

Notice of Violation of IEEE Publication Principles

“Reversible Data Hiding and Smart Multimedia Computing Using Big Data in Remote Sensing Systems”

by Rahul Malik, Aditya Khamparia, Sahil Garg, Deepak Gupta, Bong Jun Choi, M. Shamim Hossain

in IEEE Access, Vol 8, August 2020, pp. 153546- 153560

After careful and considered review of the content and authorship of this paper by a duly constituted expert committee, this paper has been found to be in violation of IEEE’s Publication Principles.

This paper is a duplication of the original text from the paper cited below. The original text was copied without attribution (including appropriate references to the original author(s) and/or paper title) and without permission.

“Big Data Geospatial Processing for Massive Aerial LiDAR Datasets”

by David Deibe, Margarita Amor, and Ramón Doallo

in Remote Sensing, MDPI, February 2020

Received August 5, 2020, accepted August 9, 2020, date of publication August 20, 2020, date of current version September 1, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3018326

Reversible Data Hiding and Smart Multimedia Computing Using Big Data in Remote Sensing Systems

RAHUL MALIK¹, (Member, IEEE), ADITYA KHAMPARIA¹,
SAHIL GARG², (Member, IEEE), DEEPAK GUPTA³, (Member, IEEE),
BONG JUN CHOI⁴, (Senior Member, IEEE), AND
M. SHAMIM HOSSAIN⁵, (Senior Member, IEEE)

¹School of Computer Science and Engineering, Lovely Professional University, Phagwara 144411, India

²Electrical Engineering Department, École de technologie supérieure, Montréal, QC H3C 1K3, Canada

³Department of Computer Science Engineering, Maharaja Agrasen Institute of Technology, New Delhi 110086, India

⁴School of Computer Science and Engineering, Soongsil University, Seoul 06978, South Korea

⁵Department of Software Engineering, College of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia

Corresponding authors: M. Shamim Hossain (mshossain@ksu.edu.sa) and Bong Jun Choi (davidchoi@soongsil.ac.kr)

This work was supported in part by the National Research Foundation (NRF) funded by the Ministry of Science and ICT (MSIT), South Korea, under Grant 2019R1C1C1007277, and in part by the Researchers Supporting Project number (RSP-2020/32), King Saud University, Riyadh, Saudi Arabia.

ABSTRACT There is a need for improvement of tools to deal with large volumes of multimedia data effectively. In particular, real-time data processing is one of the major problems for multimedia data computing in remote sensing systems. Such big data systems have to offer effective management and computational efficiency for applications in real-time. In this paper, we propose a large-scale geological processing method for aerial Light Detection and Ranging (LiDAR) clouds containing multimedia data that ensures mobility and timeliness. By utilizing Spark and Cassandra, our proposed approach can significantly reduce the execution time of the time-consuming process. We investigate fast ground-only raster generation from huge LiDAR datasets. We observed that filtered cloud data ensuing from impartial consideration of neighboring zones could lead to classification errors on the boundaries. Therefore, an integrated approach is proposed to correct these errors to improve the classification consistency, achieve faster processing time, provide automatic error correction, obtain Digital Terrain Models (DTM), and minimize user intervention. These features can provide a framework for an on-demand DTM output and scalable application services. Furthermore, the proposed approach can expect to benefit other real-time applications in LiDAR systems.

INDEX TERMS Data analysis, digital terrain model, geospatial, LiDAR, multimedia.

I. INTRODUCTION

Reversible Data hiding (RDH) in photographs imperceptibly embeds a secret message in a protected Light Detection and Ranging (LiDAR) image. The original LiDAR image can be fully restored from the encrypted marked image after extraction. Using this feature, RDH could be utilized in various fields like military, healthcare image processing, and forensics – wherever lossless restoration of these initial inputs is needed. Several RDH approaches have been proposed in the past. These RDH approaches can be categorized into lossless compaction, difference development, and histogram evaluation [1]. In lossless compaction, geometric redundancy of LiDAR images is employed to preserve secret information

[2], [3], [53]. In others, information is embedded by increasing the contrasts between two neighboring pixels [4], [5]. In histogram modification methods, information integration is attained by altering the frequency distribution of the LiDAR image [6]–[8].

Currently, LiDAR engineering is amongst the most helpful geoinformatics programs. Additionally, it features extensive benefits across many technical areas, such as agroforestry and area surveillance [9]. Nevertheless, LiDAR is frequently viewed as a barrier in developing efficient systems to control the vast amounts of information that the system needs to collect and process. Thus, researchers are often required to consider different methods [10], [11].

The introduction of big data has provided new sensing devices and computing solutions [12], [54], [55]. The data processing in geospatial awareness has been discussed

The associate editor coordinating the review of this manuscript and approving it for publication was Zhihan Lv⁶.

in the literature [13]–[16]. Also, using big data in LiDAR-connected fields is discussed in [17]–[19].

Recent versions of the geographic information system (GIS) use LiDAR point cloud components, such as three-dimensional (3D) raster data, for geospatial methods, e.g., biomass estimation [20] and linear element extraction [21]. A GIS is a versatile and robust system for geographic information processing. In comparison, DSMs (Digital Surface Models) and Digital Terrain Models (DTMs) can be used for visual comparisons contrasting the consistency of the different techniques and processes [22].

The effectiveness of the system is strongly dependent on the LiDAR's ability to classify the category of dirt and ground types, which usually is time-consuming. For places where there is an abundant amount of data for collecting and processing, like GIS centers, data classification is a huge concern [23]. In consideration of the massive amount of information being managed, the use of conventional computing methods has limitations when undertaking complex computational tasks, such as poor scalability, low flexibility, high latency, and low-efficiency [24].

