BULLETIN OF THE POLISH ACADEMY OF SCIENCES
TECHNICAL SCIENCES, Vol. 73(5), 2025, Article number: e154279
DOI: 10.24425/bpasts.2025.154279

Potential of quantum machine learning for processing multispectral Earth observation data

Manish K. GUPTA[1], **Michał ROMASZEWSKI**[2], and **Piotr GAWRON**[1]*

[1] Nicolaus Copernicus Astronomical Center, Polish Academy of Sciences, ul. Bartycka 18, Warsaw 00-716, Poland
[2] Institute of Theoretical and Applied Informatics, Polish Academy of Sciences, ul. Bałtycka 5, Gliwice 44-100, Poland

Abstract. Quantum computers with hundreds of noisy qubits are already available for the research community. They have the potential to run complex quantum computations well beyond the computational capacity of any classical device. It is natural to ask the question, what application these devices could be useful for? Land use and land cover classification of multispectral Earth observation data collected from the earth observation satellite mission is one such problem that is hard for classical methods due to its unique characteristics. In this work, we compare the performance of several classical machine learning algorithms on the stilted re-labeled dataset of the Copernicus Sentinel-2 mission, when the algorithm has access to projected quantum kernel (PQK) features. We show that the classification accuracy increases drastically when the model has access to PQK features. We then naively study the performance of these algorithms with and without access to PQK features on the original Copernicus Sentinel-2 mission data set. This study provides key evidence that shows the potential of quantum machine learning methods for Earth observation data.

Keywords: quantum computing; quantum machine learning; projected quantum kernel; multispectral image; Earth observation; Sentinel-2; land use and land cover classification; remote sensing.

1. INTRODUCTION

Quantum computers with more than 1000 noisy qubits are available to researchers; they will be able to run complex quantum calculations that are well beyond the computational capacity of any classical device. The computational power of these devices is expected to increase in the coming years as the noise in these devices is addressed with error-correcting codes and producers increase the number of physical qubits available. With such powerful devices at hand, it is natural to look for problems that are generally difficult for classical machines to solve and find algorithms to run on the new device. Earth observation (EO) is one such problem that is difficult due to its unique characteristics. It would be a futuristic step to develop algorithms that can run on these new devices and understand their limitation concerning Earth observation.

Earth observation and specifically land use and land cover classification (LULC) is an important task for achieving sustainable development goals (SDGs) [1]. Extracting knowledge from continuous multispectral and hyperspectral data quickly and effectively on-Earth objects and land covers, mapping them, and monitoring their changes on digital twins [2] is the need of the hour. The amount of remote sensing data that is being continuously captured by Earth observation satellites with onboard multispectral, hyperspectral, and radar sensors is in excess of 150 terabytes per day, which is not always processed efficiently [3]. The amount of data generated by EO missions falls in the category of Big Data. The massive amount of data comes with the so-called four challenges of Big Data referred to as "four Vs": Volume, Variety, Velocity, and Veracity [4,5]. Extracting meaningful information from such a huge volume of data efficiently requires special tools and methods. Although machine learning (ML) algorithms have shown great potential in terms of obtaining a detailed understanding of Earth observation data [6–9]. The amount of training data and available computational power limits the performance of these ML techniques. With the availability of a quantum computer with a promise to solve complex problems more efficiently than any available classical machine, it has become necessary to explore new quantum computing methods for understanding multispectral Earth observation data. There has been some progress in this area, where researchers have used some combination of classical and quantum parts for remote sensing data classification [10–18]. While aforementioned works focus mostly on solving machine learning tasks directly we extend the idea presented in [19] and study if there is a possibility of quantum advantage for Earth observation data processing. Our goal is more abstract and has no direct application in practice. We show that there exist scenarios where use of quantum classifiers significantly improves the results.

A large-scale, fully error-corrected quantum computer will not be built for many decades. Nevertheless, the recent advancement in their implementation allows us to study their application to real-life computational problems [20]. Quantum computers consisting of hundreds of noisy qubits are already available and can run specific quantum algorithms. One class of such algorithms uses quantum models to generate correlations between variables that are inefficient to represent using classical models of computation. Recent theoretical and experimental evidence

*e-mail: gawron@camk.edu.pl

Manuscript submitted 2024-09-13, revised 2025-03-25, initially accepted for publication 2025-04-10, published in August 2025.

suggests that quantum computers can efficiently sample probability distributions that are exponentially hard to sample classically [21, 22]. This is the type of advantage that is exploited by both quantum neural network (QNN) [23] and quantum kernel methods [24]. QNN parameterizes a distribution using a set of adjustable parameters, and the quantum kernel method encodes classical data into a quantum state as a feature map which maps the data in higher-dimensional quantum Hilbert space and uses a quantum computer to compute the inner products of quantum states [25].

It is postulated that quantum machine learning algorithms can outperform their classical counterparts, and the justification that is generally provided for this is that if the quantum circuit is hard to sample classically, then there is a potential for quantum advantage. Huang *et al.* [26] showed this argument to be incomplete and proved that with a sufficient amount of training data, classical models can be elevated to rival quantum models, even when the quantum circuit is hard to sample classically. They also proposed a "geometric difference" between kernel functions defined by classical and quantum machine learning models and showed that if the geometric difference is small, then the classical machine learning model is guaranteed to provide similar or better performance in prediction on the data set. If the geometric distance is large, then a data set exists where the quantum model exhibits a large prediction advantage. They also found that due to small geometric differences, a variety of common quantum models in the literature perform similarly or worse than the classical model. To circumvent it, they proposed the projected quantum kernel (PQK) method that generally enlarges the geometric difference between the kernels.

The mapping of land on Earth is categorized into land use (LU) classification and land cover (LC) classification. Although the two terms are used interchangeably they are different. According to the Food and Agriculture Organization (FAO) of the United Nations, "Land cover is the observed (bio)physical cover on the Earth's surface", while "The arrangements characterize land use, activities, and inputs by people to produce, change, or maintain a certain land cover type" [27]. In simple terms, land cover is what covers the surface of the Earth, e.g., classes: water, snow, grassland, deciduous forest, and bare soil and land use describe how the land is used, e.g., classes: wildlife management area, agricultural land, urban, and recreation areas. The two terms, land use and land cover are tightly coupled, and they are jointly classified, hence "land use and land cover" (LULC) classification is considered a more general concept.

