth corresponding to this scene.

- 1) Perform SLIC ($N_{\text{segments}} = 200$, smoothing kernel $\sigma = 5$; the hyperparameters of SLIC were fine-tuned experimentally, in order to compromise between the reduction rate and the spatial representativeness of the resulting training examples) segmentation.
- 2) Remove margin pixels from each segment.
- 3) Create a prototype training vector example from each segment by computing the following statistical measures for each spectral band: mean, median, interquartile range (IQR), min, max, and standard deviation.
- 4) Label the created superpixel with the majority label of the pixels contained within the corresponding superpixel.

After data reduction, we obtained approximately 0.93 million of training vectors (resulting in the massive reduction rate of more than $1300\times$) consisting of 24 features and a ground-truth label. Additionally, almost 92% of superpixels have their class label decided by at least 80–20 vote ratio.

IV. EXPERIMENTAL VALIDATION

In this section, we discuss our experimental study focused on understanding the abilities of hybrid SVMs in the context of cloud detection in satellite multispectral data. The exploited dataset is discussed in detail in Section IV-A, whereas the

experimental methodology is highlighted in Section IV-B. The results are presented and discussed in Section IV-C.

A. Dataset

We utilize satellite multispectral image data contained in the 38-Cloud dataset [38], [39]. It consists of 18 training and 20 test scene images captured by the Landsat-8 satellite (30-m ground sampling distance) over the continent of America (see Fig. 3). Scenes cover a wide range of climate zones and terrain types, including deserts, forests, meadows, mountains, agriculture, urban areas, coastlines, snow, and ice. With each scene, we are provided the ground truth for cloud binary classification. There is no gradation in the cloud labels; hence, this class includes both thick cumulus, partly transparent cirrus clouds as well as thin haze. For convenience, scene images are cropped into 8400 (training) and 9201 (test) 384×384 pixel patches by the authors of the dataset. Each pixel has five values associated with it: intensity values in four spectral bands (blue: 450–515 nm, green: 520–600 nm, red: 630–680 nm, and NIR: 845–885 nm) and a ground-truth label (cloud or background). It is worth noting that the scene images are not rotated to fit the standard rectangular image format, therefore, they include a significant amount of margin pixels, represented by $[0,0,0,0]$ vectors with the noncloud (background) class label assigned.

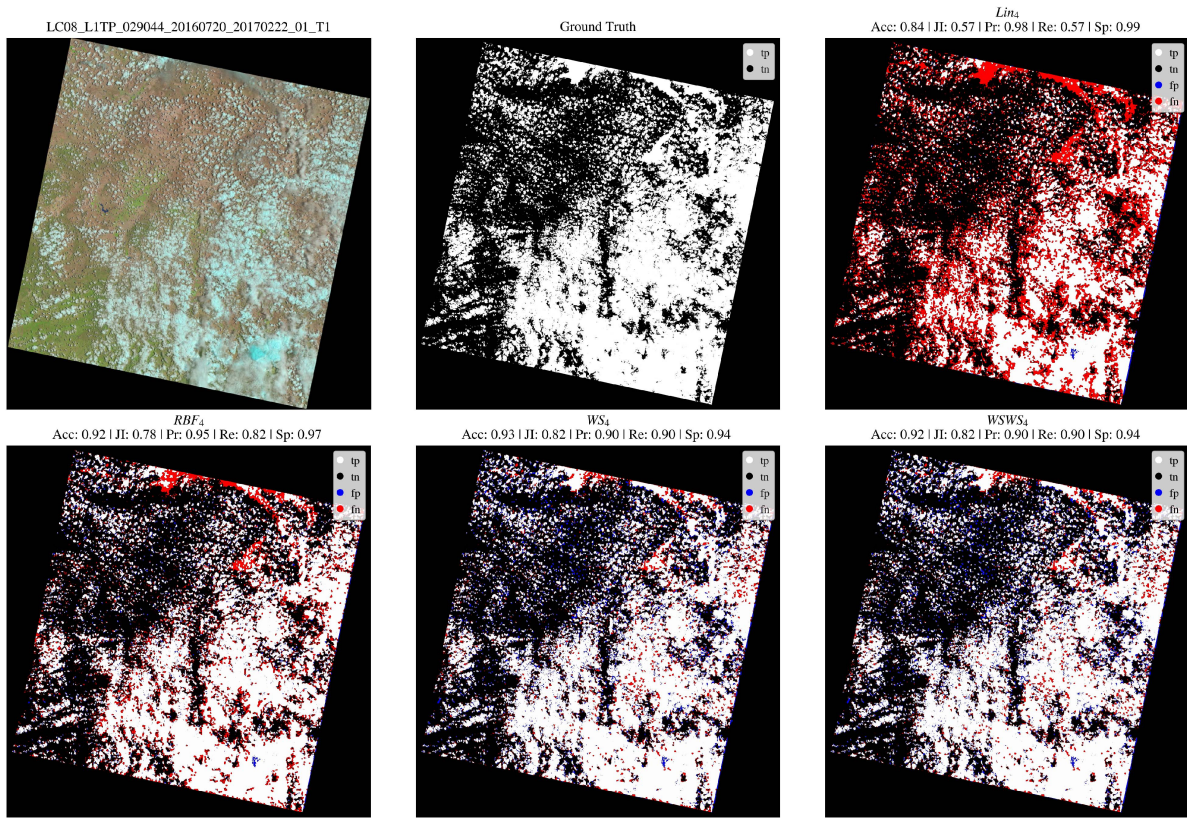


Fig. 9. Visualization of predictions of different models (Lin_4 , RBF_4 , WS_4 , WSW_4), alongside the quantitative metrics. All models were trained on one of the training samples of size $N = 1280$ (the same for all classifiers), together with the natural false color scene image (here: LC08_L1TP_029044_20160720_20170222_01_T1), and the ground truth corresponding to this scene.