Therefore, we propose a large-scale geological processing method for large aerial LiDAR clouds [25], which contain multimedia data, that ensures mobility and timeliness. We consider fast ground-only rasters to create DTMs in large clouds. The system can facilitate the setup of any form of geospatial operation and support real-time processing of aerial LiDAR clouds by providing improved reliability, effectiveness, scalability, mobility, and timeliness, in comparison with single computer methods. Based on the information is sent out using Cassandra [30]. The simplicity and accessibility of its source code and batch-oriented architecture make it possible to perform the unit dissemination with Spark [31].

The main contribution of the proposed work is highlighted as follows:

- We propose a large-scale data hiding method for multimedia data (LiDAR images) to ensure timeliness and mobility.
- The proposed LiDAR encryption method performs data extraction and boundary error correction.
- Information is sent through Cassandra, and unit dissemination is performed using Spark architecture for fast processing.
- The effectiveness of the SC-091-12 algorithm was drastically enhanced using the transfer domain concept of data-hiding.
- The proposed method fixes the classification errors in the boundaries of adjacent zones.

In sum, an automated technique is presented in this paper to fix filtered rasters in large scale clouds and improve the quality of accomplished DTMs, minimizing human intervention [29].

The rest of the article is structured as follows. Section 2 presents a large-scale strategy for geospatial analysis. Section 3 presents the proposed data hiding, data extraction,

and error correction methods. Section 4 provides a performance evaluation of quality, applicability, and DTM system. Section 5 concludes the paper with findings and further research.

II. FLEXIBLE BIG DATA APPROACH TO GEOSPATIAL ANALYSIS

We focus primarily on distance, value, and velocity from the 5 Vs (volume, velocity, variety, veracity, and value) because our goal is to deliver a beneficial output (surge of land points). The use of storage technologies necessitates the ability to extract information, offer resistance to faults, and ensure good quality [30]. We aim to create a scalable geospatial processing system via distributed computing. Here, we assume that LiDAR datasets are scattered through N different nodes, which can execute complicated geospatial processes while reducing the deployment time [31].

The proposed system supports computing on both a single LiDAR cloud and multiple LiDAR clouds. The research focus has been to offer higher freedom with support for various geospatial processes [32]. We consider DTM fast-from-big aerial stage clouds because DTMs are among the most useful data programs obtained from the LiDAR dataset. The geospatial application library includes progressive morphological filtering for geographical DTM processing (subsection 2.A), a distributed storage to support LiDAR datasets (subsection 2.B), and a batch processing framework (subsection 2.C).

A. GEOGRAPHICAL DTM PROCESSING

DTMs are obtained in rasters from the cloud in the form of points which are further processed using morphological filters to form terrains. As a consequence, a raster is obtained that contains only ground points, representing the final DTM. Note that the image data is represented in terms of points [60]. So, the output result is constructed by the triangulation of these points [34]. Here, a simple algorithm, called SC-091-12 [35], is selected to perform this task. The reasons for this particular choice were (1) the consistency of output of the SC-091-12 is better than that of the Fusion [28], [36], (2) due to the development of DTM [26], and (3) its association with Spark and Cassandra, as the algorithm is programmed with Java which is a programming language promoted by Spark and Cassandra.

The accuracy of SC-091-12 in point classification depends on the selection of different parameters and more detail on the SC-091-12 along with its parameters is available in [35]. In [37], the authors considered large values for all the parameters. However, the CS details were customized in conjunction with all the attributes and the stage of clouds being handled, such as in the seaside, agricultural, mixed, or flat areas.

The point difference is not explicitly relevant to the entire input cloud, as discussed above, but to some raster. Our proposal divides the raster into a grid of regular cells, the dimensions of which are driven by the CSinput-parameter, which is probably the lowest 'Z' fundamental point in each unit [34]. This process improves the final accuracy of DTMs'

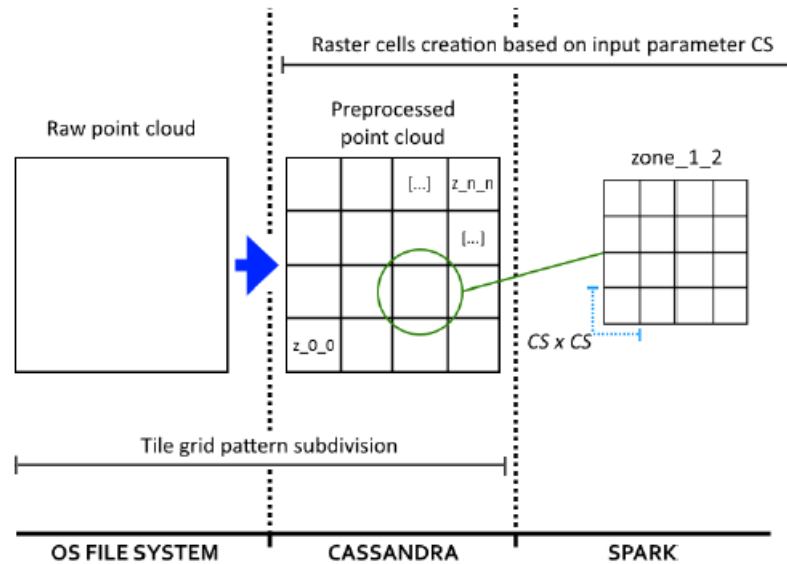


FIGURE 1. Overview of data processing in the proposed approach.

places that may be positioned at lowest altitudes that are far more appropriate for land areas. The cellular area's closest stages can be inspired to be greater than a single problem with potentially the lowest volume. The law is enforced to ensure that accurate executions continue to select a similar point. For example, selecting only the initial point packed at possibly the lowest amount could deliver a particular output in an assortment of executions, as information could be packed from one delivery to another, so this is different during our blunder correction method (see section 3). Choosing the nearest point from the center might improve the efficiency. When selecting the points, it should be a place in the close proximity of the cellular boundaries, as the stage will not reflect other cell points in those places [40].

The SC-091-12 algorithm takes advantage of multi-core CPUs to minimize the processing time [37]. However, the use of the multi-core CPU, which involves multiple threads during the air filtering process is not well investigated in the literature. Some works use single clouds in parallel for the large scale model of the filtering algorithm.