Remote sensing missions capture the imagery using optical, thermal, or synthetic aperture radar (SAR) imaging systems. The optical sensor is sensitive to a spectrum range from visible to mid-infrared radiation from Earth's surface and captures panchromatic, multispectral, or hyperspectral images. Commonly, images with more than 2 to 13 spectral bands per pixel are called multispectral, while the images with hundreds of spectral bands are called hyperspectral.

The LULC classification problem is mathematically defined as an assignment $f : X \to Y$ from the set of spectral images to the set of pixel class arrays. The input space $X \subseteq \mathbb{R}^{W\times H\times B}$ represents the set of possible images with W, H, B being respectively the image width, height, and the number of spectral bands. The output space for pixel-level land cover classification is represented as $Y \subseteq C^{W\times H}$, where $C = \{0, 1, \ldots, N\} \subseteq \mathbb{Z}^{0+}$ is the set of LULC categories. A variety of approaches is used to perform the land use and land cover classification task. The approach that is used depends on the resolution of the image being processed, the exact task to be performed, and the available resources [28].

For this work, we use the Sentinel-2 data, which contains multispectral data with pixels of 10m resolution. For such a dataset, a common task is to perform semantic segmentation, where labels are assigned to every pixel [29] individually. The traditional machine learning (ML) techniques that are used for this task are support vector machines, decision trees, and perceptron-like neural networks. Semantic segmentation is often a two-step process. In the first step, pixel-by-pixel classification is performed where only the spectral information is used. In the second step, the labels are smoothed out by employing a probabilistic graphical model such as Markov random field or conditional random field. Occasionally, spatial and spectral information are used jointly in the semantic segmentation task [30]. In this work, we focus solely on spectral information classification, and we deliberately ignore the spatial relationship between pixels.

In this work, we are particularly interested in answering the question whether quantum machine learning [31, 32] is suitable for the classification of multispectral data such as, for example, data gathered by the Earth observation satellites. The specific goal is to study a quantum machine learning system for land use and land cover classification of the Earth's surface is based on Sentinel-2 images. We find evidence that shows that there exists a data set where classical model test accuracy increases drastically when the model has access to projected quantum kernel features. It is a follow-up to the previous work on multispectral image classification with QNN, where one of the authors of this work showed that the QNN-based classifier achieved a score of 66% in multi-class classification scenario [10] and extension of the work [19]. In the next section, we will start by providing a brief overview of key concepts used in this work, such as quantum kernel methods, geometric distance, and projected quantum kernel (PQK) method. We then describe the experiment, discuss the result, and conclude.

2. METHODOLOGY

2.1. Quantum kernel methods

In machine learning, kernel function $k : \mathbb{R}^B \times \mathbb{R}^B \to \mathbb{R}^{0+}$ can be understood as a measure of similarity — a generalized scalar product — between given feature vectors $x_i \in \mathbb{R}^B$ and $x_j \in \mathbb{R}^B$. Given a set of feature vectors $\{x_1, x_2, \ldots, x_{N_{\text{samples}}}\}$ and a kernel function k we can calculate a kernel matrix $[K_{i,j}]_{i=1,j=1}^{N_{\text{samples}},N_{\text{samples}}}$ with elements $K_{i,j} = k\left(x_i, x_j\right)$ that stores similarity between all N_{samples} feature vectors. Matrix K is positive semi-definite. And therefore we can calculate the geometric difference between two Kernel matrices K^1, K^2 associated with two kernel functions k^1, k^2. The geometric difference is defined over a dataset as

$$g\left(K^1||K^2\right) = \sqrt{\left\|\sqrt{K^1}\left(K^2\right)^{-1}\sqrt{K^1}\right\|_\infty}, \tag{1}$$

where $\|\cdot\|_\infty$ is the spectral norm of the resulting matrix. We assume that $\mathrm{Tr}\left\{(K^1)\right\} = \mathrm{Tr}\left\{(K^2)\right\} = N_{\text{samples}}$ [33]. To evaluate a potential for quantum advantage, we must calculate the geometric difference between quantum kernel and classical kernel. It is a crucial test for comparing classical and quantum machine learning models. The geometric difference for a quantum kernel is defined with respect to the closest efficient classical model. If the geometric distance is small, then the classical machine learning model is guaranteed to provide similar or better performance in prediction on the data set, independent of the function values or labels. If the geometric distance is large, then a data set exists where the quantum machine learning model exhibits a large prediction advantage. These notions provide an important test for finding potentially useful quantum machine learning models.

A number of quantum machine learning models found in literature can be shown to perform similarly or worse than classical machine learning models due to their small geometric differences. The small geometric difference is often due to the fact that encoded features are too far apart because of the exponentially large Hilbert space employed by existing quantum models. To solve this problem, the projected quantum kernel method that circumvents this issue and enlarges the geometric difference was proposed in [33].

Reading the information from the quantum computer requires performing a quantum measurement what requires the entire quantum circuit has to be executed on the quantum computer. Because of this fact calculating the value of a quantum kernel function between feature vectors requires multiple executions of the quantum circuit implementing said function. It was recently observed that in many cases one can need an exponentially growing, in function of number of qubits, number of measurements [34,35] needed in order to be able to estimate the value of a kernel function. This makes the application of quantum kernel futile in such a case since any possible quantum advantage is lost. Fortunately, in our case, we use the PQK kernel with relatively shallow linear entanglement generating circuit — it was shown in [35] that this particular kind of kernel does not exhibit concentration properties and therefore, in principle, could provide quantum advantage.

2.2. Projected quantum kernel

Projected quantum kernels (PQK) are a family of kernels that work by projecting the quantum states to an approximate classical representation, for example, reducing physical observables or classical shadows [33] and then defining the kernel function using the classical representation. The modified quantum kernel is referred to as the projected quantum kernel. The PQK method was first introduced by Huang *et al.* in [26]. The projection reduces the large training set dimension to a smaller classical space that generalizes better. The projected quantum kernel is defined on the classical feature space to evade the difficulty in learning due to the exponential dimension in quantum Hilbert space. Projecting an exponentially large Hilbert space using a projected quantum kernel is a difficult task on a classical computer. One of the simplest examples of a projected quantum kernel is to measure the one-particle reduced density matrix (1-RDM) on all qubits for the encoded state, $\rho_k(x_i) = \mathrm{Tr}_{j\neq k}\left[\rho(x_i)\right]$, and then define the kernel as

$$k^{\mathrm{PQ}}(x_i, x_j) = \exp\left(-\gamma^{\mathrm{PQ}} \sum_k \left\|\rho_k(x_i) - \rho_k(x_j)\right\|_F^2\right), \quad (2)$$

where γ^{PQ} is a real positive hyperparameter. The partial trace $\mathrm{Tr}_{j\neq k}$ over qubits labeled by j can be defined as follows

$$\mathrm{Tr}_{j\neq k}\left[\rho(x_i)\right] = \sum_{j\in\{0,1\}^{k-1}} \sum_{j'\in\{0,1\}^{D-k+1}} \mathrm{Tr}[(P_j \otimes \mathbb{1} \otimes P'_{j'})\rho(x_i)],$$

where $P_j = |j\rangle\langle j|$.

