






Cloud-Based Analysis of Large-Scale Hyperspectral Imagery for Oil Spill Detection

Juan M. Haut , *Senior Member, IEEE*, Sergio Moreno-Alvarez , *Student Member, IEEE*,
Rafael Pastor-Vargas , *Senior Member, IEEE*, Ambar Perez-Garcia ,
and Mercedes E. Paoletti , *Senior Member, IEEE*

Abstract—Spectral indices are of fundamental importance in providing insights into the distinctive characteristics of oil spills, making them indispensable tools for effective action planning. The normalized difference oil index (NDOI) is a reliable metric and suitable for the detection of coastal oil spills, effectively leveraging the visible and near-infrared (VNIR) spectral bands offered by commercial sensors. The present study explores the calculation of NDOI with a primary focus on leveraging remotely sensed imagery with rich spectral data. This undertaking necessitates a robust infrastructure to handle and process large datasets, thereby demanding significant memory resources and ensuring scalability. To overcome these challenges, a novel cloud-based approach is proposed in this study to conduct the distributed implementation of the NDOI calculation. This approach offers an accessible and intuitive solution, empowering developers to harness the benefits of cloud platforms. The evaluation of the proposal is conducted by assessing its performance using the scene acquired by the airborne visible infrared imaging spectrometer (AVIRIS) sensor during the 2010 oil rig disaster in the Gulf of Mexico. The catastrophic nature of the event and the subsequent challenges underscore the importance of remote sensing (RS) in facilitating decision-making processes. In this context, cloud-based approaches have emerged as a prominent technological advancement in the RS field. The experimental results demonstrate noteworthy performance by the proposed

cloud-based approach and pave the path for future research for fast decision-making applications in scalable environments.

Index Terms—Cloud computing (CC), disaster monitoring, hyperspectral images (HSIs), remote sensing (RS), spectral indices.

I. INTRODUCTION

THE field of Earth observation (EO) has experienced significant advancements as a result of technological progress in remote sensing (RS) instruments. These instruments capture vast amounts of data in the optical domain, including hyperspectral images (HSIs) [1]. Particularly, HSI sensing instruments capture the solar radiation reflected by the ground materials, which exhibits variability across different wavelengths due to the atomic structure of the surface. Consequently, each material has a distinctive spectral signature associated with its specific combination of physical and chemical properties. The spectral signature is represented by each pixel in the form of an N -dimensional vector, with N corresponding to the number of bands [2]. The information enfolded is highly distinctive and valuable, given the local properties of every pixel [3]. Consequently, the volume of data is predominantly influenced by the spectral, spatial, and temporal resolutions [4]. Some examples where the application of HSI finds particular significance include environmental studies, disaster management [5], precision agriculture [6], and urban planning [7].

A. Toward Big RS Data Analysis

Regarding data acquisition, RS has experienced a notable surge in the accessibility of advanced hyperspectral missions, resulting in a continuous and large stream of data [8]. Several missions have been launched to capture valuable data for different applications. The airborne visible/infrared imaging spectrometer (AVIRIS) [9] is the most popular one, operated by the National Aeronautics and Space Administration (NASA) Jet Propulsion Laboratory. It measures up-welling spectral radiance across 224 contiguous bands. Its primary purpose is to identify, measure, and monitor various constituents present in the Earth's surface and atmosphere. AVIRIS operates at a data collection rate of 2.5 MB/s (approximately 9 GB/h), generating substantial data volumes during its flights. The Environmental Mapping and Analysis Program (EnMAP), a satellite mission led by the German Aerospace Center, includes a high-resolution hyperspectral imagery comprising 230 bands. It captures more than 20 TB/day

Manuscript received 17 August 2023; revised 23 October 2023 and 17 November 2023; accepted 1 December 2023. Date of publication 25 December 2023; date of current version 8 January 2024. This work was supported in part by Consejería de Economía, Ciencia y Agenda Digital of the Junta de Extremadura and the European Regional Development Fund (ERDF) of the European Union, under Grant GR21040, in part by the European Regional Development Fund (ERDF) of the European Union Interreg V-A Espana-Portugal (POCTEP) 2021-2027 program, under Grant 0206_RAT_EOS_PC_6_E, in part by the European Regional Development Fund (ERDF/FEDER) and by “Junta de Extremadura” under Grant IB20040, in part by the Ministry of Science and Innovation through the Project PID2019-110315RB-I00/AEI/10.13039/501100011033 (APRISA), and in part by the UNED. The work of Ámbar Pérez-García was supported in part by the “Agencia Canaria de Investigación, Innovación y Sociedad de la Información (ACIISI)” of the “Consejería de Economía, Conocimiento y Empleo”, and in part by the European Social Fund (FSE). (*Corresponding author: Rafael Pastor-Vargas.*)

Juan M. Haut and Mercedes E. Paoletti are with the Department of Technology of Computers and Communications, University of Extremadura, 10001 Cáceres, Spain (e-mail: juanmariohaut@unex.es; mpaoletti@unex.es).

Sergio Moreno-Alvarez is with the Department of Languages and Computer Systems, National University of Distance Education, 28040 Madrid, Spain (e-mail: smoreno@lsi.uned.es).

Rafael Pastor-Vargas is with the Department of Communication Systems and Control, National University of Distance Education, 28040 Madrid, Spain (e-mail: rpastor@scc.uned.es).

Ambar Perez-Garcia is with the Institute for Applied Microelectronics, University of Las Palmas de Gran Canaria, 35001 Las Palmas, Spain (e-mail: ambar.perez@ulpgc.es).

Code: <https://github.com/mhaut/scalable-ndoi>

Digital Object Identifier 10.1109/JSTARS.2023.3344022

of biophysical and chemical data for multiple scientific and environmental applications [10]. Similarly, the PRecursore IperSpettrale della Missione Applicativa (PRISMA), an Italian Space Agency mission, carries a hyperspectral sensor equipped with approximately 240 bands. Its objectives encompass forest monitoring, agricultural assessments, water, and soil analysis, as well as climate change and environmental research [11]. Looking ahead, new satellite missions are on the horizon, including the Copernicus Hyperspectral Imaging Mission for the Environment (CHIME), jointly designed by the European Union and the European Space Agency. CHIME aims to provide regular hyperspectral imaging to facilitate the monitoring of land cover changes and support sustainable agricultural practices [12].

This proliferation of information brings about the inherent benefits of data richness, enabling more detailed and comprehensive analysis. However, data abundance poses significant challenges in terms of the scalable and efficient processing of hyperspectral data [13]. To address these challenges, it becomes imperative to develop robust and effective strategies for handling the vast amounts of data generated by the aforementioned sensors. Such strategies should be designed not only to guarantee scalability, but also to aim for the optimal utilization of computational resources, ensuring reliable and timely extraction of valuable information. Specifically, and as part of the scope of this work, the utilization of HSIs assumes a pivotal role in the field of disaster management. This imagery offers valuable insights for identifying areas prone to natural disasters, including floods, landslides, and earthquakes [14], [15]. Also, it enables effective environmental monitoring, which facilitates the identification of changes in vegetation, water quality, and other crucial environmental factors [16]. In these contexts, the information can be instrumental in developing early warning systems that produce alerts to be disseminated to individuals, thereby mitigating potential disasters prior to their occurrence [17], [18]. Concretely, the management of ocean disasters, such as the monitoring and characterization of oil spills at sea, considers the interaction between electromagnetic waves and water based on the optical properties of water. The ability of remote sensors to capture responses is attributed to their sensitivity toward the presence, concentrations, and types of substances in the water [19]. Within this problematic area, the enhanced spectral and spatial resolutions facilitate the differentiation of false positives and accurate determination of oil types [20], [21]. Nevertheless, the high correlation between bands reduces the efficiency and accuracy of analysis and classification and poses new challenges [22]. A practical solution resides in selecting the most relevant bands for the study, thereby mitigating the storage and computational burdens while preserving essential information for analysis.

B. Analysing HSI Data: Use of Spectral Indices

Spectral indices are widely adopted as a technique to reduce the volume of data and facilitate feature detection. These spectral indices are combinations of bands obtained through simple arithmetic operations for highlighting particular spectrum patterns and suppressing the background. Therefore, each index employs

only a relatively narrow range of bands of interest to detect a particular phenomenon.

Numerous spectral indices that have been extensively tested in the literature, such as the fluorescence index (FI) [23], rotation-absorption index (RAI) [23], normalized difference oil index (NDOI) [24], hydrogen index (HI) [25], and oil slope index (OSI) [26]. In addition, some seawater and vegetation indices are also included, such as the water absorption feature (WAF) [27] and colored dissolved organic matter (CDOM) [28], or chlorophyll content (CHL) [29], normalized difference vegetation index (NDVI) [30], normalized difference water index (NDWI) [31], and ratio B_2/B_{11} [32]. Specifically, the NDOI is particularly noteworthy due to its ability to account for varying spill thicknesses, rendering it highly effective for coastal area spills, especially when suspended sand is involved. This distinctive characteristic sets it apart from the aforementioned alternatives, contributing to a better performance.

The performance of spectral indices, coupled with the increasing volume of HSI data generated from acquisition missions, and the inherent intricacies associated with calculating spectral indices, lead to a clear conclusion: a compelling requirement emerges to accelerate these computations, facilitating a fast and effective response to natural disasters. This poses new challenges in handling and processing RS data. Traditional data processing methods (i.e., MATLAB, R, or ENVI) have limited possibilities when dealing with such vast volumes of data [33]. Moreover, real-time applications are increasingly in demand [34], where the processing speed is essential [35]. To surmount these challenges, the utilization of highly scalable parallel and distributed architectures becomes indispensable. In this context, graphics processing units (GPUs) have notably improved the parallelization of computations. Nevertheless, processing extensive HSI data archives is still a research interest, given the limited by memory and availability constraints [36], [37]. Traditional computational methodologies, which include high-performance computing (HPC), cluster computing, and grid computing, provide viable solutions for processing large datasets. These clusters stand out as distributed systems composed of interconnected computing resources, such as GPUs, functioning collectively as an integrated resource [38]. Building upon this foundation, grid computing aims to establish an interconnected network of clusters that share computational and storage capacities. These approaches have undergone a thorough investigation concerning the processing of HSI data [39], [40].

C. Clouds Rise: Cloud-Computing for Mass Processing

In particular, cloud computing (CC) has emerged as a promising solution for distributed data processing due to its notable attributes, including scalability, cost-effectiveness, service-oriented architecture, high-performance capabilities, and on-demand resource provisioning [41]. The inherent attributes of CC offer clear advantages over traditional computing platforms. First, traditional platforms require significant upfront investments in hardware infrastructure. In contrast, cloud computing adopts a pay-as-you-go model with two major advantages: 1) users are billed only for the used resources, and 2) scalability is

achieved through on-demand access to a large pool of computing resources. The benefits include flexible and scalable resource accessibility. By contrast, traditional distributed computing has issues with accessibility and scalability since it depends on specialized hardware and more strict resource allocation models. This dynamic resource allocation may result in costs that are prohibitively high and jeopardize the central architecture, data management, and scalability. Second, CC excels in providing user-friendly interfaces that enable seamless interaction and resource provisioning through intuitive web-based interfaces. Third, the management of concurrent executions by multiple users without relying on a conventional job queue presents a significant advantage, which distinguishes CC from traditional computing methods.