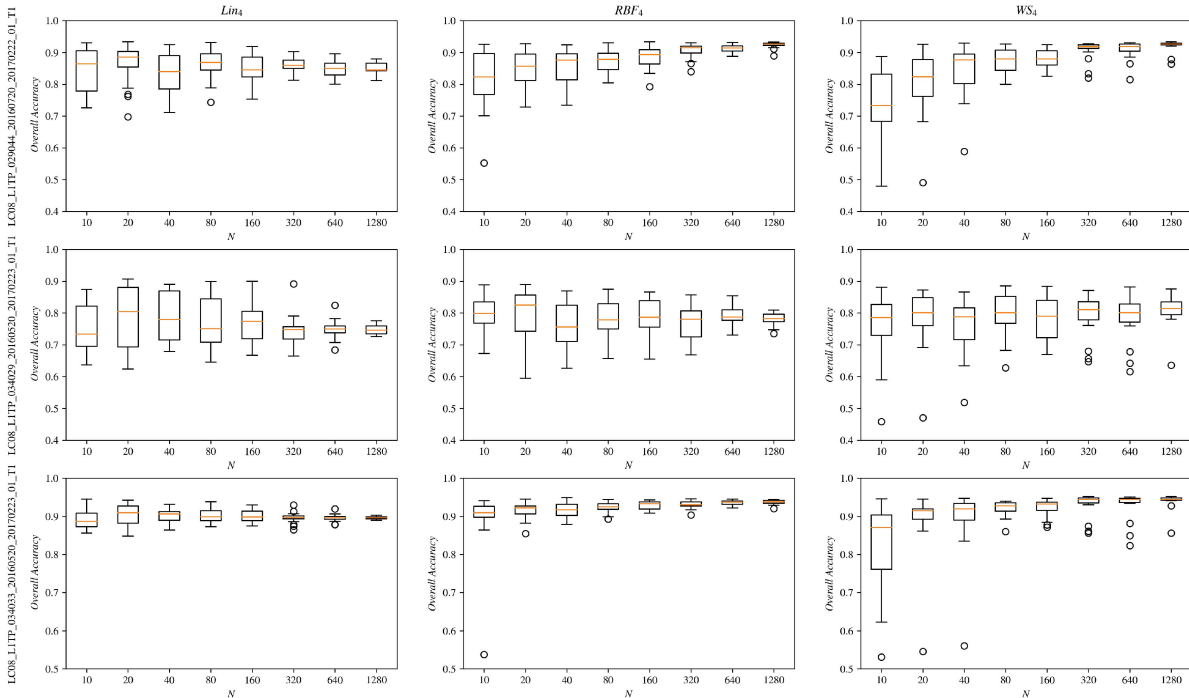


Fig. 10. Box plots showing Acc obtained within 20 independent executions (for 20 reduced training sets of the N size) for the three scenes shown in Figs. 7–9. Each rectangular box shows IQR of data points, with median indicated by the orange horizontal bar. The whiskers extend to the extreme data points within 1.5 IQR from the rectangular box. The outliers are marked with circles.