2.3. Computing PQK features

To compute the PQK features for a given data instance x_i, we encode this data instance into the quantum state

$$|\psi_i\rangle = V(x_i/n_{\text{trotter}})^{n_{\text{trotter}}} U_{\text{qb}}|0\rangle, \quad (3)$$

where $U_{\text{qb}} = \bigotimes_{j=1}^{N} R_x(\phi_1^j) R_y(\phi_2^j) R_z(\phi_3^j)$ is the tensor product of Pauli rotations operators, $R_x(\phi) = e^{-iX\phi/2}$, angles are randomly selected once as $\phi \sim U(-2\pi, 2\pi)$ and remain equal for all data points. The integer n_{trotter} is a hyperparameter — the number approximation steps for approximating time evolution [36, 37], we set arbitrarily $n_{\text{trotter}} = 10$ in our experiments. The unitary $V(\hat{\theta})$ is defined as

$$V(\hat{\theta}) = \exp\left(-i \sum_{j=1}^{N} \sum_{l\in\{X,Y,Z\}} \hat{\theta}_j \sigma_j^l \sigma_{j+1}^l\right), \quad (4)$$

where σ_j^l acts on j-th qubit and N is number of qubits. We compute the PQK features based on the 1-RDM by measuring the expectation values of $\langle\psi_i|\sigma_j^l|\psi_i\rangle$, where i indexes over data points, j indexes over qubits and l indexes over Pauli operators $\{X, Y, Z\}$. Mathematically it can be represented as $f_{\text{PQK}} : \mathbb{R}^d \to \mathbb{R}^{3(d+1)}$ where $f_{\text{PQK}}(x_i) = [\langle\psi_i|\sigma_j^l|\psi_i\rangle]_{j\in\{1,2,\ldots,d+1\}, l\in\{X,Y,Z\}}$ and $d = N - 1$ is the number of features.

2.4. Preparing stilted dataset

To achieve maximum separation between quantum and classical models, we artificially re-label the dataset using the spectral information found in the classical and PQK kernel matrices. To achieve that, we will perform the following three steps.

We first train the radial basis function (RBF) kernel support vector machine using the "scikit-learn" library [38] and "scikit-optimize" library to obtain the best gamma $\gamma_{\text{best}}^{\text{RBF}}$ using the original dataset — original feature vectors and labels.

Next, we compute the kernel matrix for the best classical model using found $\gamma_{\text{best}}^{\text{RBF}}$ and the original feature vectors. We also compute the quantum kernel matrix using feature vectors transformed by f_{PQK}, and $\gamma^{\text{PQ}} = 1$.

Finally, we construct the new stilted dataset that will yield the largest separation between quantum and classical models from a learning-theoretic sense by assigning new labels.

The new labels $\boldsymbol{y}_{\text{relabel}}$ are obtained from vector $v' = \sqrt{K^Q}v$, where v is the eigenvector of $\sqrt{K^Q}\left(K^C\right)^{-1}\sqrt{K^Q}$ corresponding to the eigenvalue of $g^2 = \left\|\sqrt{K^Q}\left(K^C\right)^{-1}\sqrt{K^Q}\right\|_\infty$, by assigning $\boldsymbol{y}_{\text{relabel},i} = \mathbb{1}_{v'_i > \text{median}(\{v'_i\}_i)}$ and changing 5% labels randomly.

This relabeling of the data maximizes the separation between the quantum and classical models by maximizing the geometric distance between the classical kernel and the PQK kernel. For a detailed description, we refer to Appendix G: "Constructing dataset to separate quantum and classical model" of [26]. The geometric distance [26] between the kernel of classical and quantum models is defined as

$$g\left(K^C||K^Q\right) = \sqrt{\left\|\sqrt{K^Q}\left(K^C\right)^{-1}\sqrt{K^Q}\right\|_\infty}, \tag{5}$$

where K^C and K^Q are kernel matrices for the classical and quantum models respectively.

2.5. Classifiers

The support vector machine (SVM) with the radial basis function (RBF) kernel was used in our classification experiments as a high-performance method with efficient, stable implementation [38]. The classifier was also used to prepare the stilted dataset in Section 2.4 and is a standard reference classifier for hyperspectral and multispectral classification [39]. As a reference, we also used three well-known classifiers combining high performance with explainable decision-making: a k-nearest neighbors (k-NN), a decision tree (DT), and a naive Bayes (NB).

3. EXPERIMENTS

In this study, we use Copernicus Sentinel-2 Earth observation land cover multispectral image data. Sentinel-2 is a European wide-swath, high-resolution, multispectral imaging mission. It comprises a constellation of two polar-orbiting satellites that aim to capture land cover changes monitoring and natural disaster management. It carries an onboard multispectral imager (MSI) sensor that samples 13 spectral bands: four bands at 10 m, six bands at 20 m, and three bands at 60 m spatial resolution. The orbital swath width is 290 km [40]. The dataset used in the experiments, presented in Fig. 1, is a multispectral cube $\boldsymbol{C} \in \mathbb{R}^{512\times512\times12}$ of measured reflectance values. Multispectral pixels are categorized into sixteen land use (LU) categories [10, 41]. For this work, we use only four land use categories: "Artificial surfaces and constructions", "Cultivated areas", "Broadleaf tree cover" and "Herbaceous vegetation" and perform a binary classification task on pairs of classes selected out of the combinations of four classes.