These attributes, coupled with its cost-effectiveness and simplified provisioning mechanisms, have made CC a compelling option for hyperspectral data analysis [42], [43], [44]. Indeed, CC has a crucial and very positive role to play in facilitating effective decision-making for a wide range of applications based on hyperspectral data, especially in the context of risk monitoring and disaster management. The scalability of cloud platforms is especially remarkable in the era of Big Data, where on-demand resource models provide instant access to computing power and distributed processing techniques help reduce overall processing time. Recognizing the importance of sensitive data, cloud platforms incorporate strong security precautions such as data encryption, secure protocols, backups, redundancy, user identification, and secure data transmission. These protocols collectively strengthen the safeguarding of hyperspectral data, reinforcing the suitability of cloud platforms for applications that demand both efficiency and data integrity. Lastly, considerations extend to challenges such as the inefficient management of heterogeneous computational resources for land cover approaches [45], or interconnectivity and data dependency at sensitive times, especially in disaster monitoring [46].

Cloud platforms comprise multicore systems that facilitate the distribution and parallelization of computational tasks across numerous processing units. These platforms utilize distributed file systems to store, manage, and process large data, which simplifies access and analysis [47]. To facilitate appropriate interaction between data processing algorithms and large-scale datasets, the distribution of data and execution of operations on partitioned subsets across cloud machines have emerged as instrumental approaches [48]. This data management involves the formulation of a high-level programming language, which conceals the intricacies associated with directly engaging the distributed programming model, such as MapReduce. This is translated into data flow instructions to split the dataset and assign data segments to different nodes, ensuring efficiency, scalability and security with file systems such as Hadoop Distributed File System (HDFS) [49], [50]. Then, the framework runs processing algorithms on the distributed data, which supports parallel processing of computation-intensive tasks [51]. Also, users are enabled to customize algorithms and settings for data processing tasks using user-defined functions (UDFs).

To summarize, the cloud brings several benefits to RS, including scalability, elasticity, cost-efficiency, and reliability [52].

In this regard, the integration of CC into disaster management holds substantial promise for improving operational efficiency and facilitating prompt responses through real-time processing.

D. Contributions of the Research

This article presents a cloud-based approach for the detection of oil spills using the NDOI in HSIs [24]. The study focuses on the case of the Gulf of Mexico oil rig explosion in 2010, a significant incident that had profound environmental and economic consequences. The proposed approach leverages Apache Pig to incorporate UDFs into the data processing workflow. Additionally, HDFS is used for efficient distributed storage of the hyperspectral data. The implementation of the proposed approach is based on the MapReduce parallel model, which is utilized to leverage the capabilities of CC environments. By harnessing the parallel processing and distributed computing capabilities offered by the MapReduce model, the proposed approach aims to maximize the potential of CC environments for efficient and scalable data processing. This enables the effective utilization of resources and facilitates the execution of computationally intensive tasks in a distributed manner. This research highlights the advantages of utilizing sophisticated algorithms in RS applications. Furthermore, experimental results demonstrate the effectiveness of the proposed approach in handling large volumes of data efficiently, showcasing its scalability as the dataset size increases. As a summary, the contributions of this research are the following.

- 1) A novel distributed implementation of the NDOI tailored for the efficient processing of extensive datasets in the context of oil spill detection.
- 2) The establishment of a crucial link between the imperative need for robust and fast decision-making capabilities in sensitive scenarios and distributed computing.
- 3) A practical showcase of the seamless deployment of the proposal within cloud-based environments, facilitated by user-friendly frameworks.

The rest of this article is organized as follows. Section II sets out the previous work relevant to the current proposal. Section III outlines the distributed framework design that will be used in our implementation. Section IV presents the experimental validation and discussion. Finally, Section V concludes this article.

II. RELATED WORK

Cloud solutions excel in the efficient processing of remotely sensed data, outperforming alternative approaches, such as grid computing [40]. The flexibility and efficiency of CC have promoted the incorporation of traditional, artificial intelligence, and deep learning (DL) techniques for RS data processing. On the one side, parallel and distributed dimensionality reduction [53] and fast principal component analysis [54] have been developed on cloud architectures. Moreover, a noteworthy approach in the realm of HSI classification pertains to the utilization of scheduling meta-heuristics. This strategy aims to ensure an equitable and automated distribution of computational tasks across different CC resources [47].

In recent studies, significant attention has been given to the advancement of cloud-based adaptations of complex and computationally demanding algorithms. For instance, one notable contribution involves the adaptation of multinomial logistic regression [55] for HSI analysis in unmanned aerial vehicles (UAVs) using Apache Spark. The adaptation of multinomial logistic regression (MLR) in the cloud environment yields notable time reductions, demonstrating the potential for accelerated HSI analysis. It is crucial to highlight that the performance of the algorithm remains consistent. In addition, the K-means algorithm has also been adapted for Apache Spark [42]. This algorithm plays a crucial role in unsupervised clustering of HSI and represents a notable advancement in the literature by introducing a distributed framework for clustering massive hyperspectral datasets. Finally, novel algorithms have undergone adaptations to leverage such computational capabilities. Among these adaptations, a notable contribution is the implementation of auto-encoders (AE) in Apache Spark [43]. The cloud-based AE exhibits remarkable flexibility, enabling its application in several tasks, such as spectral unmixing. This represents a substantial milestone in the pursuit of complex deep models that consume extensive computational power and memory resources. Apache Spark presents a versatile and efficient data processing solution within a comprehensive ecosystem boasting substantial support and an extensive repository. In contrast, Apache Hive [56] prioritizes in-memory capabilities, particularly suited for real-time data processing scenarios. Such advancements hold significant implications, particularly in areas like environmental monitoring or decision-making.

Apache Pig emerges as a promising data flow approach to simplify large datasets processing. Leveraging the automatic parallelization and high-level data manipulation language provided by Apache Pig, users can effortlessly address complex tasks across large-scale clusters. For instance, Quirita et al. [57] proposed a distributed N-finder end-member extraction algorithm. Also, Apache Pig has been also implemented for specific classification algorithms, such as decision trees (DT) or random forest (RF) [58]. Furthermore, different methods have been proposed as alternatives for remotely sensed image processing, storage, and management. Specifically, these include a global-oriented spatial-temporal data simulation approach using OpenStack [59], a parallel content-based HSI retrieval system [60], an efficient segmentation model [61], a MapReduce method for image management [62], and an extension of Apache Hadoop for efficient data processing [63]. Finally, the data distribution offered by Apache Pig serves as a motivation to develop novel distributed solutions for hyperspectral data analysis, where spectral indices stand out as essential tools for disaster management. These indices use simple arithmetic operations to enhance specific spectral features while minimizing background interference. The strategic combination of bands not only accentuates pertinent information, but also addresses the challenges posed by the high dimensionality of hyperspectral data. Spectral indices rationalize this demand by focusing on the most relevant bands, which optimizes processing and resource utilization. In this context, CC emerges as a pivotal solution. CC environments provide the necessary resources for streamlined

processing and align seamlessly with the objective of managing and optimizing storage concerns related to high-dimensional spectral images.

III. SPECTRAL-INDEX FRAMEWORK FOR CLOUD STRATEGIES

Conventional data processing methods encounter challenges in the effective management of extensive data volumes, particularly in terms of computational demands and memory requirements. As aforementioned, the cloud has emerged as a promising approach to overcome these challenges effectively. The integration of CC with efficient distributed techniques presents a compelling solution for addressing these difficulties encountered in large-scale data analysis [55], [64], [65].

In this context, the proposed methodology contributes valuable insights into the successful integration of CC for HSI processing [57], [66]. The research explores crucial factors, architectural principles, and best practices that underpin this integration. Moreover, the study addresses not only technical considerations but also important aspects related to security, privacy, and data management. These considerations play a vital role in ensuring the safeguarding of sensitive data and compliance with regulatory requirements. Finally, this work includes guidelines that serve as a road-map to optimize data workflows and fully leverage the potential of cloud-based parallelization.

A. Architecture Design

The proposed method enables efficient and seamless interaction between data processing algorithms and large-scale datasets. The employed approach adheres to a fundamental methodology that entails the dissemination of data and the execution of operations on segregated subsets. These subsets are allocated to individual tasks, which are deployed across interconnected machines within the cloud infrastructure following a master-worker design. The framework encompasses several key components, which are introduced as follows.

1) *Application Definition*: This component encompasses the settings and preconditions required by the algorithms. The primary objective is to facilitate user-machine interaction through a friendly interface. Users are empowered to define algorithm specifications and configure various execution parameters, such as the number of nodes, dataset characteristics, or cores per node, among others. To ensure the ease of use and integration with subsequent components, the Java programming language is employed. This choice is based on the language simplicity and the ability to integrate with the overall framework. Furthermore, users have the flexibility to provide UDFs at this stage, enabling customization and extending the functionality of the algorithms.

2) *Storage and Data Management*: This module leverages the information provided by the preceding stage to perform the crucial tasks of specify the data flow and divide the dataset into slices. The data flows through the Pig framework, utilizing the Pig Latin language for efficient processing, while HDFS handles the storage system. Specifically, HDFS is designed to manage large datasets in a secure and cost-effective environment.