In the air filtering algorithm, the raw point cloud (unprocessed and original one) is classified by a double-dimensional (2D) grid architecture to achieve the degree of parallelism. Because of the 2.5-dimensional presence of aerial LiDAR stage clouds, much more complicated data partitioning is necessary [43]. The zones are combined into one file and sent individually to Cassandra. This section is performed during offline preprocessing until the entirety of the locations is estimated, as illustrated in Figure 1. These choices could directly impact the output of the system and the method of obtaining a DTM, as will be discussed in Section 4. The files introduced are called impartial and are then processed in parallel by Spark.

B. DISTRIBUTED STORAGE: CASSANDRA

Cassandra is the most recent version where storage devices are analyzed for the best method to manage large LiDAR

datasets and to encourage a web-based visualization process [24]. This empirical research assessed a selection of different types of repository designs: distributed file processing, wide-column storage, record storage, and key storage. With this assessment, the Hadoop Distributed File System (Redis [40], Cassandra [30], MongoDB [39], and HDFS [38]) were analyzed. Among them, Cassandra is selected as the storage technology because of its capability to query, similarity to many other classical SQL databases and its capability to integrate into large scale data processing solutions like Spark.

C. DISTRIBUTED COMPUTING: SPARK

The technique described below is usually viewed as a batch processing framework, as geospace processes run on huge static LiDAR datasets without supervision. Spark is the most appropriate option for the task because of its full integration with Cassandra [42]. Spark supports Java programming languages (same as the filtering algorithm used), batch processing, and flexibility. Because of these features, it is more suited for batch processing than stream processing. Hadoop [46] was not selected due to the limitation of programming availability besides map reduction. Many recent computing approaches, like Storm [44] and Flink [45], are being discarded due to their architecture based more on streaming.

Moving or executing algorithms in nodes is faster than transferring information, which is one of the core tenets of big data [59]. This theory assures the best possible parallelism through the assembly of a Spark on each Cassandra cluster node during the delivery of geospatial tasks. A message of the geospatial procedure code is sent out to a Spark agent by the Spark master to offload its computational tasks to subsets of LiDAR zones placed at Cassandra [50], [51]. This makes sure that every Spark Cassandra worker functions are kept in its very own node, thus preventing information transfer between nodes. Based on the parallel zones outlined in subsection 2.A, the time-consuming task is processed in

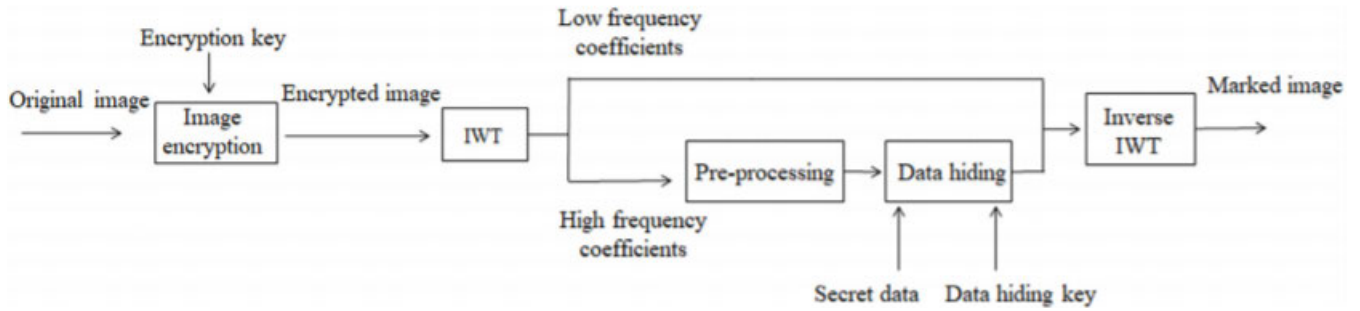


FIGURE 2. Data hiding and image encryption in the proposed approach.

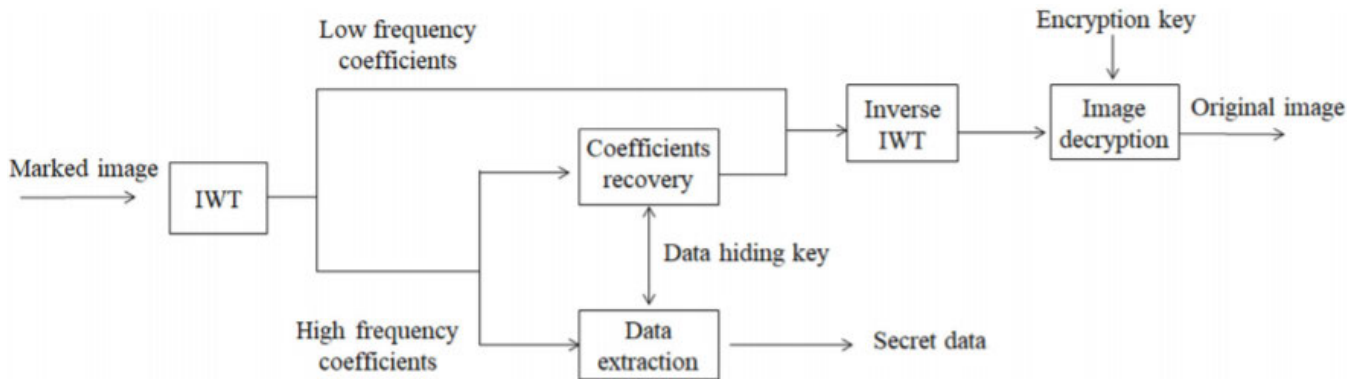


FIGURE 3. Data extraction and image recovery in the proposed approach.

parallel by dispersing all available physical cores on every processor of the individual Spark.

To spread information equally between nodes, Cassandra makes use of table keys. When the information delivery style suits the working distribution model, the maximum output is achieved. Growing Spark worker is assigned the same workload if each node in the group holds a similar portion of the total data, together with geospatial treatment, using the perfect information as input. Section 4 provides an analysis of such a situation. Optimizing the work distribution is out of the scope of this work, but can be referenced from [47].