The experiments aim to evaluate the impact of PQK feature extraction on multispectral classification accuracy. To achieve this, we compare classifier performance (SVM, k-NN, DT, NB) using PQK feature vectors with performance using PCA-reduced spectral vectors (original data prior to PQK extraction). Additionally, a baseline, naive comparison is conducted using the original Sentinel-2 dataset, evaluating classifier performance

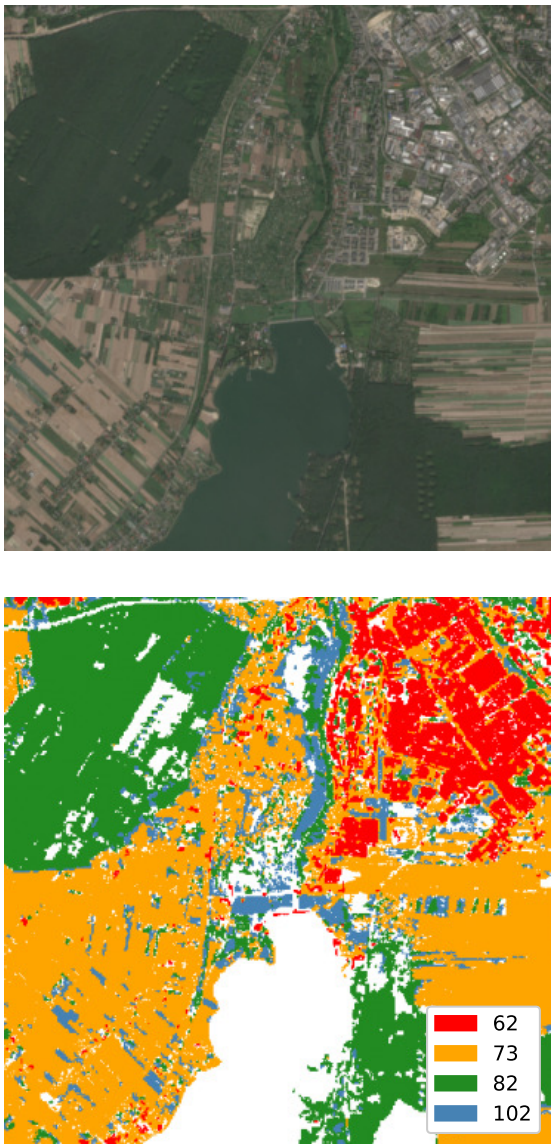


Fig. 1. RGB and GT visualization of the dataset used in experiments with four classes: "Artificial surfaces and constructions" (62), "Cultivated areas" (73), "Broadleaf tree cover" (82), "Herbaceous vegetation" (102). Note that the stilted datasets generated, as explained in Section 2.4, have different sets of labels.

with both PQK and PCA-reduced spectral vectors using original class labels. Each experimental setup involves three main steps: binary dataset preparation, stilted dataset generation, and classification.

3.1. Binary datasets preparation

The experiments utilize binary (two-class) datasets generated by selecting subsets of labeled multispectral pixels from a multispectral image. Labels are created from all combinations of four land use (LU) classes, resulting in six class pairs. To manage memory requirements associated with computing PQK features, we randomly select a subset of 1000 labeled pixels for each class and reassign class labels to 0 and 1. These reduced datasets are

employed in subsequent experiments. This sampling procedure is repeated $N_{\text{repetitions}} = 10$ times to create multiple instances of the training dataset $\mathscr{T}_i^c$ where c denotes a pair of classes and $i \in \{0, 1, \ldots, N_{\text{repetitions}} - 1\}$ denotes an instance of the experiment.

3.2. Stilted dataset preparation

A binary dataset $\mathscr{T} = (\boldsymbol{X}, \boldsymbol{y})$, where $\boldsymbol{X} \in \mathbb{R}^{l \times m}$ denotes an array of pixels with multispectral features, and $\boldsymbol{y}$ denotes the vector of class labels. The aim of this step is to engineer a stilted dataset $\mathscr{T}_{\text{stilted}} = (\boldsymbol{X}^{\text{pqk}}, \boldsymbol{y}_{\text{relabel}})$ where $\boldsymbol{X}^{\text{pqk}}$ denotes an array of PQK features and $\boldsymbol{y}_{\text{relabel}}$ denotes a vector of new labels, resulting from the relabeling of class labels such as to maximize the geometric distance between classical and quantum models. The procedures are described in Section 2.2, Section 2.3 and Section 2.4 and can be summarized in the following steps:

1. *Train/test split*: Binary dataset $\mathscr{T} = (\boldsymbol{X}, \boldsymbol{y})$ is split into train set $\mathscr{T}_{\text{train}} = (\boldsymbol{X}_{\text{train}}, \boldsymbol{y}_{\text{train}})$ and test set $\mathscr{T}_{\text{test}} = (\boldsymbol{X}_{\text{test}}, \boldsymbol{y}_{\text{test}})$.
2. *Standardization (calculating z-score)*: Arrays $\boldsymbol{X}_{\text{train}}, \boldsymbol{X}_{\text{test}}$ are standardized into $\boldsymbol{X}_{\text{train}}^{\text{standarised}}, \boldsymbol{X}_{\text{test}}^{\text{standarised}}$.
3. *PCA (dimension reduction)*: Arrays $\boldsymbol{X}_{\text{train}}^{\text{standarised}}, \boldsymbol{X}_{\text{test}}^{\text{standarised}}$ are reduced to 4 features $\boldsymbol{X}_{\text{train}}^{\text{reduced}}, \boldsymbol{X}_{\text{test}}^{\text{reduced}}$. We use only $\boldsymbol{X}_{\text{train}}^{\text{reduced}}$ to fit the PCA model.
4. *SVM kernel optimization*: Grid-search is performed on $(\boldsymbol{X}_{\text{train}}^{\text{reduced}}, \boldsymbol{y}_{\text{train}})$ to find the best value of the parameter γ.
5. *PQK features extraction*: PQK procedure as described in Section 2.3 is used on the arrays $\boldsymbol{X}_{\text{train}}^{\text{reduced}}, \boldsymbol{X}_{\text{test}}^{\text{reduced}}$ to compute $\boldsymbol{X}_{\text{train}}^{\text{pkq}}, \boldsymbol{X}_{\text{test}}^{\text{pkq}}$.
6. Stilted dataset creation: The training set $\boldsymbol{X}_{\text{train}}^{\text{reduced}}, \boldsymbol{X}_{\text{test}}^{\text{reduced}}$, PQK features $\boldsymbol{X}_{\text{train}}^{\text{pkq}}, \boldsymbol{X}_{\text{test}}^{\text{pkq}}$, and best gamma value γ are used to assign new labels $\boldsymbol{y}_{\text{relabel}}$ as described in Section 2.4.