Each data slice is systematically assigned a unique identifier ranging from 1 to N , corresponding to the respective worker

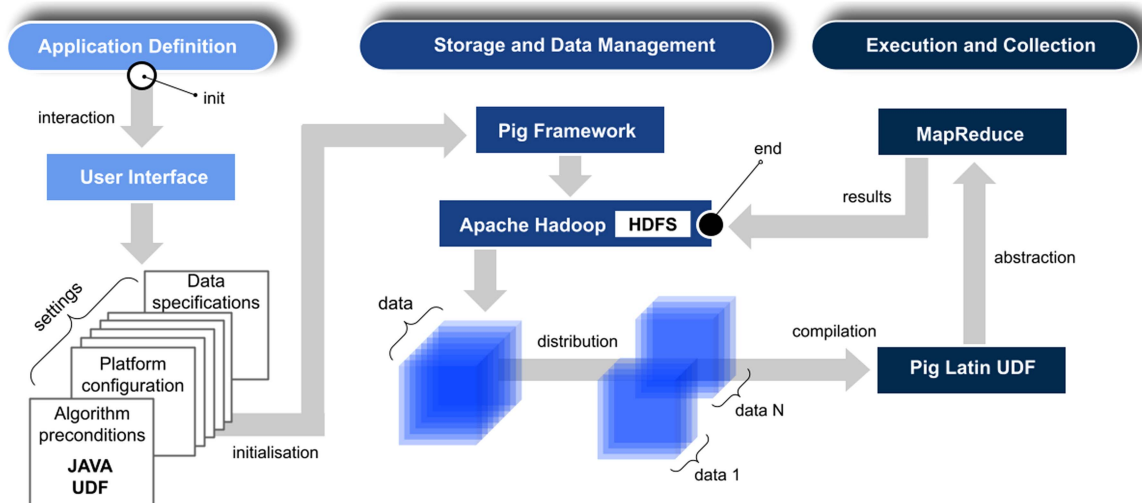


Fig. 1. Workflow diagram of the deployed CC framework, showcasing the primary components. The execution flow from the initial stage to the final point is depicted.

node responsible for its processing. This strategic allocation of data slices enables effective parallelization and distributed execution within the computational environment. Moreover, this step plays a pivotal role in ensuring the overall performance and scalability of the application. Also, critical considerations such as load balancing and data privacy are addressed.

3) *Execution and Collection*: This phase encompasses the distributed data processing implemented according to the assigned data in each worker. For this purpose, UDFs are translated into MapReduce jobs, facilitating efficient computation across the computational environment. At the end, the results are collected. The execution of these algorithms follows a well-defined pipeline, comprising distinct stages that compose the processing workflow.

In the subsequent sections, a thorough breakdown of the fundamental concepts that form the foundation of each individual component is provided. The architectural overview of the framework is illustrated in Fig. 1.

4) *Pig Framework*: Apache Pig [49] presents a highly efficient and streamlined approach for incorporating UDFs into the execution workflow. The framework is assisted by a compiler that enables the interpretation of Pig Latin scripts into optimized MapReduce jobs. The scripts provide the possibility to integrate external libraries using multiple languages, such as Java. This inherent versatility empowers users to tailor data transformations and operations to meet the specific requirements of a wide range of computationally demanding tasks. In addition, UDFs facilitate code reusability and modular-paradigm, enhancing collaboration and maintainability within the system.

Pig Latin provides a sequential structure in a nested data model, where each step corresponds to a unique data operation or transformation. The operations are performed using a straightforward and high-level programming paradigm, akin to SQL. Indeed, a range of operations, including filtering, grouping, joining, and aggregation, among others, are efficiently encapsulated in Pig Latin. Pig operates directly on files, and hence,

eliminates the need for costly data movements. Furthermore, it adopts a simple data model with a limited number of data types, including: 1) *atomics* for strings and numbers; 2) *tuples* for data sequences; and 3) *maps* as collections of key-value items. These operations greatly facilitate the compilation of MapReduce operations, streamlining the overall process. To further enhance usability, a debugging environment is provided, which proves indispensable when dealing with complex workflows.

As a consequence of the above, the Pig framework offers a comprehensive solution to address current challenges in distributed large-scale data processing, which is well-suited for rich spectral data.

5) *Apache Hadoop*: Hadoop is developed by the Apache Software Foundation¹ and presents a highly scalable implementation of the MapReduce operation. This powerful framework enables the efficient processing of vast datasets by distributing computational tasks across multiple workers, thereby leveraging the advantages of data distribution within a computationally efficient environment [67], [68]. To achieve this objective, Hadoop employs two critical components. First, the HDFS is utilized to ensure efficient data retrieval and storage. Second, the MapReduce model is employed for facilitating parallel computation. The integration of these components forms the foundation of the Hadoop capabilities to handle large-scale data processing tasks in a distributed computing environment.

The HDFS incorporates a meticulously designed partitioning and replication mechanism to efficiently distribute data among workers. This process enables the strategic allocation of data to workers with a higher likelihood of being mapped. Moreover, HDFS leverages the concurrency capabilities provided by a large-scale node infrastructure and effectively mitigates the impact of failures through data replication, ensuring both fault tolerance and data availability. Moreover, it includes block

¹Apache Software Foundation. (2010). Retrieved from <https://hadoop.apache.org>

duplication techniques that play a critical role in preserving data integrity and availability. The block-based approach involves partitioning blocks based on the overall data size by utilizing a hierarchical structure consisting of a central NameNode and multiple DataNodes. The NameNode works as a centralized repository for data files and essential metadata, and runs on the master node to control the distribution of metadata to DataNodes. Efficient replication of data blocks in memory is made possible through a well-established connection between the NameNode and DataNodes.

Although Hadoop provides a fault-tolerant environment, the data loading process into HDFS is often time-consuming and data-intensive. Consequently, several related issues have been extensively investigated in the literature. For instance, the study [69] proposes energy enhancements by the usage of two resource boxes: interactive or batch jobs. Alternatively, an altered version of Hadoop deployed within the cluster incorporates an efficient scheduler [70], which utilizes resources more efficiently.

6) *MapReduce Programming Model*: MapReduce is implemented as a purpose-built computational model designed to harness the capabilities of widely available hardware [50], [71]. The primary goal is to efficiently process massive volumes of data within a scalable environment. This is achieved through a two-step process: applying a map function to the relevant data blocks, followed by the reduction of results to obtain the desired output. The master node manages a significant dataset stored in HDFS, which is then subdivided into smaller subdatasets. Each of these subtasks is independently assigned to individual worker nodes for processing, and the results are subsequently aggregated. The workflow of MapReduce can be divided into five distinct stages: 1) *data reading*; 2) *mapping*; 3) *shuffling*; 4) *reducing*; 5) *output results*. This workflow facilitates the dissemination of data to workers during run-time through a projection process. An exemplary implementation of this model is Amazon Elastic MapReduce (EMR), which empowers users to instantiate Hadoop clusters for processing extensive datasets. Alternatively, resources such as Amazon Elastic Compute Cloud (EC2) and additional Amazon Web Services (AWS), ensure scalable processing capabilities.

MapReduce operates on discrete data blocks, enabling the parallel execution of operations within each block to expedite application processing. As previously introduced, HDFS works in combination with MapReduce providing data blocks to be distributed to accomplish reliable and extremely fast computations. Replication is implemented to ensure feasibility, where DataNodes balance the data in case of need to ensure a high replication. Also, the NameNode constantly tracks which blocks need to be replicated. In this context, parallelization takes place among various data blocks during both the Map and Reduce phases. This is also optimized through the use of key-value pairs assigned to MapReduce tasks, enabling the reduction of specific data before the completion of all tasks. Standard MapReduce frameworks consist of four fundamental concepts: 1) *workers*, 2) *job trackers*, 3) *task trackers* and 4) *task runners*. The job tracker initiates the job and partitions the tasks among workers. Each task is associated with a unique identifier that corresponds

Algorithm 1: Pig Latin framework outline for seamless integration into novel algorithms. The settings, libraries and UDF paths are represented as *CONFIG*, *LIBS*, and *UDF*, respectively. The implemented algorithm as $f(\cdot)$, dataset \mathcal{X} , outputs \mathcal{O} , p partition, and $k \in K$ iterations. Bold typing refers to Pig Latin instructions.

```

1: REGISTER CONFIG & LIBS & UDF ▷ Distribution
   and parallelization
2: DEFINE f(UDF)
3: LOAD  $\mathcal{X}$  from S3
4: FOR EACH  $\mathcal{X}_p \in \mathcal{X}$  GENERATE  $\mathcal{O}_p$ 
5: REDUCE  $\mathcal{O} \leftarrow \mathcal{O}_p$ 
6: STORE S3  $\leftarrow \mathcal{O}$ 
7:
8: if  $k \neq K$  then                                ▷Exit condition
9:     Repeat from step 5.
10: end if
11:
12: Obtain and validate results.

```

to a specific data partition and launched by a runner within a Java Virtual Machine (JVM). Also, the job trackers manages and updates in execution time the status of the workers, which are facilitated by the task tracker. This tracker periodically reports the progress of workers during the execution of tasks and retrieves the Java archive (JAR) file containing UDFs from the HDFS. Upon task completion and when a worker becomes idle, the job tracker dispatches a new task using a scheduling algorithm, ensuring optimal utilization of available resources. This iterative process continues until all required task slots are filled. Finally, the reduce task is executed by a task tracker to consolidate the results obtained from the different tasks.

B. NDOI: A Distributed Solution

The NDOI (η) has exhibited its proficiency in detecting oil spills in marine environments [24]. Specifically, the reflectance at wavelengths R_{599} and R_{870} has been identified as a crucial region of interest for this purpose. The computation of this index is determined as follows:

$$\eta = \frac{R_{599} - R_{870}}{R_{599} + R_{870}}.$$

At first glance, its calculation seems straightforward. However, it becomes increasingly intricate due to the substantial number of bands within HSI scenes. In this regard, the HSI data cube is denoted as $\mathcal{X} \in \mathbb{R}^{H \times W \times B}$, where H, W, B are the height, width, and bands, respectively. Moreover, the number of bands varies for each sensor, owing to their different properties. To effectively address the spatial-spectral information encompassed by the scene, the significance of memory requirements becomes paramount. The data require partitioning across multiple computing nodes or workers, which perform computations proportional to their assigned data portions to calculate η . In this context, the proposed procedure harnesses

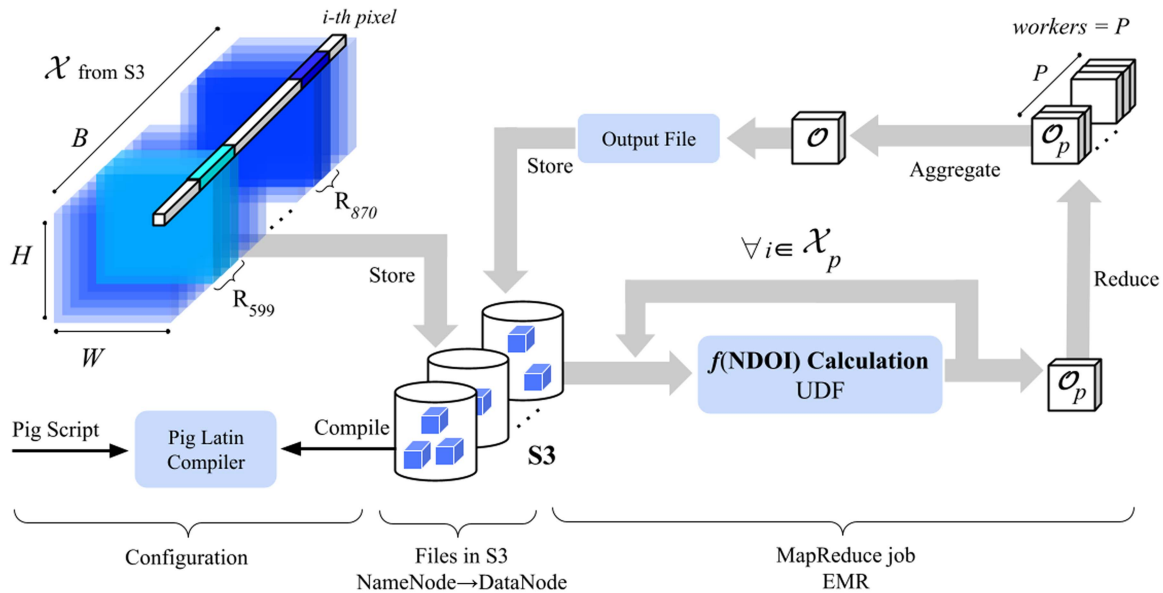


Fig. 2. Execution workflow of the proposed CC architecture, based on Apache Hadoop and Pig. The wavelengths required for the calculation of η are highlighted, with R_{599} represented by the light blue color and R_{870} represented by the dark blue color.

cloud-based methodologies that align with the architectural design introduced previously. The ensuing procedure is as follows.