III. PROPOSED APPROACH

In the proposed approach, the LiDAR images are ciphered by encryption. The encoded image is first decomposed with integer wavelet transformation (IWT). The frequency sub-bands (i.e., LL, HH, HL, and LH) are preprocessed to hide secret information. The changes are made in such a way that it provides effective data hiding of the proposed approach. Data hiding and image encryption in the proposed approach are shown in Figure 2. Conversely, the data extraction and image recovery in the proposed approach are shown in Figure 3. The receivers can extract the secret data using the data hiding key and recover the original image using the decryption key. The approaches are discussed in detail in the following subsections.

A. LiDAR ENCRYPTION METHOD

The LiDAR image is encrypted using a modified cipher. The modified cipher processes the pixels to a secure element [27].

Consequently, the encrypted image called a marked image, is subject to random noise. The static features of the original image remain unchanged. In theory, the modification cipher can be done by modifying a matrix Q . We employ Arnold's cat map [4] to code a modified cipher using various coefficients repeatedly. Let (M, N) represent the coordinates of the original pixel. Then, normal Arnold's cat map is defined as:

$$\begin{pmatrix} M' \\ N' \end{pmatrix} = \begin{pmatrix} 2 & 2 \\ 2 & 5 \end{pmatrix} \begin{pmatrix} M \\ N \end{pmatrix} \pmod{2}. \quad (1)$$

We utilized a discretized variant of this particular transform [5] created for $Z \times Z$ images as follows:

$$\begin{pmatrix} M' \\ N' \end{pmatrix} = \begin{pmatrix} 2 & c \\ d & cd + 1 \end{pmatrix} \begin{pmatrix} M \\ N \end{pmatrix} \pmod{Z}. \quad (2)$$

Subsequently, the simplified encryption key has three parameters: i , d , and c , where i is the number of iterations, d and c are positive integers. All the rectangular images can be split into several square sub-images for encryption.

B. DATA HIDING

In the information hiding stage [57], these encrypted images are first subject to the IWT to lift the parameters of the 4 sub-bands (i.e., HH, LL, LH, and HL). The IWT is given by:

$$e(i) = f(2i + 1) - \left[\frac{f(2i) + f(2i + 2)}{2} + \frac{1}{2} \right], \quad (3)$$

$$d(i) = f(2i) - \left[\frac{e(i - 1) + e(i)}{4} + \frac{1}{2} \right], \quad (4)$$

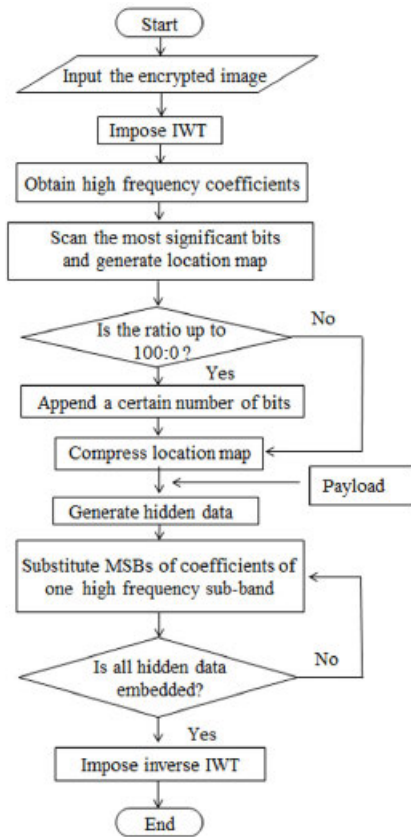


FIGURE 4. Flowchart of data embedding and preprocessing.

where $f(i)$ is the first signal, i.e., initial picture pixels, $e(i)$ is the higher frequency portion, and $d(i)$ the low frequency component. The inverse IWT is given by:

$$f(2i) = d(i) - \left[\frac{e(i-1) + e(i)}{4} + \frac{1}{2} \right], \quad (5)$$

$$f(2i+1) = d(i) + \left[\frac{f(2i) + f(2i+2)}{2} + \frac{1}{2} \right]. \quad (6)$$

C. DATA EXTRACTION AND IMAGE RECOVERY

At the receiver’s, there can be three different cases depending on the availability of keys.

Case 1: The receiver has the data hiding key: The receiver is capable of performing decryption on the secreta data. The receiver can extract MSB (Most Significant Bits), the high-frequency coefficients, to recover secreta data using a data hiding key.

Case 2: The receiver has the decryption key: The receiver can decrypt the marked image to get a picture that is similar to the original image.

Case 3: The receiver has both the data hiding key and the decryption key: The receiver is not only able to extract the secreta data but also completely recover the original image. The receiver first extracts the secreta data using the data hiding key. The LiDAR original image can be recovered by extracting the condensed insert maps in the MSB using the data hiding key and use arithmetic decoding to reconstruct the initial location map.

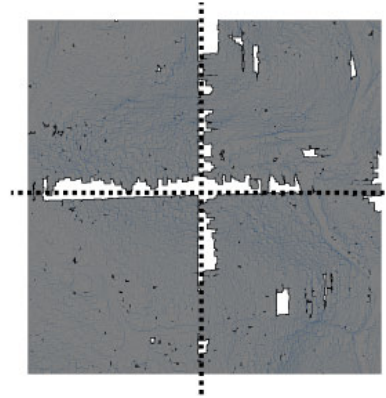


FIGURE 5. Raster showing the ground areas along the boundaries.

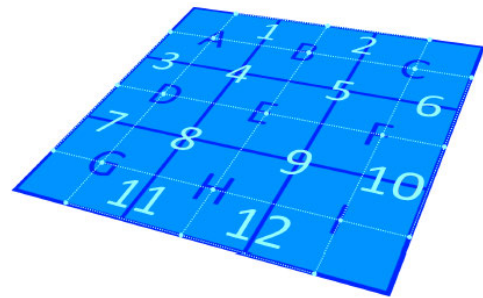


FIGURE 6. Schematic illustration of the instantaneous boundary error correction method. Each zone is labeled with a letter.