3.3. Classification experiment

To compare the classification accuracy of classifiers (SVM, k-NN, DT, NB) over the original and the stilted datasets, we use a standard two-stage cross-validation experiment. For every binary dataset $\mathscr{T} = (\boldsymbol{X}, \boldsymbol{y})$, its labeled examples are divided into training sets $\mathscr{T}_{\text{train}}^k = (\boldsymbol{X}_{\text{train}}^k, \boldsymbol{y}_{\text{train}}^k)$ and test sets $\mathscr{T}_{\text{test}}^k = (\boldsymbol{X}_{\text{test}}^k, \boldsymbol{y}_{\text{test}}^k)$ using k-fold stratified cross-validation with a number of folds $k = 5$. For every fold, the following steps are performed:

1. *Standardization*: Training and tests are standardized using mean and standard deviation computed on the training set.
2. *Parameter selection*: Classifier parameters are selected using grid-search and internal stratified 3-fold cross-validation on the training set.
3. *Classifier testing*: The accuracy of the classifier that is trained on the training set is computed using a test set.

The final accuracy of a classification experiment is computed by averaging accuracies for every fold.

3.4. Parameter selection and implementation

In our experiments, we used classifiers and PCA implementation from scikit-learn [38] library. Unless stated otherwise, datasets were partitioned into training and testing sets using an 80%/20% split.

The range of selected parameters for each classifier is as follows:

- SVM: Parameters $\gamma^{\text{RBF}} \in \{10^{-2}, \ldots, 10^2\}$, $C \in \{10^{-2}, \ldots, 10^2\}$.
- k-NN: Number of neighbors $k \in \{1, \ldots, 15\}$ with two different neighbors weighting strategies — *uniform*: where all neighbors have the same weight, *distance*: where neighbors are weighted by the inverse of their distance.
- DT: The minimum number of examples required to split a node $s \in \{2, 3, 4\}$, maximum depth of the tree $d \in \{2, 3, 5, 10, \alpha\}$, where α denotes expansion of nodes until all leaves contain less than s examples.
- NB: The classifier is non-parametric.

An experimental run, encompassing dataset preparation, PQK feature computation, and classification across 10 repetitions, required approximately 9 hours to complete.

3.5. Classifier comparison

To compare results between classifiers, we use the Bayesian approach, adapted from [42]. This method avoids limitations of traditional null hypothesis significance testing (NHST), for example, the fact that point-wise null hypotheses are usually false, provided that a sufficiently large number of data points is available, as well as the difficulty in interpreting outcomes upon rejection of the null hypothesis.

Bayesian approach evaluates the posterior distribution of classifier performance differences, employing a region of practical equivalence (ROPE) to establish meaningful differences. The outcome can be directly interpreted as probability P(B) that, on average, method B is more accurate than method A (or that methods are practically equivalent). Our analysis involves multiple datasets (six class pairs); therefore, we adopted a hierarchical Bayesian approach with a ROPE value of 2% accuracy. The method also includes a simplex visualisation of the comparison outcome described in detail in [42].

3.6. Experiment list

In order to fully assess the influence of PQK features on the classification accuracy, the classification experiment described in Section 3.3 was repeated for four combinations of original and PQK features on original and stilted datasets:

- Classification with reduced spectral features on stilted dataset: binary datasets had the form: $\mathscr{T} = (\boldsymbol{X}^{\text{reduced}}, \boldsymbol{y}_{\text{relabeled}})$.
- Classification with PQK features on stilted dataset: binary datasets had the form: $\mathscr{T} = (\boldsymbol{X}^{\text{pqk}}, \boldsymbol{y}_{\text{relabeled}})$.
- Classification with reduced spectral features on original dataset: binary datasets had the form: $\mathscr{T} = (\boldsymbol{X}^{\text{reduced}}, \boldsymbol{y})$.
- Classification with PQK features on original dataset: binary datasets had the form: $\mathscr{T} = (\boldsymbol{X}^{\text{pqk}}, \boldsymbol{y})$.

4. RESULTS AND DISCUSSION

Results of the experiments are presented in two tables summarising averages over all 6 combinations of labels, 10 experiment repetitions, and 5 cross-validation rounds. Since our classifica-

tion dataset is balanced and the label assignment is arbitrary (with no inherent positive/negative class distinction), accuracy well represents classification performance. However, since the Matthews correlation coefficient (MCC) can sometimes provide a more interpretable measure, MCC results are included in the appendix.

Table 1 contains mean training and test accuracies for the classification task using relabeled stilted datasets. The introduction of PQK features substantially enhances classification accuracy compared to spectral features. This difference is significant when comparing best classifiers in both scenarios using the methodology described in Section 3.5 with $P(\text{SVM}) = 1$.

Table 1
Mean training and test accuracy of classification for relabeled datasets

Features classifier	Mean training accuracy		Mean test accuracy	
	Original	PQK	Original	PQK
DT	0.83 ± 0.09	0.94 ± 0.04	$\mathbf{0.70 \pm 0.03}$	0.78 ± 0.02
KNN	$\mathbf{0.99 \pm 0.03}$	$\mathbf{0.98 \pm 0.05}$	0.69 ± 0.02	0.83 ± 0.02
NB	0.57 ± 0.02	0.82 ± 0.01	0.56 ± 0.03	0.82 ± 0.02
SVM	0.91 ± 0.06	0.93 ± 0.01	0.68 ± 0.03	$\mathbf{0.91 \pm 0.02}$

Table 2 reports mean training and test accuracies for datasets with their original labels. In this — more realistic — scenario, we do not observe an improvement in classification accuracy. In contrast, the transformation of original features using PQK features leads to a notable decrease in accuracy.