The initial step involves configuring the cloud environment, which includes setting up Amazon EC2 instances, Amazon S3 storage, HSI data \mathcal{X} , scripts, and the UDF (η). The entirety of the data is stored in HDFS along with the necessary files for the workers tasks. Subsequently, the Pig script is compiled, and then, the execution starts. MapReduce parallelizes the calculations of η by initially mapping the partitioned data \mathcal{X}_p among the workers, where $p \in P$ denotes the number of partitions corresponding to the number of workers. For each data partition, its i th pixel is utilized for calculating η using the corresponding wavelengths, thereby generating a worker output \mathcal{O}_p . Concretely, the total number of pixel is $H \times W$. Ultimately, the individual outputs are aggregated, resulting in the generation of \mathcal{O} , which is subsequently stored in HDFS. The complete procedure is illustrated in Fig. 2.

C. Integration Guidelines

The employed architecture is specifically engineered to enhance the utilization of innovative and diverse data-intensive processing applications. One key aspect contributing to its effectiveness is the incorporation of UDFs. Therefore, integrating this framework into new methodologies requires a structured set of steps. First, configure a cloud resource provider. Next, develop UDFs tailored to the specific algorithm and the corresponding configurations. Lastly, deploy the combined Apache Hadoop and Pig environment.

The procedure is outlined in Algorithm 1. The initial phase involves the distribution and parallelization of data, which is realized through the implementation of the Pig Latin script. Significant emphasis should be placed on recognizing that each

subset of data is different. Subsequently, each worker independently processes its assigned data subset. Then, the final results are stored in the Amazon S3 storage. These individual outputs are then combined to evaluate the outcomes of the implemented algorithm, which may be encapsulated in a JAR file. Finally, a termination condition is assessed, based on factors such as reaching an iteration limit, achieving the desired results, or other predetermined criteria.

IV. EXPERIMENTAL RESULTS

A. Environment and Platform Details

Experiments are conducted on AWS, where S3 is used to store the data, the UDFs and all the required libraries. Also, EMR² is used to manage the Hadoop framework, to distribute and process the data between the cluster nodes, which is dynamically built using EC2 instances. This makes it possible to deploy configurable Hadoop/Spark clusters based on the available tools and the execution nodes capabilities [72]. Worker nodes are m5.xlarge³ EC2 instances, each equipped with 16 GB of RAM, 4 virtual CPUs, and an Intel Xeon Scalable processor running at a maximum clock speed of 3.1 GHz (Skylake 8175 M or Cascade Lake 8259CL). Data files are stored in S3 and then distributed among workers for parallel processing. Each node incorporates a 32 GB volume of Elastic Block Store (EBS/gp2) type, resulting in a total solid state drive (SSD) size of $P \times 32$. The networking configuration boasts a 10 Gbps network bandwidth and 4750 Mbps EBS bandwidth.

The UDF function (η calculation) is integrated within a PIG script, with the objective of using the native capabilities of Hadoop for the parallelization of operations based on the

²[Online]. Available: <https://aws.amazon.com/es/emr/>

³[Online]. Available: <https://aws.amazon.com/es/ec2/instance-types/m5/>

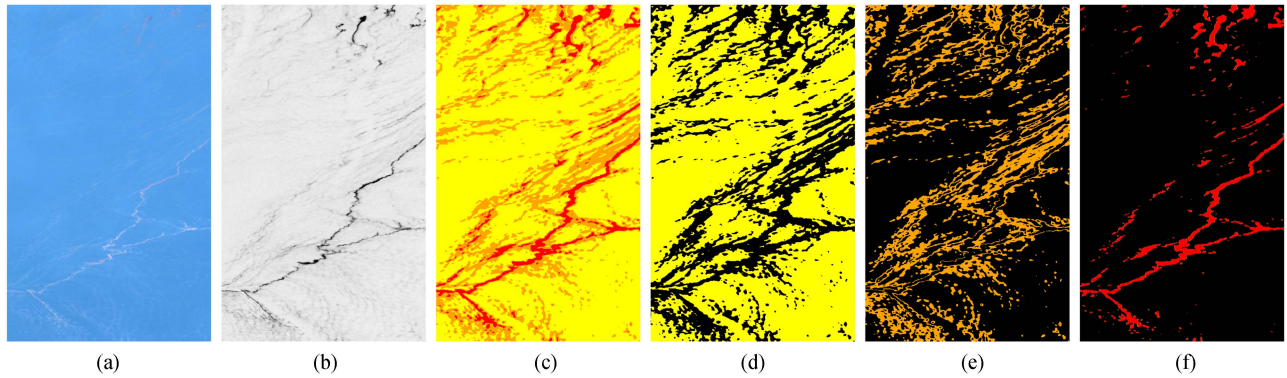


Fig. 3. Visualization of the dataset, captured by AVIRIS on May 17, 2010 (f100517t01p00r11). The colors observed in (b) indicate the intensity of oil within a specific pixel, ranging from white (representing water) to black, with complete whiteness indicating the absence of oil in that pixel. (a) RGB. (b) NDOI (η). (c) Spill map. (d) Thin spill. (e) Medium spill. (f) Thick spill.

MapReduce paradigm [73]. The used environment includes several applications, such as Hadoop 3.3.3, HBase 2.4.15, Hive 3.1.3, Hue 4.10.0, Oozie 5.2.1, Pig 0.17.0, Sqoop 1.4.7, Tez 0.10.2, and ZooKeeper 3.5.10.

To run the experiments, the execution script is designed and loaded on the master node. The script incorporates the essential code and input data, which are stored within the S3 service. Therefore, the target UDF (η) is executed on each cluster, which consists of a predefined set of computing nodes. The experiments are conducted using a range of input file sizes. For each combination of experiments, involving different cluster configurations sizes and input file sizes, five Monte Carlo executions are performed. The time evaluation includes the average values obtained from these five executions. The overall execution of the experiments spanned approximately 30 hours, incurring a cost of 0.214 USD/h per equivalent normalized EC2 instance hour of execution in the cluster.

In order to analyze the performance of the η calculation, a profiling option is incorporated. Each conducted experiment records the number of invocations to the UDF, which directly relies on the file size, along with the corresponding execution time in microseconds on the nodes. While these specific data are not considered in the current dataset analysis, they offer valuable insights, particularly in confirming the linearity of the η calculation regardless of the input data loading and output data storage operations in S3.

B. Dataset Description

The target scene corresponds to the Deepwater Horizon (DWH) mission in the Gulf of Mexico, situated southeast of the Mississippi River Delta. In May 2010, the imagery was acquired utilizing the AVIRIS sensor [74]. Acquired images possess a spatial resolution ranging from 20 to 1 m, which compress hyperspectral data across the visible and near-infrared (VNIR) and short-wave infrared (SWIR) spectral ranges, spanning from 366 to 2496 nm, with a lower bandwidth of 10 nm. The scene encompasses 224 bands from an altitude of approximately 28 000 feet.

The scene shown in Fig. 3(a) [33] exposes diverse magnitudes of oil slicks, distinguished by multiple luminous lines traversing the lower portion of the image, suggestive of emulsification. Moreover, conspicuous dark crimson rounded marks in the upper region of the image denote substantial oil blemishes. The η result, shown in Fig. 3(b), highlights contaminated areas, appearing more prominently in areas of thicker oil. This oil spill occurred within the geographical coordinates of 26° to 30° north and 84° to 92° west. In order to conduct a thorough analysis, the attention is exclusively directed toward the region of utmost significance within the image. Concretely, the classification map illustrated in Fig. 3(c) [33], serves to differentiate between multiple levels of oil spill thickness. This classification yields three distinct categories. The thick spill shown in Fig. 3(d) highlights the presence of emulsion lines traversing the scene. In addition, it shows the perception of black spots in the higher section of the image. In the second category, the medium spill is shown in Fig. 3(e), where a grainy pattern that resembles irregular spots is evident. Finally, the fine spill from Fig. 3(f) reveals the existence of thin films of oil, commonly referred to as sheen, on the water surface.

In order to perform extensive testing of memory and computational resources on the cloud platform, the entire scene is replicated to achieve the desired file size. This approach allows for an in-depth examination of the cloud performance and its ability to effectively handle varying data sizes, simulating real-world scenarios where data volumes may vary significantly.

C. Experimental Discussion

The proposed cloud-based methodology has undergone rigorous evaluation through three different experiments, aimed at comprehensively assessing its performance, efficiency, and scalability.

In this context, the size of the dataset plays a critical role in determining the efficiency and scalability of the proposed approach. The first experiment aims to investigate the impact of file size on the performance of the cloud-based methodology, concretely for 1, 4, 8, 16, and 32 GBs. For this purpose, multiple workers configuration (1, 2, 4, 8, 16, and 32 workers) are used

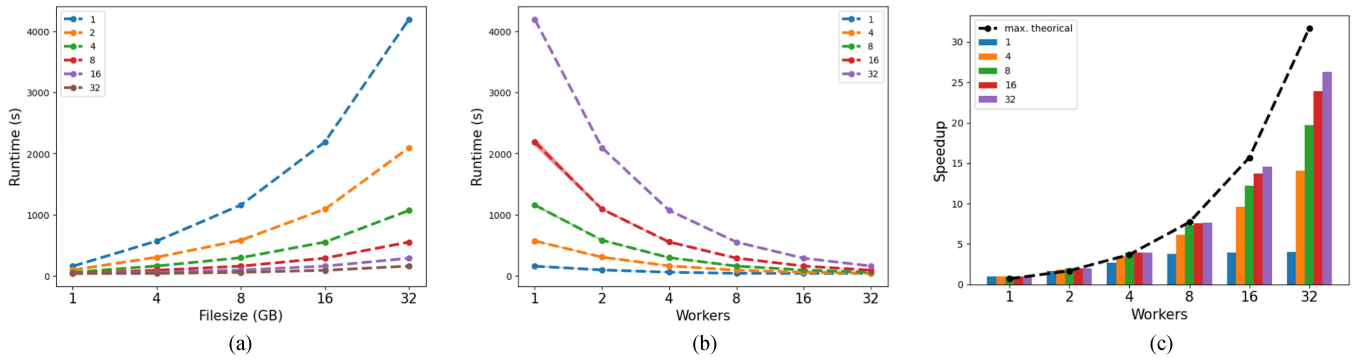


Fig. 4. Experimental results for the AVIRIS dataset encompassing various file size and workers configurations are presented. All three experiments are displayed in ascending order from left to right. Standard deviations are depicted in the plots and indicated with a small line positioned at the top of the bars.

to evaluate the obtained run-time for each configuration. In this regard, as the file size increases, a corresponding increase in the computational resources required to process the data emerges. This, in turn, produces longer execution times and potentially impact the environment responsiveness. Moreover, an understanding of the proposed method's performance across different file sizes of upmost importance in identifying potential bottlenecks or scalability issues. This is particularly relevant in real-world applications, where data amounts are often considerable. Such insights can facilitate the optimization of the proposed method performance and its applicability to practical scenarios.