TABLE I
RESULTS FOR RBF₄ CLASSICAL MODEL, AS WELL AS FOR THE WS_4 AND $WSWS_4$ HYBRID MODELS

| N | Acc | \mathcal{J} | Pr | Re | Spec |
|----------------------------|----------------------|----------------------|----------------------|----------------------|----------------------|
| The RBF ₄ model | | | | | |
| 10 | 0.820 (0.138) | 0.515 (0.123) | 0.631 (0.132) | 0.744 (0.177) | 0.841 (0.201) |
| 20 | 0.849 (0.084) | 0.525 (0.105) | 0.652 (0.089) | 0.721 (0.163) | 0.889 (0.086) |
| 40 | 0.872 (0.067) | 0.550 (0.084) | 0.680 (0.088) | 0.712 (0.130) | 0.913 (0.076) |
| 80 | 0.884 (0.054) | 0.576 (0.075) | 0.685 (0.080) | 0.754 (0.075) | 0.919 (0.061) |
| 160 | 0.898 (0.036) | 0.581 (0.068) | 0.701 (0.063) | 0.743 (0.080) | 0.939 (0.036) |
| 320 | 0.907 (0.026) | 0.606 (0.058) | 0.729 (0.059) | 0.757 (0.060) | 0.948 (0.029) |
| 640 | 0.911 (0.020) | 0.609 (0.047) | 0.732 (0.048) | 0.763 (0.040) | 0.954 (0.019) |
| 1280 | 0.919 (0.010) | 0.628 (0.035) | 0.757 (0.036) | 0.763 (0.029) | 0.963 (0.009) |
| The WS_4 hybrid model | | | | | |
| 10 | 0.769 (0.156) | 0.446 (0.157) | 0.599 (0.131) | 0.684 (0.245) | 0.810 (0.213) |
| 20 | 0.821 (0.128) | 0.503 (0.119) | 0.621 (0.125) | 0.717 (0.178) | 0.847 (0.191) |
| 40 | 0.838 (0.128) | 0.530 (0.112) | 0.642 (0.130) | 0.737 (0.142) | 0.858 (0.196) |
| 80 | 0.870 (0.066) | 0.554 (0.091) | 0.675 (0.094) | 0.737 (0.119) | 0.905 (0.070) |
| 160 | 0.881 (0.057) | 0.553 (0.096) | 0.708 (0.097) | 0.703 (0.135) | 0.929 (0.055) |
| 320 | 0.893 (0.044) | 0.568 (0.092) | 0.709 (0.088) | 0.731 (0.122) | 0.935 (0.036) |
| 640 | 0.895 (0.053) | 0.573 (0.100) | 0.717 (0.087) | 0.731 (0.129) | 0.940 (0.040) |
| 1280 | 0.911 (0.031) | 0.602 (0.064) | 0.723 (0.061) | 0.757 (0.076) | 0.949 (0.026) |
| The $WSWS_4$ hybrid model | | | | | |
| 10 | 0.770 (0.156) | 0.445 (0.159) | 0.600 (0.132) | 0.683 (0.247) | 0.811 (0.213) |
| 20 | 0.822 (0.127) | 0.505 (0.117) | 0.622 (0.124) | 0.718 (0.175) | 0.847 (0.191) |
| 40 | 0.856 (0.082) | 0.541 (0.095) | 0.654 (0.089) | 0.737 (0.144) | 0.888 (0.088) |
| 80 | 0.870 (0.067) | 0.554 (0.092) | 0.674 (0.093) | 0.737 (0.120) | 0.904 (0.071) |
| 160 | 0.881 (0.059) | 0.552 (0.096) | 0.710 (0.096) | 0.703 (0.140) | 0.930 (0.058) |
| 320 | 0.894 (0.044) | 0.575 (0.082) | 0.698 (0.081) | 0.744 (0.104) | 0.931 (0.038) |
| 640 | 0.899 (0.048) | 0.582 (0.089) | 0.714 (0.081) | 0.745 (0.114) | 0.940 (0.038) |
| 1280 | 0.910 (0.031) | 0.602 (0.064) | 0.723 (0.062) | 0.758 (0.076) | 0.949 (0.026) |

The best results are boldfaced for each model—We report the average (standard deviation) of the corresponding metric obtained across 20 independent executions for each size of the refined training set.

B. Experiment Methodology

We investigate the performance of classical and hybrid learning models based on SVMs. The classical SVMs are trained and tested for the RBF and linear kernel. Hybrid, classical-quantum models consist of two parts, the quantum-kernel estimation and the classical SVM routine. The evaluated models are trained on partly random balanced samples from the reduced training set obtained by the procedure described in Section III-B. The training data sample of size N is obtained by randomly selecting $\frac{N}{2}$ superpixels, which have a *cloud* label and $(\frac{N}{2} - 1)$ superpixels corresponding to a *noncloud* label. Then, the last (N th) *noncloud* superpixel is added—it contains zeroed features and represents the “margin” superpixel. Due to the substantial number of margin pixels in each scene in the 38-Cloud dataset, not including a “margin” training example could result in the significant drop in the model’s performance. We evaluated all models on a fixed set of training samples. For each training set size $N \in \{10, 20, 40, 80, 160, 320, 640, 1280\}$, we randomly sample 20 sets (which remain unchanged across all investigated SVM models). Hence, each model is trained 160 times. The investigated hybrid approaches were implemented in the PennyLane python package, and the experiments were run with the `default.qubit` simulator on classical computers.

The dimension of the quantum Hilbert space, to which we encode the data, grows exponentially with the number of qubits n that we use in quantum feature maps. All maps that we use encompass exactly the same amount of qubits as the number of features m , hence, $n = m$. The dimension of Hilbert space needed to encode superpixels in the training dataset is 2^{24} ($= 16,777,216$). Performing large-scale simulations of such big space on modern classical computers is unfeasible. Therefore, in the case of the hybrid models, we perform feature extraction by PCA. In the vast majority of cases, extracting two (four)

principal components explains 95% (99%) of variance in the training data.

With each training set, we draw an additional balanced validation set of size $\max(N/2, 300)$. For the classical SVM with linear kernel and the hybrid S_4 model, we tuned only the hyperparameter C . It was done by testing different classifiers on a validation sample in the hyperparameter range $C \in [0.01, 147.01]$ with the step size of 3. For the classical SVM with the RBF kernel, the procedure is the same, but two hyperparameters (C, γ) were tuned (the ranges of hyperparameters stay the same). For hybrid models, we first optimized kernel function parameters in order to maximize the kernel-target alignment $\mathcal{T}(K)$ [see (18)] on the training sample. For this task, we used the Adam optimizer [40]. In the case of hybrid WS_4 and ES_4 models, we optimize the $\omega_0, \dots, \omega_3; \theta_0, \dots, \theta_3; \varphi_0, \dots, \varphi_3$ parameters present in the W and E variational layers (see Fig. 2). These 12 parameters encode the families of available kernel functions in the WS_4 and ES_4 models. In the case of $WSWS_4$ model, we optimize 24 kernel parameters ($\omega_0^0, \dots, \omega_3^0; \theta_0^0, \dots, \theta_3^0; \varphi_0^0, \dots, \varphi_3^0$ and $\omega_0^1, \dots, \omega_3^1; \theta_0^1, \dots, \theta_3^1; \varphi_0^1, \dots, \varphi_3^1$) present in two W variational layers. Once the variational layers have been fine tuned, we performed C hyperparameter tuning in the same manner as for other classifiers.