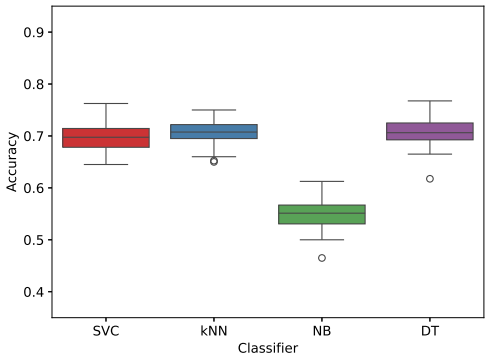
Table 2
Mean training and test accuracy of classification for datasets with original labels

Features classifier	Mean training accuracy		Mean test accuracy	
	Original	PQK	Original	PQK
DT	0.94 ± 0.01	0.79 ± 0.13	0.92 ± 0.01	0.60 ± 0.03
kNN	$\mathbf{0.98 \pm 0.03}$	$\mathbf{0.98 \pm 0.06}$	0.92 ± 0.01	0.66 ± 0.03
NB	0.89 ± 0.01	0.61 ± 0.01	0.89 ± 0.01	0.60 ± 0.03
SVM	0.93 ± 0.01	0.81 ± 0.08	$\mathbf{0.93 \pm 0.01}$	$\mathbf{0.68 \pm 0.03}$

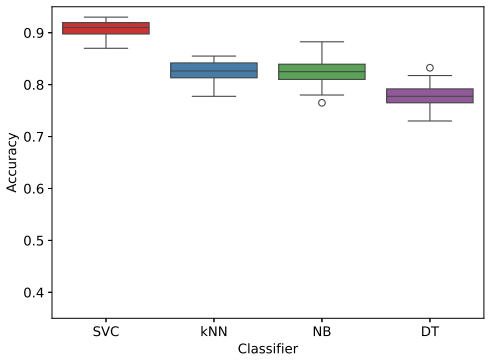
The results in Table 1 indicate that two of the classifiers, SVM and NB, achieve high accuracy for PQK features, and their result for test data is similar to the result for training data. This is not the case with k-NN and DT classifiers, for which significantly higher accuracy on the test set may indicate overtraining. This effect can also be observed in Table 2 for the k-NN classifier.

Since for small and medium-sized datasets SVM is considered an SOA classifier, we compare its accuracy with next best method in every scenario. For relabeled datasets (Table 1) applying the comparison method described in Section 3.5, we determine that for PQK features SVM outperforms the next best kNN with probability $P(\text{SVM}) = 1$ while for spectral features, SVM is practically equivalent to the DT classifier with $P(\text{ROPE}) = 0.822$ and $P(\text{SVM}) = 0.02$. For datasets with original labels (see Table 1) and spectral features, we determined that SVM is practically equivalent to other methods with $P(\text{ROPE}) = 0.997$ and $P(\text{SVM}) = 0.02$. This equivalence may result from a simplification of input data due to PCA. However, with PQK features, SVM gains some advantage over the next-best kNN classifier with $P(\text{SVM}) = 0.97$, suggesting that more complex models (e.g., neural network-based) might uncover additional meaningful structures. A simplex visualization of this case is provided in Fig. 4.

For the result visualization, we provide box plots of classification accuracies in Fig. 2 and Fig. 3 that represent the accuracy of classifiers (SVM, k-NN, DT, NB) over the stilted and original datasets created using the spectral data for two classes: "Broadleaf tree cover" and "Herbaceous vegetation". The classification accuracies for the original features are shown in Fig. 2a. The classification accuracies for the PQK features are shown in Fig. 2b. Visibly, the accuracy increases significantly over the same stilted datasets if the classifiers have access to the PQK features. We also naively repeat the experiment over original datasets to assess the effectiveness of PQK features, and the result is shown in Fig. 3a and Fig. 3b.

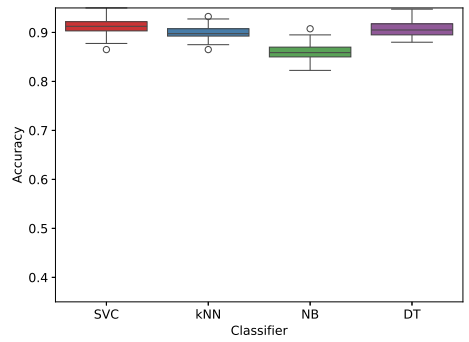


(a)

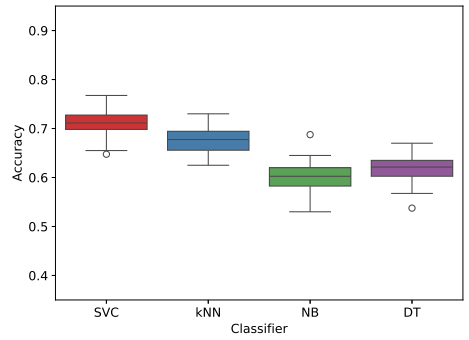


(b)

Fig. 2. 5-fold cross-validation accuracy on relabeled data for classes "Broadleaf tree cover — 82", and "Herbaceous vegetation — 102". (a) Without PQK features. (b) With PQK features



(a)



(b)

Fig. 3. 5-fold cross-validation accuracy on original data for classes "Broadleaf tree cover — 82", and "Herbaceous vegetation — 102". (a) Without PQK features. (b) With PQK features

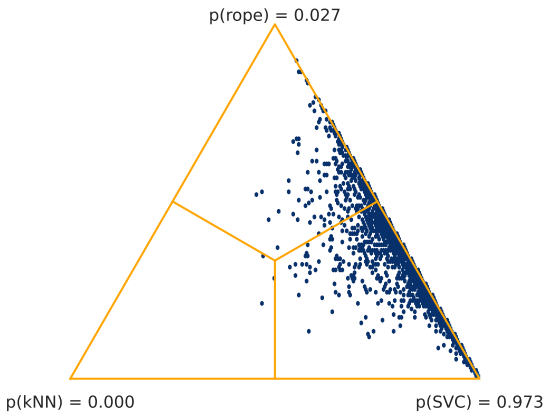


Fig. 4. Advantage of SVM over kNN in classification experiment for datasets with original labels and PQK features, visualized using method from [42]. There is a 97.3% probability that average SVM performance is higher for a rope value of 2%

5. CONCLUSIONS

The main conclusion we draw from this empirical investigation is that there exists a dataset that is easy for the quantum model to learn and hard for the classical model to learn. We showed that all the important classical ML models generally used for image segmentation underperform on the stilted data set using standard spectral features. We provide key evidence that suggests that quantum procedures, in addition to classical methods, could give an advantage in learning tasks for Earth observation datasets. There are at least two open questions that needs further investigation:

- Are there any other classical ML models that perform better using only the standard spectral features over the stilted dataset?
- Does there exist a natural EO dataset that matches the characteristics of the stilted dataset?

If we can show the answer to the first question as false and find a natural EO dataset where we see an advantage with PQK features, then we can confidently claim that quantum computers would be useful in processing EO data.

Our investigation, presented in this paper, is limited by many factors: due to substantial computational resource requirements associated with generating PQK features, our experimental datasets are small subsets of the whole image; we use only a particular data encoding and quantum kernel out of many; we consider a small ideal simulated quantum computer, and we also assume an infinite number of samples sampled from the quantum circuits that encode the data. Yet, we can claim that this empirical study shows the potential for quantum machine learning methods to Earth observation data analysis and encourages us to perform further investigation.