The second experiment is designed to ascertain the elapsed time necessary for completing the execution. This metric holds significant importance for measuring the system's overall efficiency. In this context, the run-time serves as an indicator of possible bottlenecks or inefficiencies within the environment. Addressing these issues becomes imperative to enhance the methodology overall performance. Thus, this experiment plays a crucial role in providing valuable perspectives for the refinement and optimization of the cloud-based methodology.

The third experiment emphasizes classification to identify oil spill areas in the original image, pixel by pixel, and distinguish them from watery regions. In this regard, the evaluation considers different spectral indices, such as, HI, RAI, NDWI, and NDOI (η). Four classifiers are employed for such evaluation: support vector machine (SVM), random forest (RF), k -nearest neighbors (KNN), and multinomial logistic regression (MLR). Therefore, this experiment holds significant importance as it demonstrates the classification performance of the η against multiple indices.

The last experiment focuses on the speedup, a crucial metric to evaluate the degree of improvement achieved in the scalability performance. Speedup quantifies the advantage gained by parallelize computations across multiple workers (in a similar way to the previous experiment). Two key insights arise from this metric. First, a speedup greater than 1 signifies that the cloud-based approach surpasses traditional computing. Therefore, this result underscores the benefits of harnessing distributed computing resources. Secondly, a speedup value close to or equal to the

number of cloud nodes employed indicates near-linear scalability, which demonstrates the proposed method proficiency in efficiently handling larger workloads.

1) *Experimentation on File Size:* The results obtained from the first experiment are depicted in Fig. 4(a). Notably, when executing with a baseline configuration of a single worker, the required run-time exhibits a nearly exponential growth as the file size increases. As a consequence, the worker becomes saturated, leading to a substantial increase in run-time. This observation underscores the limitations of nondistributed approaches in terms of performance. As the number of workers increases, the required run-time significantly decreases. For instance, employing two workers results in approximately half the time required compared to using only one worker. Similar trends are observed with subsequent configurations. This clearly demonstrates that the distribution and parallelization of the workload aim to proportionally reduce the run-time.

Another important observation is that, for the same number of workers, the execution time increases with larger file sizes. Finally, for a large number of workers, such as 16 or 32, the run-time reaches a minimum. This minimum ensures a proficient execution of the workers. From these findings, it can be inferred that when the required execution time begins to exhibit linear growth, for example, with four workers for a 16 GB workload, the addition of more workers could be considered. Particularly, this fact assumes particular importance when the maintenance of a stable run-time becomes paramount in the application. This determination could be crucial in scenarios where the calculation of the η value is critical, as the reduction in execution time enables the implementation of real-time considerations. Furthermore, it should be observed that when considering a memory capacity of 32 GBs, the introduction of additional workers leads to a greater decrease in run-time up to a specific threshold (with 16 and 32 workers). However, beyond this threshold, the incorporation of new nodes ceases to yield significant advantages.

Overall, these results provide valuable observations into the relationship between the number of workers, file size, and execution time. Such ideas guide the optimization and decision-making processes in designing efficient and responsive cloud-based methodologies.

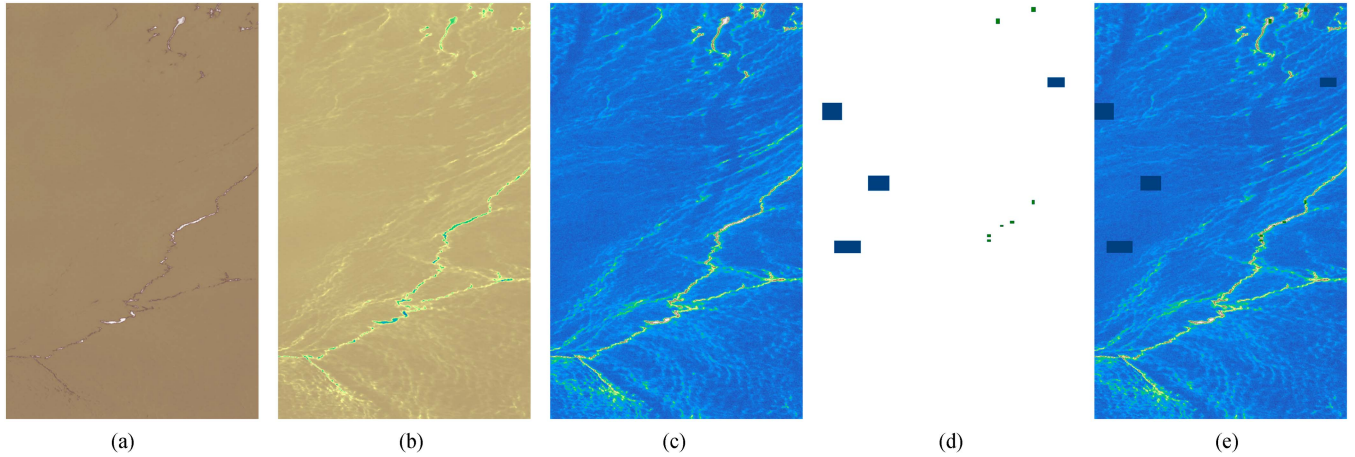


Fig. 5. Experimental showcase for classification purposes. The (a) and (b) images show the intermediate results of the η calculation final result (c). In (d) and (e), the set of pixels colored in blue represents water regions, while the green determines oil spill zones. (a) $R_{599} - R_{870}$. (b) $R_{599} + R_{870}$. (c) NDOI (η). (d) Labels. (e) Overlapping.

TABLE I
TIMES IN SECONDS (S) OBTAINED FOR ALL WORKER AND FILE SIZE CONFIGURATIONS

File Size (GBs)	Number of workers					
	1	2	4	8	16	32
1	157.76	95.96	59.03	41.46	40.25	39.50
4	570.01	304.98	161.65	92.54	59.59	40.56
8	1156.96	581.05	296.43	158.71	94.77	58.73
16	2188.27	1089.22	551.66	289.66	159.36	91.41
32	4194.52	2094.09	1068.21	551.01	287.68	159.76

2) *Experimentation on Run-Times*: The second experiment studies the evolution of run-time when varying numbers of workers, as depicted in Fig. 4(b). To comprehensively assess the performance, four different configurations were executed based on the file sizes from the previous experiment.

The results clearly demonstrate that higher file sizes correspond to longer execution times. Additionally, a notable trend is observed wherein an increase in the number of workers leads to a considerable reduction in run-time, up until the maximum number of workers, i.e., 32, is reached. At this worker configuration, the execution time becomes nearly consistent for all file sizes. Furthermore, this indicates the continued effectiveness of parallelization. The consistent execution time observed across various file sizes when employing the maximum worker configuration suggests the potential for enhanced utilization of computational resources with larger datasets. Also, this insight implies that the proposed cloud-based methodology can effectively scale with the available resources, accommodating larger datasets without compromising performance. However, when contemplating a smaller file size, such as 1 GB, the execution time maintains a relatively consistent pattern even with increased distribution, such as deploying more than eight workers. This phenomenon occurs due to the detrimental effects of partitioning data into excessively small sizes. In such instances, the overhead associated with partitioning surpasses the advantages of parallelization with a substantial idle time for computational resources. Consequently,

the time reduction achieved in this scenario is minimal. The results are shown in Table I.

The insights of this experiment provide valuable insights to optimize the proposed cloud-based methodology by strategically allocating the right number of workers based on file size and computational requirements.

3) *Experimentation on Classification*: The performance of the η calculation is contingent upon the classification of each pixel within the dataset. Moreover, different zones within the original image are specifically chosen for classification. Fig. 5 shows the concern of this experiment by manually labeling. Also, the intermediate results to obtain the final classification provided by the η calculation are included.

The outcomes of the classification are presented in Table II. A comparative analysis is conducted against various established methods in the literature for oil split detection. Notably, the results obtained from the proposed η calculation surpass the performance of the other indices. The best results are obtained with the KNN classifier. In addition, the classification maps are provided in Fig. 6. RAI achieves significant results by detecting not only thick spills, as in the case of NDWI, but also medium and thin spills, where the oil dispersion in the water is visible. However, the η manages to improve the classification of fine-grain oil splits within the entire image, for instance, multiple pixels in the upper left corner.

It is important to highlight that the simplicity of the proposed calculation, relying on only two spectral bands, underscores the outstanding quality of the achieved results.