The learning models are evaluated on all 20 test scenes in the 38-Cloud dataset. It means that each model is evaluated 3200 times (20 training samples \times 8 training sample sizes \times 20 test scenes). To quantify the performance of the investigated classification models, we exploited accuracy: $\text{Acc} = (\text{TP} + \text{TN}) / (\text{TP} + \text{TN} + \text{FN} + \text{FP})$, Jaccard index: $\mathcal{J} = \text{TP} / (\text{TP} + \text{FN} + \text{FP})$, precision: $\text{Pr} = \text{TP} / (\text{TP} + \text{FP})$, recall: $\text{Re} = \text{TP} / (\text{TP} + \text{FN})$, and specificity: $\text{Spec} = \text{TN} / (\text{TN} + \text{FP})$, where TP, TN, FN, and FP denote true positives, true negatives, false positives, and false

TABLE II
ADJUSTED P-VALUES FOR FRIEDMAN'S WITH *Post Hoc* DUNN'S MULTIPLE COMPARISONS TESTS FOR THE EVALUATION METRICS ELABORATED USING THE Lin_4 , RBF_4 , WS_4 , AND $WSWS_4$ MODELS

| N | Lin_4 vs. RBF_4 | Lin_4 vs. WS_4 | Lin_4 vs. $WSWS_4$ | RBF_4 vs. WS_4 | RBF_4 vs. $WSWS_4$ | WS_4 vs. $WSWS_4$ |
|---------------|---------------------|--------------------|----------------------|--------------------|----------------------|---------------------|
| Acc | | | | | | |
| 10 | >0.9999 | 0.0087 | 0.0132 | 0.0023 | 0.0036 | >0.9999 |
| 20 | 0.8499 | 0.0087 | 0.1649 | <0.0001 | 0.0014 | >0.9999 |
| 40 | 0.0423 | 0.0858 | >0.9999 | <0.0001 | 0.3003 | 0.0087 |
| 80 | 0.0057 | 0.8499 | >0.9999 | 0.3972 | 0.0423 | >0.9999 |
| 160 | <0.0001 | 0.0607 | 0.1649 | 0.0858 | 0.0291 | >0.9999 |
| 320 | 0.0023 | 0.5185 | 0.0607 | 0.3972 | >0.9999 | >0.9999 |
| 640 | 0.0291 | >0.9999 | >0.9999 | 0.0197 | 0.6681 | >0.9999 |
| 1280 | 0.0858 | 0.1649 | 0.5185 | >0.9999 | >0.9999 | >0.9999 |
| \mathcal{J} | | | | | | |
| 10 | >0.9999 | 0.0014 | 0.1649 | 0.0423 | 0.0087 | <0.0001 |
| 20 | 0.8499 | 0.1198 | >0.9999 | 0.0009 | 0.0607 | >0.9999 |
| 40 | 0.0009 | >0.9999 | 0.3003 | 0.0002 | 0.3972 | 0.1198 |
| 80 | <0.0001 | 0.0132 | 0.0132 | 0.2240 | 0.2240 | >0.9999 |
| 160 | <0.0001 | 0.0023 | 0.0197 | 0.5185 | 0.1198 | >0.9999 |
| 320 | <0.0001 | 0.1198 | 0.0003 | 0.0607 | >0.9999 | 0.5185 |
| 640 | <0.0001 | 0.1649 | 0.0005 | 0.0036 | 0.5185 | 0.5185 |
| 1280 | <0.0001 | 0.0001 | 0.0009 | >0.9999 | >0.9999 | >0.9999 |
| Pr | | | | | | |
| 10 | 0.2240 | <0.0001 | <0.0001 | 0.0002 | 0.0057 | >0.9999 |
| 20 | >0.9999 | <0.0001 | 0.0001 | <0.0001 | 0.0009 | >0.9999 |
| 40 | >0.9999 | <0.0001 | 0.0197 | <0.0001 | 0.0197 | 0.3003 |
| 80 | >0.9999 | 0.5185 | 0.0036 | >0.9999 | 0.0423 | 0.5185 |
| 160 | >0.9999 | 0.5185 | >0.9999 | >0.9999 | >0.9999 | >0.9999 |
| 320 | >0.9999 | 0.3003 | 0.0023 | 0.6681 | 0.0087 | 0.6681 |
| 640 | >0.9999 | 0.2240 | 0.2240 | >0.9999 | >0.9999 | >0.9999 |
| 1280 | >0.9999 | 0.4713 | 0.6141 | 0.1018 | 0.1423 | >0.9999 |
| Re | | | | | | |
| 10 | 0.0087 | >0.9999 | >0.9999 | 0.1198 | 0.0057 | >0.9999 |
| 20 | 0.2240 | >0.9999 | 0.2240 | >0.9999 | >0.9999 | >0.9999 |
| 40 | 0.8499 | 0.0132 | 0.0001 | 0.6681 | 0.0291 | >0.9999 |
| 80 | 0.0003 | 0.0057 | 0.0197 | >0.9999 | >0.9999 | >0.9999 |
| 160 | <0.0001 | 0.0014 | 0.1649 | >0.9999 | 0.1649 | 0.8499 |
| 320 | <0.0001 | 0.1198 | <0.0001 | 0.2240 | >0.9999 | 0.1649 |
| 640 | <0.0001 | 0.5185 | 0.0003 | 0.0057 | >0.9999 | 0.1198 |
| 1280 | <0.0001 | 0.0023 | 0.0009 | >0.9999 | >0.9999 | >0.9999 |
| Spec | | | | | | |
| 10 | 0.2240 | <0.0001 | 0.0001 | 0.0009 | 0.1649 | 0.6681 |
| 20 | >0.9999 | <0.0001 | 0.0001 | 0.0001 | 0.0014 | >0.9999 |
| 40 | >0.9999 | <0.0001 | 0.0423 | <0.0001 | 0.0197 | 0.0423 |
| 80 | >0.9999 | 0.1649 | 0.0002 | 0.5185 | 0.0014 | 0.3003 |
| 160 | >0.9999 | 0.1198 | 0.1649 | 0.0858 | 0.1198 | >0.9999 |
| 320 | >0.9999 | 0.3003 | 0.0014 | >0.9999 | 0.0197 | 0.5185 |
| 640 | 0.5185 | 0.0057 | 0.0009 | 0.6681 | 0.2240 | >0.9999 |
| 1280 | 0.5185 | 0.0036 | 0.0014 | 0.5185 | 0.3003 | >0.9999 |