D. AUTOMATED BOUNDARY ERROR CORRECTION

The categorization or classification of a place into the ground or nonground depends mainly on the surrounding zones. Classification mistakes are common at such websites, as locations situated on the boundaries of neighboring filters individually miss useful details about the neighbors. Figure 5 shows a raster of terrain points obtained when the image was split into four preprocessing zones. Every zone was individually processed and then later combined. The boundaries show empty areas that do not exhibit natural geographic patterns. Such areas are misclassified as nonground and are not shown on the output image. Typically, semi-automatic or manual (scripted) processes can be used to avoid these errors. However, this involves human intervention [58], where a person first needs to distinguish the overlapping zones between neighboring zones.

Therefore, we propose an automatic error correction strategy that creates newer overlapping rasters between the zones. The areas from these overlapping rasters are marked, called correction patches, to avoid a potential error. For example, the square grids in Figure 6 show LiDAR zones and overlapping rasters between the zones.

E. CREATION OF CORRECTION PATCHES

Distribute processing by the Spark-Cassandra approach often creates overlapping rasters. The overlapping zones are determined using the pattern observed in Figure 6, considering all LiDAR zones being filtered. Bordering Spark agents share the overlapping raster. For example, the agent will acquire

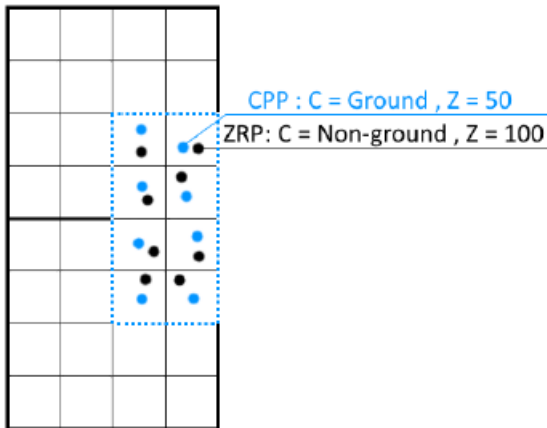


FIGURE 7. Illustration of a classification error modification. The zone raster point (ZRP) is misclassified as nonground and is replaced by correction patch point (CPP) that has a more accurate Z value.

zones B, E, C, and F for the overlapping raster 5. The CS input parameter decides the sizes on the overlapping rasters like the rasters on the places (as illustrated in Figure 1).

Lastly, correction patches are produced after the zones in every neighboring raster are determined by classifying all such rasters through the classification method. Correction patches are temporarily kept in Cassandra until the whole process is complete.

F. FILTERING AND ERROR CORRECTION OF THE LiDAR ZONES

LiDAR zones are transferred to the filtering algorithm as input. Any Spark operator runs the algorithm over the places within the same image, to ensure the information position. For each zone, a raster is produced with CS (as observed in Figure 1), which in turn classifies the points as ground or nonground.

Using Spark, the agent retrieves the respective modification rasters, which overlaps from Cassandra defined LiDAR zone

is filtered. As shown in Figure 6, in order to correct the raster errors obtained from zone H, the worker in charge need to request correction of patches 8, 9, 11 and 12. To correct this error, raster from the zone raster point (ZRP) and others from the correction patch point (CPP) is used to repair the error as follows:

- 1) When both components are marked with the same rating, the ZRP is considered correct.
- 2) The ZRP is deemed right if a ZRP is classified as a ground area and the CPP is classified as a nonground area. If an error occurred, the results will show gaps which correspond to the correction boundaries. The rasters overlap because of their very own restricting errors.
- 3) The ZRP is deemed misclassified when a ZRP is identified or categorized as nonground when the CPP is categorized as ground, and the ZRP should be called as ground.

IV. RESULTS AND DISCUSSION

A variety of experiments are carried out to show the effectiveness and feasibility of the proposed method, including various encryption techniques and correction strategies. We present an evaluation of execution time in subsection 4.A, a quantitative and visual evaluation of the boundary error quality in subsection 4.B, and computational and functional considerations for preprocessing in subsection 4.C.

To begin with, a regular LiDAR test image is used to demonstrate the effectiveness of the suggested algorithm, as shown in Figure 9.

Note that other well-known encryption techniques, such as modular encryption (ME) and stream Encryption (SE), are also suitable for the suggested system. Stream encryption moves the XOR process bit by bit. Modular encryption changes every pixel with $C(j, k) = (c(j, k) + q(j, k)) \bmod 256$, where $q(j, k)$ are the unique pixel coordinates at (j, k) , and $c(j, k)$ is a random number between 0 and 255.