4) *Experimentation on Speedup*: This last experiment focuses on determining the speedup achieved by the different worker configurations for all file sizes. Results are included in Fig. 4(c). As observed, the maximum theoretical speedup is determined by the number of workers. Although this value is theoretical, the obtained speedup value closely approximates this theoretical maximum. Consequently, starting with a single worker, no distribution occurs and the speedup serves as the baseline. As additional worker configurations are introduced,

TABLE II
CLASSIFICATIONS RESULTS FOR DISCRIMINATING BETWEEN WATER AND OIL CLASSES; THE OVERALL ACCURACY (OA), AVERAGE ACCURACY (AA), AND KAPPA (Kx100) METRICS ARE PRESENTED; BEST RESULTS FOR EACH CLASSIFIER ARE HIGHLIGHTED IN BOLD

	Support Vector Machine				Random Forest				K-Nearest Neighbours				Multinomial Logistic Regression			
	HI	RAI	NDWI	NDOI (η)	HI	RAI	NDWI	NDOI (η)	HI	RAI	NDWI	NDOI (η)	HI	RAI	NDWI	NDOI (η)
water	99.75±0.143	99.99±0.022	99.85±0.335	100.0±0.0	100.0±0.0	99.96±0.061	99.94±0.037	99.98±0.025	100.0±0.0	99.98±0.025	99.86±0.057	99.93±0.116	99.98±0.025	99.93±0.039	99.83±0.188	99.93±0.078
oil	76.15±1.425	77.14±3.156	89.47±0.804	93.92±1.078	66.94±0.543	75.82±34.286	88.0±0.286	96.22±0.286	69.24±0.286	75.16±34.667	88.49±0.567	97.37±0.066	69.9±0.281	77.96±32.476	91.78±0.985	93.75±1.887
OA	98.19±0.039	98.48±2.189	98.98±0.292	99.6±0.07	97.82±0.038	98.37±2.246	99.15±0.02	99.73±0.017	97.97±0.017	98.34±2.279	99.11±0.048	99.73±0.07	97.99±0.033	98.48±2.113	99.27±0.134	99.52±0.083
AA	87.95±0.641	88.56±16.575	94.56±0.306	96.96±0.541	83.47±0.274	87.9±17.136	93.97±0.126	98.1±0.135	84.62±0.143	87.57±17.324	94.18±0.27	98.63±0.274	84.94±0.149	88.95±16.216	95.79±0.424	96.84±0.916
K(x100)	83.85±0.056	81.25±28.745	91.56±2.159	96.64±0.615	79.09±0.406	79.66±30.471	92.75±0.163	97.76±0.153	80.79±0.208	79.09±31.24	92.44±0.4	97.79±0.554	81.09±0.336	81.66±27.449	93.96±0.99	96.02±0.72

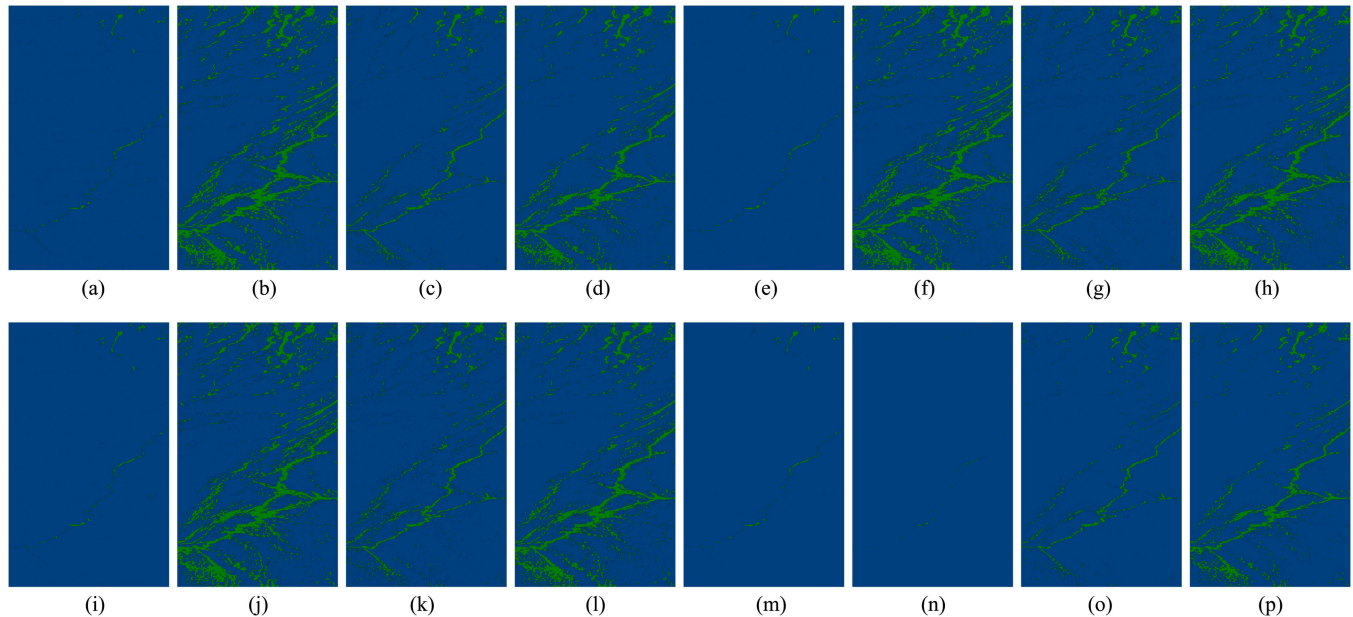


Fig. 6. Visual representation of classification maps illustrating water (blue) and oil (green) categories for various spectral indices. (a) HI-SVM. (b) RAI-SVM. (c) NDWI-SVM. (d) NDOI(η)-SVM. (e) HI-RF. (f) RAI-RF. (g) NDWI-RF. (h) NDOI(η)-RF. (i) HI-KNN. (j) RAI-KNN. (k) NDWI-KNN. (l) NDOI(η)-KNN. (m) HI-MLR. (n) RAI-MLR. (o) NDWI-MLR. (p) NDOI(η)-MLR.

the speedup value increases significantly. However, noteworthy implications emerge from this metric.

First, when the file size does not necessitate a high number of workers, the speedup value stabilizes even with an increment in the number of workers. For instance, with eight workers, the obtained speedup values for file sizes of 8 and 16 GBs are remarkably similar. This observation leads to the conclusion that the usage of a higher number of workers may not be necessary in such cases, as it may not result in a substantial increase in speed. The cloud environment configuration offers the capability to adapt executions to these requirements.

Second, the discrepancy between the speedup theoretical value and the obtained value for 32 workers configuration indicates that workers can effectively handle more data in this scenario. However, it also suggests that the computational resources of these workers may not be fully exploited. This analysis highlights the potential for further optimization to achieve enhanced resource utilization. Given this insight, it becomes feasible to ascertain the tradeoff between parallelization, resources, and economic considerations. Undoubtedly, the observed ascending trend in positive speedup with the usage of 32 workers and 32 GBs provides evidence supporting the notion that the distribution achieved is optimal, with a maximum speedup obtained of 26.26. The speedup values are included in Table III.

TABLE III
SPEEDUP VALUES OBTAINED FOR ALL WORKER AND FILE SIZE CONFIGURATIONS

File Size (GBs)	Number of workers					
	1	2	4	8	16	32
1	1.0	1.64	2.67	3.8	3.92	3.99
4	1.0	1.87	3.53	6.16	9.57	14.05
8	1.0	1.99	3.90	7.29	12.21	19.70
16	1.0	2.01	3.97	7.55	13.73	23.94
32	1.0	3.93	7.61	7.61	14.58	26.26

As a summary, the complete set of experiments contribute significantly to the understanding of the cloud-based methodology performance. The evaluation of the experiments demonstrates that the proposed cloud implementation achieves noteworthy results in all cases.

V. CONCLUSION

This article introduces a new cloud-based implementation of the NDOI (η) calculation, serving as an effective spectral index for discriminating oil spills through CC-based HSI data analysis. Leveraging the inherent advantages of cloud platforms, the proposed methodology addresses the considerable memory and

computational demands entailed in processing extensive HSI scenes. The adoption of the Pig framework offers an easy-to-use architecture, facilitating the distributed deployment of the NDOI algorithm. Furthermore, harnessing Amazon services, such as EC2 and S3, increases the scalability of our mechanism and provides a well-suited platform for future endeavors encompassing complex scenes and the implementation of intricate algorithms. The obtained experimental results demonstrate a high degree of scalability, with both the number of workers and the file size exhibiting notable reductions in run-times. The efficient performance observed during the experiments serves as a promising indicator of the proposed methodology proficiency and capacity to accommodate increasingly demanding workloads. The ultimate outcome highlights the considerable advantage of utilizing cloud implementations for real-time disaster response.

As limitations to consider, the scalability of the η calculation is tied to the memory requirements for dataset storage. In our particular case, attempts to speed up the calculations have proved negligible, given the simplicity of these calculations. However, as we move toward the processing of large data sets, there are two critical challenges to be tackled: the efficient storage of gargantuan datasets and the intricacies associated with distributed processing. Large hyperspectral datasets, with their multitude of spectral bands, offer an ideal environment for assessing the potential of cloud-based technologies when handling these drawbacks. Additionally, the majority of capturing sensors are capable of detecting the target wavelengths for the NDOI calculation, which contributes to the reliability of the suggested index computation.

In our forthcoming research, we plan to explore the implementation of new algorithms tailored to resource-intensive computations and memory-demanding tasks. Moreover, deploying these algorithms in a real-time environment is identified as a future target for disaster monitoring and decision-making tasks. These efforts will broaden the scope of RS applications, catering to a variety of needs and driving the field forward in environmental monitoring and disaster response scenarios. In light of this, it is imperative to deploy a dedicated cloud-based system for real-time disaster management. Future work involves introducing additional methodologies utilizing machine learning algorithms to predict the spread of different oil splits. Such approach aims to offer valuable insights for anticipating and implementing solutions to minimize future damage.