APPENDIX

Classification performance measured with Matthews correlation coefficient (MCC) is presented in Table 3 for the relabeled datasets and in Table 4 for datasets with original labels.

The source code and dataset allowing for replication of this study are available at https://doi.org/10.5281/zenodo.15513886

Table 3
Mean training and test Matthews correlation coefficient (MCC) for relabeled datasets

Features classifier	Mean training accuracy		Mean test accuracy	
	Original	PQK	Original	PQK
dtree	0.66 ± 0.18	0.86 ± 0.09	$\mathbf{0.40 \pm 0.05}$	0.55 ± 0.04
knn	$\mathbf{0.98 \pm 0.06}$	$\mathbf{0.96 \pm 0.09}$	0.38 ± 0.05	0.66 ± 0.04
nb	0.14 ± 0.03	0.65 ± 0.02	0.13 ± 0.06	0.64 ± 0.05
SVM	0.84 ± 0.12	0.85 ± 0.02	0.37 ± 0.05	$\mathbf{0.81 \pm 0.03}$

Table 4
Mean training and test Matthews correlation coefficient (MCC) for datasets with original labels

Features classifier	Mean training accuracy		Mean test accuracy	
	Original	PQK	Original	PQK
dtree	0.87 ± 0.02	0.56 ± 0.24	0.83 ± 0.03	0.21 ± 0.05
knn	$\mathbf{0.96 \pm 0.06}$	$\mathbf{0.96 \pm 0.12}$	0.84 ± 0.02	0.31 ± 0.05
nb	0.79 ± 0.01	0.23 ± 0.03	0.79 ± 0.03	0.21 ± 0.06
SVM	0.87 ± 0.01	0.66 ± 0.18	$\mathbf{0.86 \pm 0.03}$	$\mathbf{0.37 \pm 0.05}$

ACKNOWLEDGEMENTS

The data used in this work was prepared by project team: S. Lewiński, R. Malinowski, M. Rybicki, E. Gromny, M. Jenerowicz, Marcin Krupiński, C. Wojtkowski, Michał Krupiński from Space Research Centre of the Polish Academy of Sciences, E. Krätzschmar and S. Guñther from IAB GmbH. The authors would like to acknowledge support from IRAP Astro-CeNT (MAB/2018/7) funded by FNP from ERDF and ESA under the contract No. 4000137375/22/NL/GLC/my. The project W6/ESA/2023 was co-funded by the Polish Ministry of Science and Higher Education under the International Co-Financed Projects program (contract No. 5304/ESA/2023/0).

REFERENCES

[1] B. Ferreira, M. Iten, and R.G. Silva, "Monitoring sustainable development by means of earth observation data and machine learning: a review," *Environ. Sci. Eur.*, vol. 32, no. 1, p. 120, 2020, doi: 10.1186/s12302-020-00397-4.

[2] P. Bauer, B. Stevens, and W. Hazeleger, "A digital twin of earth for the green transition," *Nat. Clim. Chang.*, vol. 11, no. 2, pp. 80–83, 2021, doi: 10.1038/s41558-021-00986-y.

[3] ESA Earth Online, "Working towards AI and Earth observation." Mar 2019. [Online]. Available: https://www.esa.int/Applications/Observing_the_Earth/Working_towards_AI_and_Earth_observation

[4] P. Zikopoulos and C. Eaton, *Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data*. McGraw-Hill Osborne Media: New York, NY, USA, 2011, 2011.

[5] A. Vali, S. Comai, and M. Matteucci, "Deep learning for land use and land cover classification based on hyperspectral and multispectral earth observation data: A review," *Remote Sens.*, vol. 12, no. 15, p. 2495, 2020, doi: 10.3390/rs12152495.

[6] L. Zhang and B. Du, "Deep learning for remote sensing data: A technical tutorial on the state of the art," *IEEE Geosci. Remote Sens. Mag.*, vol. 4, no. 2, pp. 22–40, 2016, doi: 10.1109/MGRS.2016.2540798.

[7] N. Audebert *et al.*, "Deep learning for urban remote sensing," in *2017 Joint Urban Remote Sensing Event (JURSE)*, 2017, pp. 1–4, doi: 10.1109/JURSE.2017.7924536.

[8] X.X. Zhu *et al.*, "Deep learning in remote sensing: A comprehensive review and list of resources," *IEEE Geosci. Remote Sens. Mag.*, vol. 5, no. 4, pp. 8–36, 2017, doi: 10.1109/MGRS.2017.2762307.

[9] G. Cheng, X. Xie, J. Han, L. Guo, and G.S. Xia, "Remote sensing image scene classification meets deep learning: Challenges, methods, benchmarks, and opportunities," *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 3735–3756, 2020, doi: 10.1109/JSTARS.2020.3005403.

[10] P. Gawron and S. Lewiński, "Multi-spectral image classification with quantum neural network," in *IGARSS 2020 – 2020 IEEE International Geoscience and Remote Sensing Symposium*, 2020, pp. 3513–3516, doi: 10.1109/IGARSS39084.2020.9323065.

[11] D.A. Zaidenberg, A. Sebastianelli, D. Spiller, B. Le Saux, and S.L. Ullo, "Advantages and bottlenecks of quantum machine learning for remote sensing," in *2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS*, 2021, pp. 5680–5683, doi: 10.1109/IGARSS47720.2021.9553133.

[12] S. Otgonbaatar and M. Datcu, "Assembly of a coreset of earth observation images on a small quantum computer," *Electronics*, vol. 10, no. 20, p. 2482, 2021, doi: 10.3390/electronics10202482.

[13] 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," *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.*, vol. 15, pp. 565–580, 2022, doi: 10.1109/JSTARS.2021.3134785.

[14] S. Otgonbaatar and M. Datcu, "Classification of remote sensing images with parameterized quantum gates," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, pp. 1–5, 2022, doi: 10.1109/LGRS.2021.3108014.

[15] S.Y. Chang, B.L. Saux, S. Vallecorsa, and M. Grossi, "Quantum convolutional circuits for earth observation image classification," in *IGARSS 2022 - 2022 IEEE International Geoscience and Remote Sensing Symposium*, 2022, pp. 4907–4910, doi: 10.1109/IGARSS 46834.2022.9883992.