The background of the statistically significant ($p < 0.05$) results is grayed.

negatives, respectively. All results are reported for the test sets that were unseen during training (unless stated otherwise).

C. Results

The objectives of our experimental study is twofold.

- 1) To understand the impact of an increasing training set size on the generalization capabilities of both classical and hybrid SVMs.
- 2) To investigate the performance of the proposed quantum classifiers in a real-world Earth observation task of cloud detection from multispectral imagery.

In Fig. 5, we render accuracy (averaged across all independent executions for each training set size) for all models. We can observe that increasing the size of the reduced training sets leads to the consistent increase in the classification performance of all SVM models. It is of note that the rate of the performance increase started saturating for the RBF SVM model (RBF_4), whereas the quantum-kernel classifiers (WS_4 and $WSWS_4$) manifest more rapid improvements for larger N 's. This phenomenon can be further investigated in Table I, we gather all

quantitative metrics obtained using the best classical SVM with the RBF kernel (RBF_4), together with our QSVMs. Finally, in Fig. 6, we present the ratio of the number of support vectors elaborated during the training process of the underlying model (Lin_4 , RBF_4 , and WS_4). Since the inference time of SVMs depends linearly on the number of support vectors, their number should be minimized to ensure fast operation of the classifier. Although there are indeed outlying executions resulting in large numbers of SVs for the WS_4 SVMs, the overall trend in the number of SVs remains consistent for all N 's (see the median number of SVs rendered as orange lines in Fig. 6).

To verify if the differences across the investigated models are statistically important, we executed the Friedman's tests with *post hoc* Dunn's over all metrics, averaged across all independent executions for the sampled refined training sets (see Table II). We can appreciate that the WS_4 and $WSWS_4$ models, with the former being significantly less parameterized than the latter one, lead to statistically same cloud detection performance. Additionally, once the dataset is increased and reaches the size of

$N = 1280$, the quantum-kernel SVMs deliver statistically same quality measures as RBF_4 . In Figs. 7–9, we present three example 38-Cloud test scenes of varying segmentation difficulty (see different cloud characteristics). The qualitative analysis shows that the quantum-kernel SVMs can indeed outperform or work on par with well-established SVMs with the RBF kernel, and both of them significantly outperform linear-kernel classifiers in this task.

We are aware of some limitations of the hybrid SVMs. In Fig. 10, we render the box plots obtained for three test scenes visualized in Figs. 7–9. Although the aggregated metrics, averaged across 20 independent executions indicate that the WS_4 model is competitive with the classical RBF SVMs, the former classifier is slightly less stable, especially for lower N 's. However, increasing the size of the refined training set not only does allow for significantly enhance the generalization capabilities of the quantum-kernel SVMs, but it also improves their training stability. Although the RBF and quantum kernels (WS_4 , $WSWS_4$) lead to the similar classification accuracy, optimizing (on a classical machine) multihyperparameter quantum kernels is much more resource intensive. For the RBF kernel, only one kernel hyperparameter (γ) is optimized, while for WS_4 ($WSWS_4$), 12 (24) kernel hyperparameters are optimized.

The best results (overall accuracy of approximately 92–93%) of the proposed simple ML models based on superpixel segmentation and SVMs do not deviate to a large extent from the current state-of-the-art deep learning models benefiting from the fully convolutional architectures (overall accuracy of approximately 94–96% reported for the 38-Cloud test scenes [38], [39], [41]). Such large-capacity deep learning models, however, can effectively exploit the contextual information within the image during the segmentation process—This may be of paramount importance for cloud detection, as the objects of interest may manifest different shape and spectral characteristics. Thus, designing additional feature extractors [42], followed by feature selectors [43], may be pivotal to further improve the classification accuracy of hybrid SVMs—appropriate feature extraction and fusion strategies have been shown extremely important in satellite image analysis using ML techniques [44].

V. CONCLUSION

In this work, we introduced hybrid SVMs exploiting quantum kernels for the task of cloud detection in multispectral satellite images, which is the “hello, world” in remote sensing. Such quantum-kernel models, together with classical SVMs with RBF and linear kernels were thoroughly investigated in the experimental study performed over a widely used 38-Cloud dataset capturing Landsat-8 imagery. In our processing chain, the superpixel-powered training set selection is utilized to dramatically reduce the SVM training sets, and to pick the most informative training examples, together with the training prototypes, which are likely to become support vectors during the training process. Overall, we quantitatively, qualitatively, and statistically evaluated six SVM models—classical linear and RBF kernel-based SVMs, alongside the suggested hybrid SVMs based on the kernels elaborated by utilizing simulated quantum circuits called S_4 , WS_4 , ES_4 , and $WSWS_4$. The hybrid model

S_4 executed a *stiff* (no variational layers) classical data encoding into separate qubits, WS_4 introduced one variational layer, ES_4 added the entanglement between the qubit registers, while $WSWS_4$ was a straightforward extension of the WS_4 model achieved by doubling it (two encoding layers interwoven with two variational layers).