REFERENCES

- [1] W. Emery and A. Camps, *Introduction to Satellite Remote Sensing: Atmosphere, Ocean, Land and Cryosphere Applications*. Amsterdam, The Netherlands: Elsevier, 2017.
- [2] J. L. García, M. E. Paoletti, L. I. Jiménez, J. M. Haut, and A. Plaza, "Efficient semantic segmentation of hyperspectral images using adaptable rectangular convolution," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, pp. 1–5, 2022.
- [3] S. Li et al., "Geospatial Big Data handling theory and methods: A review and research challenges," *ISPRS J. Photogrammetry Remote Sens.*, vol. 115, pp. 119–133, 2016.
- [4] A. Dasgupta, "Big data: The future is in analytics," *Geospatial World*, vol. 3, no. 9, pp. 28–36, 2013.
- [5] K. Thangavel et al., "Autonomous satellite wildfire detection using hyperspectral imagery and neural networks: A case study on Australian wildfire," *Remote Sens.*, vol. 15, no. 3, 2023, Art. no. 720.
- [6] S. Shitharth, H. Manoharan, A. M. Alshareef, A. Yafoz, H. Alkhir, and O. M. Mirza, "Hyper spectral image classifications for monitoring harvests in agriculture using fly optimization algorithm," *Comput. Elect. Eng.*, vol. 103, 2022, Art. no. 108400.
- [7] A. Nisha and A. Anitha, "Current advances in hyperspectral remote sensing in urban planning," in *Proc. 3rd Int. Conf. Intell. Comput. Instrum. Control Technol.*, 2022, pp. 94–98.
- [8] J. M. Moleró, A. Paz, E. M. Garzón, J. A. Martínez, A. Plaza, and I. García, "Fast anomaly detection in hyperspectral images with RX method on heterogeneous clusters," *J. Supercomputing*, vol. 58, pp. 411–419, 2011.
- [9] Jet Propulsion Laboratory, "Aviris - airborne visible / infrared imaging spectrometer," 2021. Accessed: Aug. 17, 2021. [Online]. Available: <https://aviris.jpl.nasa.gov/index.html>
- [10] R. Müller et al., "The EnMAP hyperspectral satellite mission. An overview and selected concepts," in *Proc. 3rd Annu. Hyperspectral Imag. Conf.*, 2012, pp. 39–44.
- [11] S. Pignatti et al., "The PRISMA hyperspectral mission: Science activities and opportunities for agriculture and land monitoring," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2013, pp. 4558–4561.
- [12] J. Nieke and M. Rast, "Status: Copernicus hyperspectral imaging mission for the environment (CHIME)," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2019, pp. 4609–4611.
- [13] A. Plaza, J. Plaza, A. Paz, and S. Sanchez, "Parallel hyperspectral image and signal processing [applications corner]," *IEEE Signal Process. Mag.*, vol. 28, no. 3, pp. 119–126, May 2011.
- [14] S. Chabrilat, A. F. Goetz, L. Krosley, and H. W. Olsen, "Use of hyperspectral images in the identification and mapping of expansive clay soils and the role of spatial resolution," *Remote Sens. Environ.*, vol. 82, no. 2/3, pp. 431–445, 2002.
- [15] V. Bondur, "Modern approaches to processing large hyperspectral and multispectral aerospace data flows," *Izvestiya, Atmospheric Ocean. Phys.*, vol. 50, pp. 840–852, 2014.
- [16] N. Chehata, C. Orny, S. Boukir, D. Guyon, and J. Wigneron, "Object-based change detection in wind storm-damaged forest using high-resolution multispectral images," *Int. J. Remote Sens.*, vol. 35, no. 13, pp. 4758–4777, 2014.
- [17] N. Casagli et al., "Spaceborne, UAV and ground-based remote sensing techniques for landslide mapping, monitoring and early warning," *Geoenvironmental Disasters*, vol. 4, pp. 1–23, 2017.
- [18] M. N. Jha, J. Levy, and Y. Gao, "Advances in remote sensing for oil spill disaster management: State-of-the-art sensors technology for oil spill surveillance," *Sensors*, vol. 8, no. 1, pp. 236–255, 2008.
- [19] S. Hafeez et al., "Detection and monitoring of marine pollution using remote sensing technologies," in *Monitoring of Marine Pollution*, H. B. Fouzia, Ed., London, U.K.: IntechOpen, Sep. 2019. [Online]. Available: <https://ideas.repec.org/h/ito/pchaps/163515.html>
- [20] M. Wettle, P. J. Daniel, G. A. Logan, and M. Thankappan, "Assessing the effect of hydrocarbon oil type and thickness on a remote sensing signal: A sensitivity study based on the optical properties of two different oil types and the hymap and quickbird sensors," *Remote Sens. Environ.*, vol. 113, no. 9, pp. 2000–2010, 2009. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0034425709001680>
- [21] I. Leifer et al., "State of the art satellite and airborne marine oil spill remote sensing: Application to the bp deepwater horizon oil spill," *Remote Sens. Environ.*, vol. 124, pp. 185–209, 2012. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0034425712001563>
- [22] B. Zhang, L. Zhao, and X. Zhang, "Three-dimensional convolutional neural network model for tree species classification using airborne hyperspectral images," *Remote Sens. Environ.*, vol. 247, 2020, Art. no. 111938. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0034425720303084>
- [23] E. Loos, L. Brown, G. Borstad, T. Mudge, and M. Álvarez, "Characterization of oil slicks at sea using remote sensing techniques," in *Proc. Oceans*, 2012, pp. 1–4.
- [24] Á. Pérez-García, P. Horstrand, and J. F. López, "Ndoi, a novel oil spectral index: Comparisons and results," in *Proc. 12th Workshop Hyperspectral Imag. Signal Process.: Evol. Remote Sens.*, 2022, pp. 1–5.
- [25] F. K. C. author, K. Oppermann, and B. Hörig, "Hydrocarbon index—an algorithm for hyperspectral detection of hydrocarbons," *Int. J. Remote Sens.*, vol. 25, no. 12, pp. 2467–2473, 2004, doi: [10.1080/01431160310001642287](https://doi.org/10.1080/01431160310001642287).

- [26] Q. Li, L. Lu, B. Zhang, and Q. Tong, "Oil slope index: An algorithm for crude oil spill detection with imaging spectroscopy," in *Proc. Second Int. Workshop Earth Observation Remote Sens. Appl.*, 2012, pp. 46–49.
- [27] W.-Z. Lu, H. Yuan, and G. Xu, *Modern Near Infrared Spectroscopy Analytical Technology*. Beijing, China: China Petrochemical Press, 2007.
- [28] T. Kutser, D. C. Pierson, K. Y. Kallio, A. Reinart, and S. Sobek, "Mapping lake CDOM by satellite remote sensing," *Remote Sens. Environ.*, vol. 94, no. 4, pp. 535–540, 2005. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0034425704003670>
- [29] C. Hu, Z. Lee, and B. Franz, "Chlorophyll algorithms for oligotrophic oceans: A novel approach based on three-band reflectance difference," *J. Geophysical Res.: Oceans*, vol. 117, no. C1, pp. 1–25, 2012.
- [30] D. A. Carlson and T. N. Ripley, "On the relation between NDVI, fractional vegetation cover, and leaf area index," *Remote Sens. Environ.*, vol. 62, no. 3, pp. 241–252, 1997.
- [31] B. c. Gao, "NDWI—a normalized difference water index for remote sensing of vegetation liquid water from space," *Remote Sens. Environ.*, vol. 58, no. 3, pp. 257–266, 1996. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0034425796000673>
- [32] P. Kolokoussis and V. Karathanassi, "Oil spill detection and mapping using sentinel 2 imagery," *J. Mar. Sci. Eng.*, vol. 6, no. 1, 2018, Art. no. 4. [Online]. Available: <https://www.mdpi.com/2077-1312/6/1/4>
- [33] Á. Pérez-García, M. E. Paoletti, J. M. Haut, and J. F. López, "Novel spectral loss function for unsupervised hyperspectral image segmentation," *IEEE Geosci. Remote Sens. Lett.*, vol. 20, pp. 1–5, 2023.
- [34] B. Zhang et al., "Progress and challenges in intelligent remote sensing satellite systems," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 15, pp. 1814–1822, 2022.
- [35] Z. Zakria, J. Deng, R. Kumar, M. S. Khokhar, J. Cai, and J. Kumar, "Multiscale and direction target detecting in remote sensing images via modified YOLO-v4," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 15, pp. 1039–1048, 2022.
- [36] C. González, S. Sánchez, A. Paz, J. Resano, D. Mozos, and A. Plaza, "Use of FPGA or GPU-based architectures for remotely sensed hyperspectral image processing," *Integration*, vol. 46, no. 2, pp. 89–103, 2013. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0167926012000223>
- [37] S. Bernabe, S. Lopez, A. Plaza, and R. Sarmiento, "GPU implementation of an automatic target detection and classification algorithm for hyperspectral image analysis," *IEEE Geosci. Remote Sens. Lett.*, vol. 10, no. 2, pp. 221–225, Mar. 2013.
- [38] M. Zakarya and L. Gillam, "Energy efficient computing, clusters, grids and clouds: A taxonomy and survey," *Sustain. Comput.: Inform. Syst.*, vol. 14, pp. 13–33, 2017.
- [39] G. Aloisio and M. Cafaro, "A dynamic earth observation system," *Parallel Comput.*, vol. 29, no. 10, pp. 1357–1362, 2003. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0167819103001078>
- [40] J. M. Haut, M. E. Paoletti, S. Moreno-Álvarez, J. Plaza, J.-A. Rico-Gallego, and A. Plaza, "Distributed deep learning for remote sensing data interpretation," *Proc. IEEE*, vol. 109, no. 8, pp. 1320–1349, Aug. 2021.
- [41] M. Cafaro and G. Aloisio, *Grids, Clouds, and Virtualization*. Berlin, Germany: Springer, 2011.
- [42] J. M. Haut, M. Paoletti, J. Plaza, and A. Plaza, "Cloud implementation of the K-means algorithm for hyperspectral image analysis," *J. Supercomputing*, vol. 73, pp. 514–529, 2017.
- [43] J. M. Haut et al., "Cloud deep networks for hyperspectral image analysis," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 12, pp. 9832–9848, Dec. 2019.
- [44] E. Dogo, A. Salami, and S. Salman, "Feasibility analysis of critical factors affecting cloud computing in Nigeria," *Int. J. Cloud Comput. Serv. Sci.*, vol. 2, no. 4, 2013, Art. no. 276.
- [45] S. Moreno-Álvarez, M. E. Paoletti, G. Cavallaro, J. A. Rico, and J. M. Haut, "Remote sensing image classification using CNNs with balanced gradient for distributed heterogeneous computing," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, pp. 1–5, 2022.
- [46] G. Li, J. Zhao, V. Murray, C. Song, and L. Zhang, "Gap analysis on open data interconnectivity for disaster risk research," *Geo-spatial Inf. Sci.*, vol. 22, no. 1, pp. 45–58, 2019, doi: [10.1080/10095020.2018.1560056](https://doi.org/10.1080/10095020.2018.1560056).
- [47] Z. Wu et al., "Scheduling-guided automatic processing of massive hyperspectral image classification on cloud computing architectures," *IEEE Trans. Cybern.*, vol. 51, no. 7, pp. 3588–3601, Jul. 2021.
- [48] A. Fernández et al., "Big data with cloud computing: An insight on the computing environment, mapreduce, and programming frameworks," *Wiley Interdiscipl. Rev.: Data Mining Knowl. Discov.*, vol. 4, no. 5, pp. 380–409, 2014.
- [49] C. Olston, B. Reed, U. Srivastava, R. Kumar, and A. Tomkins, "Pig latin: A not-so-foreign language for data processing," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2008, pp. 1099–1110.
- [50] T. White, *Hadoop: The Definitive Guide*. Sebastopol, CA, USA: O'Reilly Media, Inc., 2012.
- [51] J. Dean and S. Ghemawat, "Mapreduce: Simplified data processing on large clusters," *Commun. ACM*, vol. 51, no. 1, pp. 107–113, 2008.
- [52] M. Carroll, A. V. D. Merwe, and P. Kotze, "Secure cloud computing: Benefits, risks and controls," in *Proc. Inf. Secur. South Afr.*, 2011, pp. 1–9.
- [53] Z. Wu, Y. Li, A. Plaza, J. Li, F. Xiao, and Z. Wei, "Parallel and distributed dimensionality reduction of hyperspectral data on cloud computing architectures," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 9, no. 6, pp. 2270–2278, Jun. 2016.
- [54] Y. Li, Z. Wu, J. Wei, A. Plaza, J. Li, and Z. Wei, "Fast principal component analysis for hyperspectral imaging based on cloud computing," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2015, pp. 513–516.
- [55] J. M. Haut and M. E. Paoletti, "Cloud implementation of multinomial logistic regression for UAV hyperspectral images," *IEEE J. Miniaturization Air Space Syst.*, vol. 1, no. 3, pp. 163–171, Dec. 2020.
- [56] J. Camacho-Rodríguez et al., "Apache hive: From mapreduce to enterprise-grade Big Data warehousing," in *Proc. Int. Conf. Manage. Data*, 2019, pp. 1773–1786, doi: [10.1145/3299869.3314045](https://doi.org/10.1145/3299869.3314045).
- [57] V. A. A. Quirita, G. A. O. P. d. Costa, and C. Beltrán, "A distributed N-FINDER cloud computing-based solution for endmembers extraction on large-scale hyperspectral remote sensing data," *Remote Sens.*, vol. 14, no. 9, 2022, Art. no. 2153.
- [58] V. A. Ayma et al., "Classification algorithms for Big Data analysis, a map-reduce approach," *Int. Arch. Photogrammetry, Remote Sens. Spatial Inf. Sci.*, vol. XL-3/W2, pp. 17–21, 2015. [Online]. Available: <https://isprs-archives.copernicus.org/articles/XL-3-W2/17/2015/>
- [59] X. Yao et al., "Enabling the big earth observation data via cloud computing and DGGs: Opportunities and challenges," *Remote Sens.*, vol. 12, no. 1, 2019, Art. no. 62.
- [60] P. Zheng et al., "A parallel unmixing-based content retrieval system for distributed hyperspectral imagery repository on cloud computing platforms," *Remote Sens.*, vol. 13, no. 2, 2021, Art. no. 176.
- [61] P. N. Happ, R. S. Ferreira, G. A. Costa, R. Q. Feitosa, C. Bentes, and P. Gamba, "Towards distributed region growing image segmentation based on mapreduce," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2015, pp. 4352–4355.
- [62] Y. Liu, B. Chen, W. He, and Y. Fang, "Massive image data management using HBase and MapReduce," in *Proc. 21st Int. Conf. Geoinformatics*, 2013, pp. 1–5.
- [63] P. Bajcsy, P. Nguyen, A. Vandecreme, and M. Brady, "Spatial computations over terabyte-sized images on Hadoop platforms," in *Proc. IEEE Int. Conf. Big Data*, 2014, pp. 816–824.
- [64] A. M. Burgueño et al., "Scalable approach for high-resolution land cover: A case study in the Mediterranean basin," *J. Big Data*, vol. 10, no. 1, pp. 1–22, 2023.
- [65] S. Iranpak, A. Shahbahrani, and H. Shakeri, "Remote patient monitoring and classifying using the Internet of Things platform combined with cloud computing," *J. Big Data*, vol. 8, pp. 1–22, 2021.
- [66] V. A. A. Quirita et al., "A new cloud computing architecture for the classification of remote sensing data," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 10, no. 2, pp. 409–416, Feb. 2017.
- [67] M. D. Assunção, R. N. Calheiros, S. Bianchi, M. A. Netto, and R. Buyya, "Big data computing and clouds: Trends and future directions," *J. Parallel Distrib. Comput.*, vol. 79, pp. 3–15, 2015.
- [68] S. R. Zeebaree, H. M. Shukur, L. M. Haji, R. R. Zebari, K. Jacksi, and S. M. Abas, "Characteristics and analysis of Hadoop distributed systems," *Technol. Rep. Kansai Univ.*, vol. 62, no. 4, pp. 1555–1564, 2020.
- [69] Y. Chen, S. Alspaugh, D. Borthakur, and R. Katz, "Energy efficiency for large-scale MapReduce workloads with significant interactive analysis," in *Proc. 7th ACM Eur. Conf. Comput. Syst.*, 2012, pp. 43–56.
- [70] J. Wolf et al., "FLEX: A slot allocation scheduling optimizer for MapReduce workloads," in *Proc. ACM/IFIP/USENIX 11th Int. Conf. Distrib. Syst. Platforms Open Distrib. Process.*, 2010, pp. 1–20.
- [71] H. Herodotou and S. Babu, "Profiling, what-if analysis, and cost-based optimization of MapReduce programs," *Proc. VLDB Endowment*, vol. 4, no. 11, pp. 1111–1122, Aug. 2011, doi: [10.14778/3402707.3402746](https://doi.org/10.14778/3402707.3402746).
- [72] A. Zarei, S. Safari, M. Ahmadi, and F. Mardukhi, "Past, present and future of Hadoop: A survey," 2022, [arXiv:2202.13293](https://arxiv.org/abs/2202.13293).