[16] A. Delilbasic, B. Le Saux, M. Riedel, K. Michielsen, and G. Cavallaro, "A single-step multiclass SVM based on quantum annealing for remote sensing data classification," *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.*, vol. 17, pp. 1434–1445, 2024, doi: 10.1109/JSTARS.2023.3336926.

[17] A. Miroszewski *et al.*, "Detecting clouds in multispectral satellite images using quantum-kernel support vector machines," *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.*, vol. 16, p. 7601–7613, 2023, doi: 10.1109/JSTARS.2023.3304122.

[18] A.M. Wijata, A. Miroszewski, B.L. Saux, N. Longépé, B. Ruszczak, and J. Nalepa, "Detection of bare soil in hyperspectral images using quantum-kernel support vector machines," in *IGARSS 2024 – 2024 IEEE International Geoscience and Remote Sensing Symposium*, Jul. 2024, p. 817–822, doi: 10.1109/IGARSS53475.2024.10641442.

[19] M.K. Gupta, M. Beseda, and P. Gawron, "How quantum computing-friendly multispectral data can be?" in *IGARSS 2022 – 2022 IEEE International Geoscience and Remote Sensing Symposium*, 2022, pp. 4153–4156, doi: 10.1109/IGARSS46834.2022.9883676.

[20] J. Preskill, "Quantum Computing in the NISQ era and beyond," *Quantum*, vol. 2, p. 79, Aug. 2018, doi: 10.22331/q-2018-08-06-79.

[21] S. Boixo *et al.*, "Characterizing quantum supremacy in near-term devices," *Nat. Phys.*, vol. 14, no. 6, pp. 595–600, 2018, doi: 10.1038/s41567-018-0124-x.

[22] F. Arute *et al.*, "Quantum supremacy using a programmable superconducting processor," *Nature*, vol. 574, no. 7779, pp. 505–510, 2019, doi: 10.1038/s41586-019-1666-5.

[23] E. Farhi and H. Neven, "Classification with quantum neural networks on near term processors," 2018. [Online]. Available: https://arxiv.org/abs/1802.06002

[24] V. Havlíček *et al.*, "Supervised learning with quantum-enhanced feature spaces," *Nature*, vol. 567, no. 7747, pp. 209–212, 2019, doi: 10.1038/s41586-019-0980-2.

[25] M. Schuld and N. Killoran, "Quantum machine learning in feature hilbert spaces," *Phys. Rev. Lett.*, vol. 122, p. 040504, Feb 2019, doi: 10.1103/PhysRevLett.122.040504.

[26] H.Y. Huang *et al.*, "Power of data in quantum machine learning," *Nat. Commun.*, vol. 12, no. 1, p. 2631, 2021, doi: 10.1038/s41467-021-22539-9.

[27] A. Vali, S. Comai, and M. Matteucci, "Deep learning for land use and land cover classification based on hyperspectral and multispectral earth observation data: A review," *Remote Sens.*, vol. 12, no. 15, p. 2495, 2020, doi: 10.3390/rs12152495.

[28] S. Talukdar, P. Singha, S. Mahato, S. Pal, Y.-A. Liou, and A. Rahman, "Land-use land-cover classification by machine learning classifiers for satellite observations—a review," *Remote Sens.*, vol. 12, no. 7, p. 1135, 2020.

[29] P. Ghamisi *et al.*, "Advances in hyperspectral image and signal processing: A comprehensive overview of the state of the art," *IEEE Geosci. Remote Sens. Mag.*, vol. 5, no. 4, pp. 37–78, 2017, doi: 10.1109/MGRS.2017.2762087.

[30] M. Romaszewski, P. Głomb, and M. Cholewa, "Semi-supervised hyperspectral classification from a small number of training samples using a co-training approach," *ISPRS-J. Photogramm. Remote Sens.*, vol. 121, pp. 60–76, 2016.

[31] 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, doi: 10.1038/nature23474.

[32] V. Dunjko and H.J. Briegel, "Machine learning & artificial intelligence in the quantum domain: a review of recent progress," vol. 81, no. 7, p. 074001, 2018, doi: 10.1088/1361-6633/aab406.

[33] H.Y. Huang, R. Kueng, and J. Preskill, "Predicting many properties of a quantum system from very few measurements," *Nat. Phys.*, vol. 16, no. 10, pp. 1050–1057, 2020, doi: 10.1038/s41567-020-0932-7.

[34] S. Thanasilp, S. Wang, M. Cerezo, and Z. Holmes, "Exponential concentration in quantum kernel methods," *Nat. Commun.*, vol. 15, no. 1, p. 5200, Jun. 2024, doi: 10.1038/s41467-024-49287-w.

[35] A. Miroszewski, M.F. Asiani, J. Mielczarek, B.L. Saux, and J. Nalepa, "In search of quantum advantage: Estimating the number of shots in quantum kernel methods," no. arXiv:2407.15776, Jul. 2024. [Online]. Available: http://arxiv.org/abs/2407.15776

[36] H. F. Trotter, "On the product of semi-groups of operators," *Proc. Am. Math. Soc.*, vol. 10, no. 4, pp. 545–551, 1959.

[37] M. Suzuki, "Generalized trotter's formula and systematic approximants of exponential operators and inner derivations with applications to many-body problems," *Commun. Math. Phys.*, vol. 51, no. 2, pp. 183–190, 1976, doi: 10.1007/BF01609348.

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

[39] P. Ghamisi, J. Plaza, Y. Chen, J. Li, and A.J. Plaza, "Advanced spectral classifiers for hyperspectral images: A review," *IEEE Geosci. Remote Sens. Mag.*, vol. 5, no. 1, pp. 8–32, 2017.

[40] M. Berger, J. Moreno, J.A. Johannessen, P.F. Levelt, and R.F. Hanssen, "Esa's sentinel missions in support of earth system science," *Remote Sens. Environ.*, vol. 120, pp. 84–90, 2012, doi: 10.1016/j.rse.2011.07.023.

[41] M. Campos-Taberner *et al.*, "Understanding deep learning in land use classification based on sentinel-2 time series," *Sci. Rep.*, vol. 10, no. 1, p. 17188, 2020, doi: 10.1038/s41598-020-74215-5.

[42] A. Benavoli, G. Corani, J. Demšar, and M. Zaffalon, "Time for a change: a tutorial for comparing multiple classifiers through Bayesian analysis," *J Mach. Learn. Res.*, vol. 18, no. 1, pp. 2653–2688, 2017.