The first observation inferred from our experiments is that the *stiff* encoding S_4 model underperforms, when compared to the overall accuracy with other models (see the results rendered in Fig. 5). Being able to embed data into vectors residing in 16-D complex linear space does not necessarily increase expressivity and performance of the model—the linear kernel, defined on 4-D space surpasses the S model for all tested cases. Therefore, one needs to introduce additional parameters to the quantum feature map in order to control and tune its behavior. However, interestingly, there is no benefit in performance by introducing the entanglement via the layer E .

The results reported here constitutes an exciting point of departure for further research. Albeit the classical and hybrid SVMs offer high-quality cloud detection, they are still slightly worse than the recent advancements in large-capacity deep learning models. This can be attributed to the fact that the SVMs investigated in this work operate on a small set of features that do not capture the subtle shape and spectral characteristics of the pixels' neighborhood. We anticipate that introducing new feature extractors to our pipeline can substantially enhance the classification capabilities of the models. Our research efforts are focused on deploying quantum-kernel SVMs for other multispectral data for cloud segmentation (and segmentation of other objects of interest as well, e.g., cultivated land [45]), especially in large-scale Sentinel-2 imagery, as well as on using them for hyperspectral image classification [46], and on quantifying their robustness against noise-contaminated data [47]. Finally, we are currently investigating the nonfunctional abilities of both classical and deep ML models, with a special emphasis put on their inference time, as it is critical in processing massively large amounts of satellite imagery captured nowadays.

REFERENCES

- [1] *Copernicus Land Monitoring Service*, Jul. 12, 2022. [Online]. Available: <https://land.copernicus.eu/>
- [2] J. Castillo-Navarro, B. Le Saux, A. Boulch, N. Audebert, and S. Lefèvre, “Semi-supervised semantic segmentation in earth observation: The MiniFrance suite, dataset analysis and multi-task network study,” *Mach. Learn.*, vol. 111, no. 9, pp. 3125–3160, 2022.
- [3] D. Hanson, J. Peronto, and D. Hilderbrand, “NOAA’s Eyes in the Sky—After five decades of weather forecasting with environmental satellites, what do future satellites promise for meteorologists and society?,” *World Meteorological Org. Bull.*, vol. 62, no. 1, pp. 17–21, 2013.
- [4] J. Nalepa et al., “The hyperview challenge: Estimating soil parameters from hyperspectral images,” in *Proc. IEEE Int. Conf. Image Process.*, 2022, pp. 4268–4272.
- [5] N. Audebert et al., “Deep learning for urban remote sensing,” in *Proc. Joint Urban Remote Sens. Event*, 2017, pp. 1–4.
- [6] B. Grabowski, M. Ziąja, M. Kawulok, and J. Nalepa, “Towards robust cloud detection in satellite images using U-Nets,” in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2021, pp. 4099–4102.
- [7] A. W. Harrow, A. Hassidim, and S. Lloyd, “Quantum algorithm for linear systems of equations,” *Phys. Rev. Lett.*, vol. 103, no. 15, 2009, Art. no. 150502.
- [8] V. Havlíček et al., “Supervised learning with quantum-enhanced feature spaces,” *Nature*, vol. 567, no. 7747, pp. 209–212, 2019.