- [73] P. Natesan, V. E. Satishkumar, S. K. Mathivanan, M. Venkatesan, P. Jayagopal, and S. M. Allayear, "A distributed framework for predictive analytics using Big Data and MapReduce parallel programming," *Math. Problems Eng.*, vol. 2023, 2023, Art. no. 6048891.
- [74] G. Vane, R. O. Green, T. G. Chrien, H. T. Enmark, E. G. Hansen, and W. M. Porter, "The airborne visible/infrared imaging spectrometer (AVIRIS)," *Remote Sens. Environ.*, vol. 44, no. 2/3, pp. 127–143, 1993.



Juan M. Haut (Senior Member, IEEE) received the B.Sc. and M.Sc. degrees in computer engineering in 2011 and 2014, respectively, and the Ph.D. degree in information technology in 2019 from the University of Extremadura, Cáceres, Spain, supported by an University Teacher Training Programme from the Spanish Ministry of Education.

He is currently a Professor with the Department of Computers and Communications, University of Extremadura. Also, he is a Member of the Hyperspectral Computing Laboratory (HyperComp), Department of

Technology of Computers and Communications, University of Extremadura. He has authored/coauthored more than 65 JCR journal articles (more than 30 in IEEE journals) and more than 30 peer-reviewed conference proceeding papers. His research interests include remote sensing data processing and high-dimensional data analysis, applying machine (deep) learning, and cloud computing approaches.

Dr. Haut was the recipient of the Outstanding Ph.D. Award at the University of Extremadura in 2019. Some of his contributions have been recognized as hot-topic publications for their impact on the scientific community. Also, he was a recipient of the Outstanding Paper Award in the 2019 and 2021 IEEE WHISPERS conferences. From his experience as a reviewer, it is worth mentioning his active collaboration in more than ten scientific journals, such as IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING, *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, and IEEE GEOSCIENCE AND REMOTE SENSING LETTERS, and he has been awarded with the Best Reviewer recognition of the *IEEE Geoscience and Remote Sensing Letters* and IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING in 2018 and 2020, respectively. Furthermore, he has guest-edited three special issues on hyperspectral remote sensing for different journals. He is also an Associate Editor of the *IEEE Geoscience and Remote Sensing Letters* and *IEEE Journal on Miniaturization for Air and Space Systems*.



Sergio Moreno-Alvarez (Student Member, IEEE) received the B.Sc. and M.Sc. degrees in computer engineering and the Ph.D. degree in computer technologies from the University of Extremadura, Badajoz, Spain, in 2017, 2019, and 2022, respectively.

Currently, he is a Member and Professor with the Department of Computer Systems and Languages, National University of Distance Education (UNED), Madrid, Spain. He has participated as author/coauthor in more than 15 JCR journal articles and multiple presentations at international and national conferences.

His research interests include high performance computing (HPC), deep learning and its optimization through parallelism approaches and remote sensing (RS).

Dr. Alvarez has served as Reviewer in multiple IEEE journals, including IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING, *IEEE Geoscience and Remote Sensing Letters*, IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTATIONAL INTELLIGENCE or *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, among others.



Rafael Pastor-Vargas (Senior Member, IEEE) received the M.Sc. degree in physics from Complutense University, Madrid, Spain, in 1994, and the Ph.D. degree in computer science from the UNED, Madrid, Spain, in 2006.

He is currently an Associate Professor with the Control and Communication Systems Department, UNED. He is currently the Dean with the Computer Science Engineering Faculty. He was an innovation Manager with the Innovation and Development Centre, UNED, from 2004 to 2009, and the General Manager for incorporating innovative services with UNED's learning model from 2009 to 2011. He has participated in a big number of projects financed in public calls, some of them with special relevance for companies and administrations with an international scope, international journals, and conferences. He teaches graduate and postgraduate courses related to network interconnection, security domains, and cloud computing, among others. His research interests include quality of service support in distributed systems, applied machine learning algorithms, and the development of infrastructure and algorithms for e-learning, remote sensing and cybersecurity, particularly on the Cloud.

Dr. Pastor-Vargas belongs to the Blockchain local group of the IEEE Spain Section and is also vocal in the IEEE Education Society (Spanish Chapter).



Ambar Perez-Garcia was born in Las Palmas de Gran Canaria, Spain, in 1997. She received the B.Sc. degree in physics from the University of La Laguna (ULL), La Laguna, Spain, in 2019 and the M.Sc. degree in remote sensing from the University of Valencia (UV), Valencia, Spain, in 2020, and the M.Sc. degree in education from the University of La Rioja (UNIR), La Rioja, Spain, in 2021. She is currently working toward the Ph.D. degree in telecommunications technologies with the University of Las Palmas de Gran Canaria (ULPGC), Las Palmas de Gran Canaria, Spain.

In 2023, she did a research stay on neural network segmentation with HYPERCOMP, University of Extremadura, Badajoz, Spain. She was also with HWM, Wageningen University & Research, The Netherlands, spectrally characterizing plastics. She is currently with the University Institute for Applied Microelectronics, IUMA, University of Las Palmas de Gran Canaria, Las Palmas de Gran Canaria, Spain. Her research interests include remote sensing, hyperspectral image processing, artificial intelligence, and the detection of marine litter.



Mercedes E. Paoletti (Senior Member, IEEE) received the B.Sc., M.Sc. degrees in computers engineering and Ph.D. degrees in computer technologies from the University of Extremadura, Cáceres, Spain, in 2014, 2016, and 2020, respectively.

She is currently an Associate Professor with the Mérida University Centre, Department of Technology of Computers and Communications, University of Extremadura and a Member of the Hyperspectral Computing Laboratory (HyperComp). Her research interests include remote sensing analysis through

deep learning models and high performance computing.

Dr. Paoletti was the recipient of the 2019 Outstanding Paper Award recognition in the IEEE WHISPERS 2019 conference, the Outstanding Ph.D. Award at the University of Extremadura in 2020, and the research award from the Spanish Scientific Society of Computer Science and the BBVA Foundation in 2022. She has been a reviewer for different journals and congresses, including the *IEEE TGRS* and *IEEE GRSL*, an activity for which she has been recognized as best reviewer in 2019 and 2020. Also, she has guest edited four special issues on hyperspectral data and deep learning analysis for Remote Sensing, and served as Topical Associated Editor of the *IEEE TGRS* between 2021 and 2023.