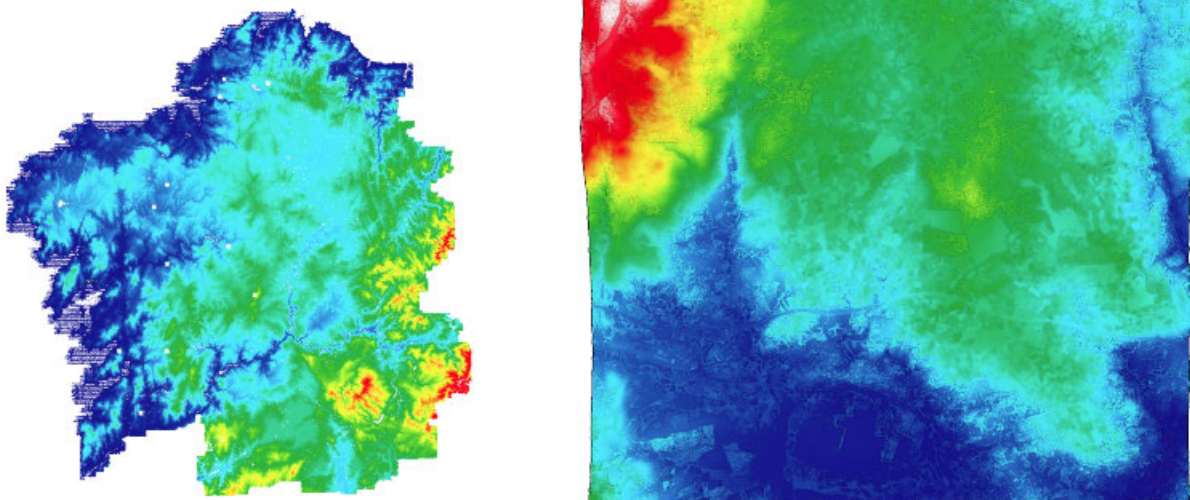


FIGURE 8. Analyze Point clouds: (a) PNOA issue cloud, (b) Guitiriz issue cloud.

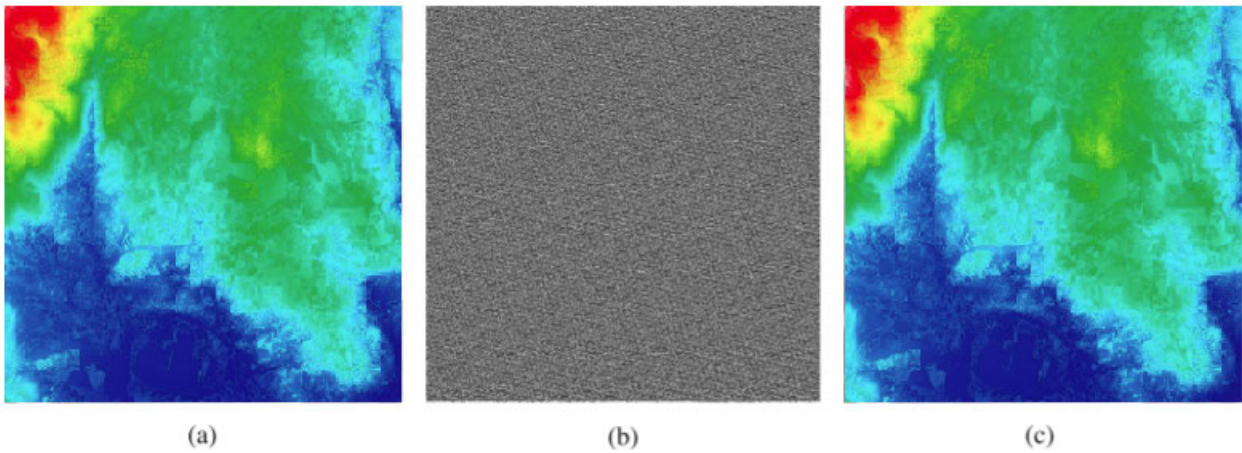


FIGURE 9. (a) LiDAR Image, (b) marked picture with secret data, and (c) reconstructed original image.

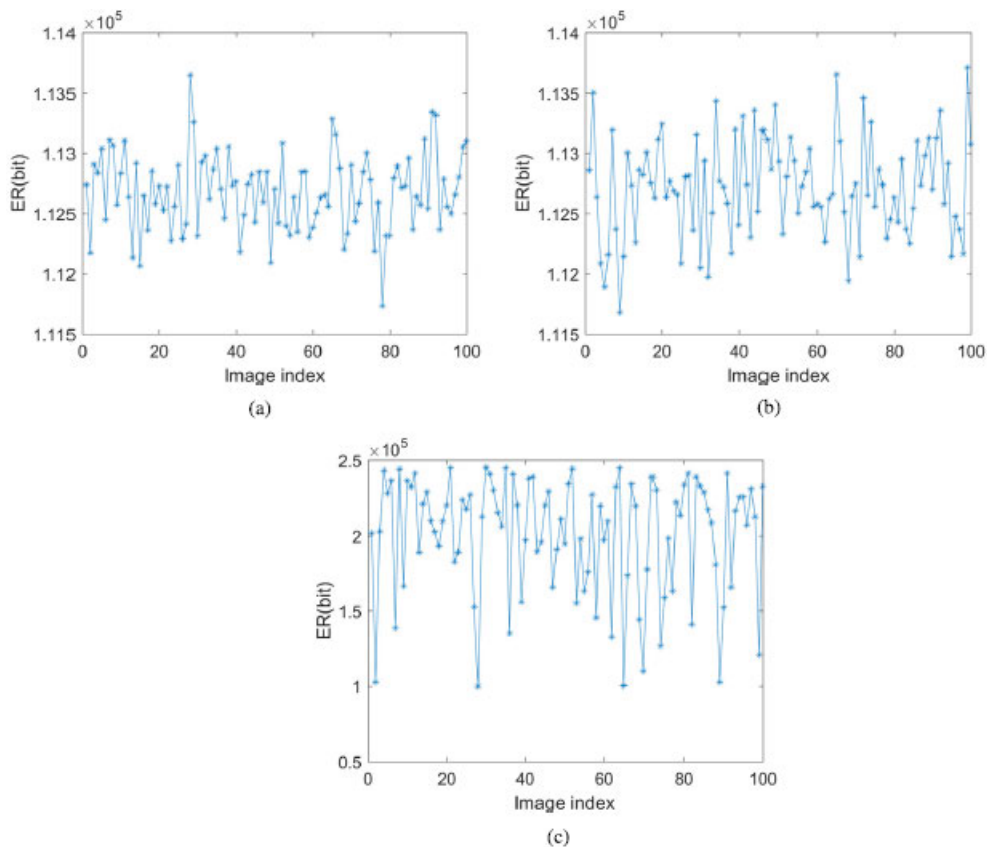


FIGURE 10. Comparison of different strategies used for encryption: (a) stream encryption, (b) modular encryption, and (c) permutation encryption.