- [9] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, "Quantum machine learning," *Nature*, vol. 549, no. 7671, pp. 195–202, Sep. 2017.
- [10] P. Rebentrost, M. Mohseni, and S. Lloyd, "Quantum support vector machine for big data classification," *Phys. Rev. Lett.*, vol. 113, no. 13, Sep. 2014, Art. no. 130503.
- [11] H.-Y. Huang et al., "Power of data in quantum machine learning," *Nature Commun.*, vol. 12, no. 1, 2021, Art. no. 2631, doi: [10.1038/s41467-021-22539-9](https://doi.org/10.1038/s41467-021-22539-9).
- [12] I. Cong, S. Choi, and M. D. Lukin, "Quantum convolutional neural networks," *Nature Phys.*, vol. 15, no. 12, pp. 1273–1278, 2019.
- [13] M. Henderson, S. Shakya, S. Pradhan, and T. Cook, "Quantum convolutional neural networks: Powering image recognition with quantum circuits," *Quantum Mach. Intell.*, vol. 2, no. 1, 2019, Art. no. 2.
- [14] S. Lloyd, M. Mohseni, and P. Rebentrost, "Quantum principal component analysis," *Nature Phys.*, vol. 10, no. 9, pp. 631–633, 2014, doi: [10.1038/nphys3029](https://doi.org/10.1038/nphys3029).
- [15] F. Xie, M. Shi, Z. Shi, J. Yin, and D. Zhao, "Multilevel cloud detection in remote sensing images based on deep learning," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 10, no. 8, pp. 3631–3640, Aug. 2017.
- [16] B. Grabowski et al., "Are cloud detection U-Nets robust against in-orbit image acquisition conditions?," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2022, pp. 239–242.
- [17] G. Giuffrida et al., "The Φ -sat-1 mission: The first on-board deep neural network demonstrator for satellite earth observation," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5517414, doi: [10.1109/TGRS.2021.3125567](https://doi.org/10.1109/TGRS.2021.3125567).
- [18] 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, doi: [10.1109/LGRS.2021.3108014](https://doi.org/10.1109/LGRS.2021.3108014).
- [19] M. Henderson, J. Gallina, and M. Brett, "Methods for accelerating geospatial data processing using quantum computers," *Quantum Mach. Intell.*, vol. 3, no. 1, 2021, Art. no. 4, doi: [10.1007/s42484-020-00034-6](https://doi.org/10.1007/s42484-020-00034-6).
- [20] A. Sebastianelli, D. A. Zaidenberg, D. Spiller, B. Le Saux, and S. L. Ullo, "On circuit-based hybrid quantum neural networks for remote sensing imagery classification," 2021. [Online]. Available: <https://arxiv.org/abs/2109.09484>
- [21] M. K. Gupta, M. Romaszewski, and P. Gawron, "Potential of quantum machine learning for processing multispectral Earth observation data," *J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 1, no. 1, 2023.
- [22] V. Dunjko, J. M. Taylor, and H. J. Briegel, "Quantum-enhanced machine learning," *Phys. Rev. Lett.*, vol. 117, no. 13, 2016, Art. no. 130501.
- [23] M. Schuld and N. Killoran, "Quantum machine learning in feature Hilbert spaces," *Phys. Rev. Lett.*, vol. 122, no. 4, 2019, Art. no. 040504.
- [24] L. A. Goldberg and H. Guo, "The complexity of approximating complex-valued Ising and Tutte partition functions," *Comput. Complexity*, vol. 26, no. 4, pp. 765–833, 2017.
- [25] T. F. Demarie, Y. Ouyang, and J. F. Fitzsimons, "Classical verification of quantum circuits containing few basis changes," *Phys. Rev. A*, vol. 97, no. 4, 2018, Art. no. 042319.
- [26] J. Nalepa and M. Kawulok, "Selecting training sets for support vector machines: A review," *Artif. Intell. Rev.*, vol. 52, no. 2, pp. 857–900, 2019.
- [27] M. Tanveer, T. Rajani, R. Rastogi, Y. H. Shao, and M. A. Ganaie, "Comprehensive review on twin support vector machines," *Ann. Oper. Res.*, Mar. 2022, doi: [10.1007/s10479-022-04575-w](https://doi.org/10.1007/s10479-022-04575-w).
- [28] J. Nalepa, W. Dudzik, and M. Kawulok, "Memetic evolution of training sets with adaptive radial basis kernels for support vector machines," in *Proc. 25th Int. Conf. Pattern Recognit.*, 2021, pp. 5503–5510.
- [29] V. N. Vapnik and A. Y. Chervonenkis, "On the uniform convergence of relative frequencies of events to their probabilities," *Theory Probability Appl.*, vol. 16, no. 2, pp. 264–280, 1971.
- [30] V. Vapnik, *The Nature of Statistical Learning Theory*. Berlin, Germany: Springer, 1999.
- [31] O. Prove, S. Foulds, and V. Kendon, "Extending the controlled swap test to higher dimensions," 2021, *arXiv:2112.04333*.
- [32] S. Thanasilp, S. Wang, M. Cerezo, and Z. Holmes, "Exponential concentration and untrainability in quantum kernel methods," 2022, *arXiv:2208.11060*.
- [33] T. Hubregtsen, D. Wierichs, E. Gil-Fuster, P.-J. H. Derks, P. K. Faehrmann, and J. J. Meyer, "Training quantum embedding kernels on near-term quantum computers," *Phys. Rev. A*, vol. 106, no. 4, 2022, Art. no. 042431.
- [34] N. Cristianini, J. Shawe-Taylor, A. Elisseeff, and J. Kandola, "On kernel-target alignment," in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, 2001, vol. 14, pp. 367–373.
- [35] T. Wang, D. Zhao, and S. Tian, "An overview of kernel alignment and its applications," *Artif. Intell. Rev.*, vol. 43, no. 2, pp. 179–192, 2015.
- [36] X. Ren and J. Malik, "Learning a classification model for segmentation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2003, vol. 2, pp. 10–17.
- [37] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2274–2282, 2012.
- [38] S. Mohajerani and P. Saeeedi, "Cloud-Net: An end-to-end cloud detection algorithm for landsat 8 imagery," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2019, pp. 1029–1032.
- [39] S. Mohajerani, T. A. Krammer, and P. Saeeedi, "A cloud detection algorithm for remote sensing images using fully convolutional neural networks," in *Proc. IEEE 20th Int. Workshop Multimedia Signal Process.*, 2018, pp. 1–5.
- [40] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Representations*, 2015.
- [41] Z. Zhu, S. Wang, and C. E. Woodcock, "Improvement and expansion of the Fmask algorithm: Cloud, cloud shadow, and snow detection for Landsats 4–7, 8, and Sentinel 2 images," *Remote Sens. Environ.*, vol. 159, pp. 269–277, 2015.
- [42] S. Mahajan and B. Fataniya, "Cloud detection methodologies: Variants and development—A review," *Complex Intell. Syst.*, vol. 6, no. 2, pp. 251–261, Jul. 2020, doi: [10.1007/s40747-019-00128-0](https://doi.org/10.1007/s40747-019-00128-0).
- [43] S. Fei, L. Li, Z. Han, Z. Chen, and Y. Xiao, "Combining novel feature selection strategy and hyperspectral vegetation indices to predict crop yield," *Plant Methods*, vol. 18, no. 1, Nov. 2022, Art. no. 119, doi: [10.1186/s13007-022-00949-0](https://doi.org/10.1186/s13007-022-00949-0).
- [44] C. Liu, R. Tao, W. Li, M. Zhang, W. Sun, and Q. Du, "Joint classification of hyperspectral and multispectral images for mapping coastal wetlands," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 982–996, 2021, doi: [10.1109/JSTARS.2020.3040305](https://doi.org/10.1109/JSTARS.2020.3040305).
- [45] J. Nalepa, M. Kawulok, N. Longépé, B. Le Saux, and J. Nalepa, "Graph neural networks extract high-resolution cultivated land maps from Sentinel-2 image series," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, 2022, Art. no. 5513105, doi: [10.1109/LGRS.2022.3185407](https://doi.org/10.1109/LGRS.2022.3185407).
- [46] J. Nalepa, M. Myller, and M. Kawulok, "Validating hyperspectral image segmentation," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 8, pp. 1264–1268, Aug. 2019.
- [47] J. Nalepa and M. Stanek, "Segmenting hyperspectral images using spectral convolutional neural networks in the presence of noise," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2020, pp. 870–873.