Comparing these two techniques with permutation encryption (PE) in terms of security strength and embedding cost [52], [56], we conclude that PE is more appropriate. SE and ME are popular encryption techniques that produce pseudo arbitrary cipher streams according to the encryption key ranging from 8 to 4096 bits. The standard size is 128 or 256 bits. As for PE, the key value is represented using 3 decimal values using the discretized version of Arnold’s cat map.

To compare the performance of the three different encryption techniques, 1000 LiDAR images were randomly

selected from the ISPRS repository [49]. Here, the adjustable correction tactic is used. Figure 10 shows the different capabilities of the three encryption approaches. We can observe that the PE encryption substantially exceeds other encryption techniques. This is primarily because the PE keeps global information of the original images completely free of distortion.

Figure 11 shows the pixel location of the encrypted and original image. We can see that the pixel is smoothly distributed for all encryption techniques, providing

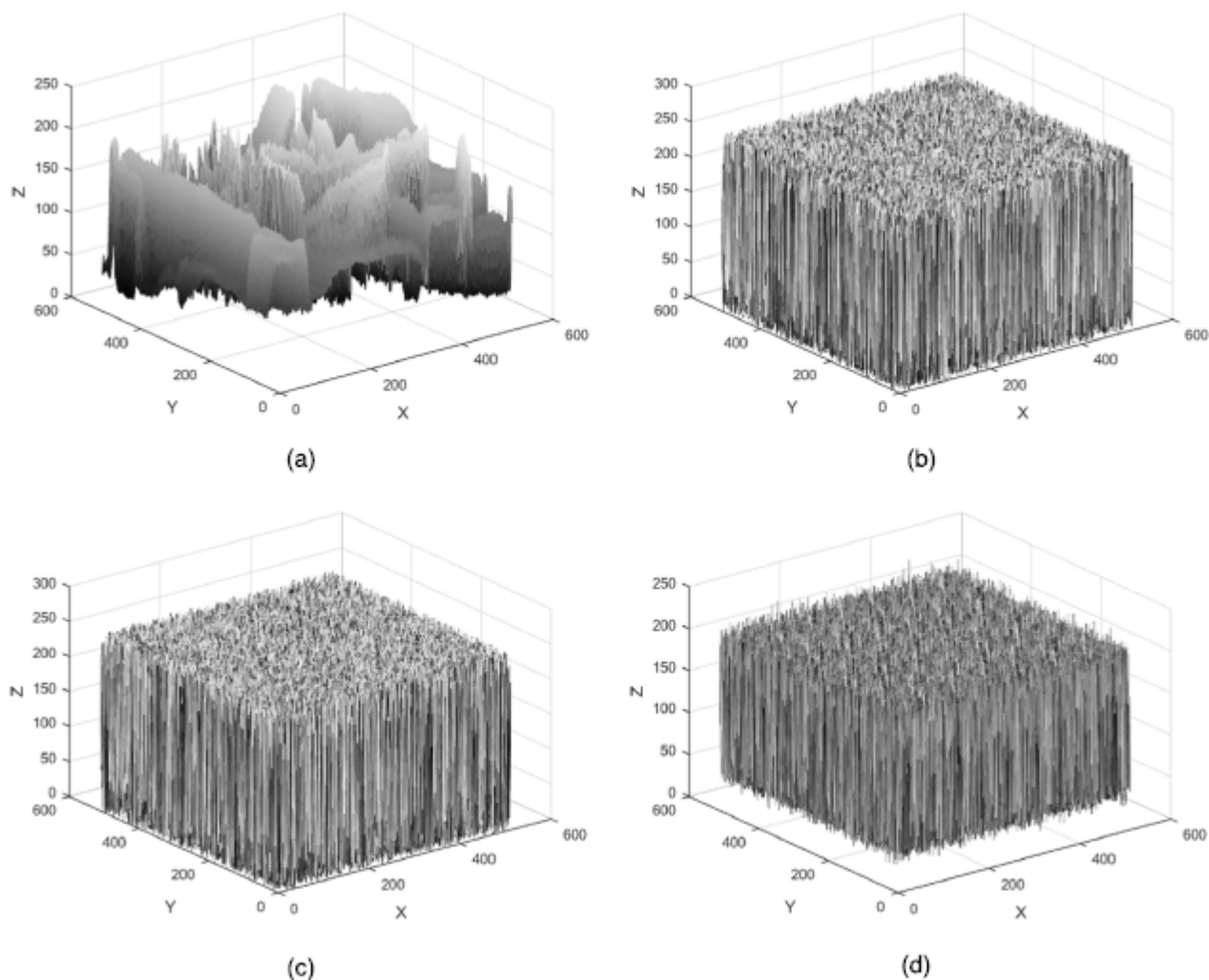


FIGURE 11. Pixel location (a) LiDAR image (b) stream encryption (c) modular encryption and (d) permutation encryption.

needed protection. PE is more susceptible to known/chosen plain-text attacks among these three encryption methods. However, this could be easily mitigated through using different key values in each iteration and image-varying keys [49].

The features of datasets applied for the analysis are outlined in Table 1. Four individual datasets have been obtained from the image repository, as detailed in Table 2. Datasets A0, A1, and A2 were constructed using three distinct zone sizes from the same cloud (PNOA issues) as shown in see Figure 8a. The study of unit output, in certain cases, calls for the utilization of A3 (created from Guitiriz stage cloud, see Figure 8b), which has massive sizes (26,472 KB per zone on average). The subdivision of the point clouds was not only performed during the preprocessing stage, but also a compression of the resultant files. To be able to reduce information processing, a compression strategy described in [10] is used.

The computing machine specification is as follows: Intel Core i9-9960 3.1 GHz (16 Cores), 64 GB RAM, SATA3 HDD, and CentOS (6.10 InfiniBand). Even with

TABLE 1. Details of LiDAR datasets selected for pre-processing: number of points (NP) in billions, number of files (NF), file extent (FE) in m², file size (FS) in KB per file, total issue cloud sizing (TPCS) in GB.