Artur Miroszewski received the Ph.D. degree in the field of theoretical physics from the National Center for Nuclear Research, Otwock, Poland, in 2021.

He is a Postdoctoral Researcher with the Jagiellonian University, Krakow, Poland. He is currently involved in a European Space Agency project, delving into the potential of quantum machine learning for satellite data analysis. His main research interest include the application of kernel methods for classification tasks.



Jakub Mielczarek received the dual M.Sc. degrees in theoretical physics and astronomy, the Ph.D. degree in physical sciences, and the habilitation degree in physical sciences, all from the Jagiellonian University, Krakow, Poland, in 2008, 2012, and 2020, respectively.

He held two postdoctoral fellowships, with the Laboratoire de Physique Subatomique et de Cosmologie (LPSC), Grenoble, France, in 2014–2015 and with the Centre de Physique Théorique (CPT), Marseille, France, in 2018–2019, accompanied by numerous shorter internships. He is currently an Associate Professor with the Jagiellonian University. He is currently leading the Quantum Cosmos Lab, an interdisciplinary team exploring novel research areas emerging at the interface of theoretical physics and advanced technologies. This notably encompasses such topics as quantum simulations of gravitational physics and the application of quantum information to space technologies. He is also the Coordinator of an academic makerspace—Garage of Complexity. This unique academic unit cultivates a conducive environment for scholars and innovators to transform scientific and technical ideas into reality.



Grzegorz Czelusta is currently working toward the Ph.D. degree in theoretical physics with the Jagiellonian University, Krakow, Poland, specializing in research at the interface between quantum mechanics and gravitational physics.



Filip Szczepanek received the B.Sc. degree in computer science from Jagiellonian University, Kraków, Poland, in 2021.

He is currently a Software Developer with Ericsson R&D, Kraków, where he works on improving 4G and 5G core network. His research interests include machine learning, practical applications of quantum computing, programming language theory, and computer networks.

Bartosz Grabowski received the master's degree in mathematics from the Silesian University of Technology, Gliwice, Poland, in 2019.

He works as a Software Engineer with KP Labs, Gliwice, developing and implementing deep learning algorithms for satellite data analysis. He participated in multiple projects with KP Labs, e.g., Beetles, Sat-2, and Intuition-1. His research interests include deep learning methods applied to multi- and hyperspectral data analysis.

Mr. Bartosz was the recipient of the Diamond Grant, awarded by the Ministry of Science and Higher Education of Poland to 100 young researchers, enabling them to conduct their own research projects.



Bertrand Le Saux (Senior Member, IEEE) received the Ms.Eng. and M.Sc. degrees in electrical engineering from Institut National Polytechnique, 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 European Space Agency Lab, Frascati, Italy. His research interests include visual understanding of the environment by data-driven techniques including computer vision and (quantum) machine learning. He is also interested in tackling practical problems that arise in Earth observation, to bring solutions to current environment and society challenges.

Dr. Le Saux was the Co-Chair, from 2015 to 2017, and the Chair, from 2017 to 2019, for the IEEE GRSS Technical Committee on Image Analysis and Data Fusion. He is an Associate Editor for the *Geoscience and Remote Sensing Letters*. He coorganizes many workshops and events in machine learning x Earth observation, notably the CVPR Earth Vision workshop series, the ESA—ECMWF workshop series on Machine Learning for Earth System Observation and Prediction, and the Humanitarian Assistance and Disaster Response (HADR) workshop.



Jakub Nalepa (Senior Member, IEEE) received the M.Sc., Ph.D., and D.Sc. degrees in computer science from the Silesian University of Technology, Gliwice, Poland, in 2011, 2016, and 2021, respectively.

He is currently an Associate Professor with the Silesian University of Technology. He is also the Head of artificial intelligence (AI) with KP Labs, Gliwice, where he shapes the scientific and industrial AI objectives of the company. He has been pivotal in designing the onboard deep learning capabilities of Intuition-1 and has contributed to other missions,

including CHIME and OPS-SAT. His research interests include (deep) machine learning, hyperspectral data analysis, signal processing, remote sensing, and tackling practical challenges that arise in Earth observation to deploy scalable solutions.

Dr. Nalepa was the General Chair of the HYPERVIEW Challenge at the 2022 IEEE International Conference on Image Processing, focusing on the estimation of soil parameters from hyperspectral images onboard Intuition-1 to maintain farm sustainability by improving agricultural practices.