Cloud Point	NP	NF	FE	FS	TPCS
PNOA (Region)	23	5415	1800 × 1800	85,297	335
Guitiriz (Village)	1	115	500 × 500	115,128	23

TABLE 2. Details of cloud datasets: number of point cloud zones (NPZ), extent of point cloud zone (NZE) in m², size of point cloud zone (NZS) in KB per zone, total dataset sizing (TDS) in GB.

Images	Source	NPZ	NZE	NZS	TDS
A0	PNOA	10,551	1500 × 1500	7814	88
A1	PNOA	170,712	300 × 300	452	88
A2	PNOA	1,76,5086	100 × 100	28	102
A3	Guitiriz	285	400 × 400	28585	7

the low computing computer, the Spark manager successfully works with the Spark agents and the Cassandra server. Several configurations are made in Spark and Cassandra. These options have to be optimized, considering the topology,

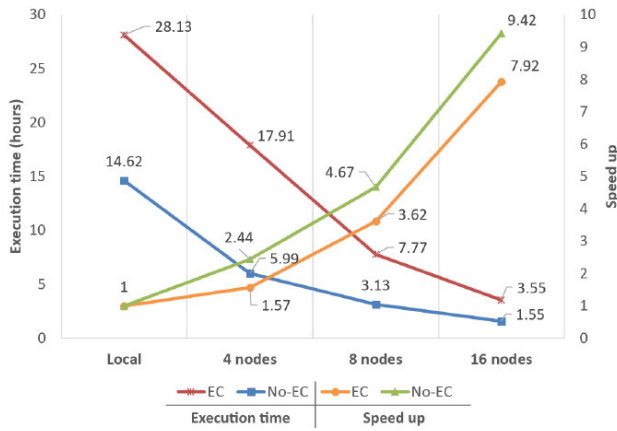


FIGURE 12. Scalability and performance comparison using A0 dataset(1500 × 1500).

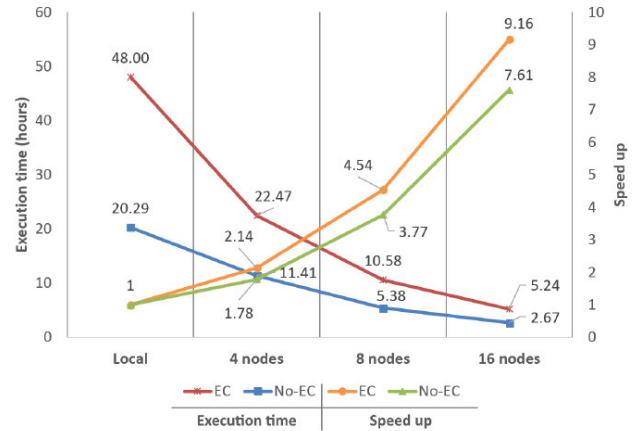


FIGURE 14. Scalability and performance comparison using A2 dataset (100 × 100).

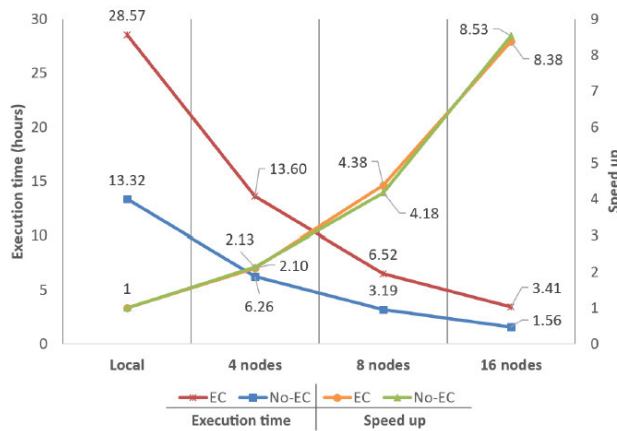


FIGURE 13. Scalability and performance comparison using A1 dataset (300 × 300).

intricacy of the method conducted, and amount of data, and type of data, to produce the best results. The filtering algorithm has been configured to set CS to 1.5 for input parameters.

A. EVALUATION OF EXECUTION TIMES

The execution times for the four data sets, presented in Table 2, are calculated to evaluate the processing efficiency and assess the scalability of the system. Note that the entire cycle of filtering involves generating rasters, classifying raster zones, and correcting errors. Two different cases were considered: EC and NO-EC. The system running in a single node called local is compared with the distributed approach running in 4, 8, and 16 nodes. The local node does not use Spark or Cassandra. The results for datasets A0, A1, and A3 for different numbers of nodes are shown in Figures 12, 13, and 14, respectively.

As shown in Figure 12, the speed up time for 4, 8, and 16 nodes using the A0 dataset compared with the local are, respectively, 1.57×, 3.62×, and 7.92× for EC 2.44×, 4.67×, and 9.42× for NO-EC. As shown in Figure 13, the result for the speed up time using the A1 dataset shows a similar trend to that of the A0 dataset, i.e., 2.10×, 4.38×, and 8.38× for EC. Finally, as shown in Figure 14, the speed up using the A2 data

set is 2.14×, 4.54×, and 9.16 for EC, and 1.78×, 3.77×, and 7.61× for NO-EC. From the results, we can expect about 2× speed up time for four-node cases. Moreover, the speed up time increases linearly as the number of nodes increases.

The fastest arrangement with the highest efficiency was the one with sixteen node configurations of zones and EC of 400 × 400 (A1) at 3.41 hours regarding the delivery time. Output times observed for A2 would possibly be always better compared to A1. Per zone, a reduction in KB increases latency and Cassandra efficiency. The variance in the volume of expertise between nodes in most separate operating scenarios and time penalty for the initialization of the air filter algorithm to illustrate these variants. Whenever a particular processing product (zone) is filtered, a selection of information structures is started, and several first estimates demanding some startup/initialization time penalties need to be made. Consequently, additional places are filtered, and time penalties would intensify for startup/initialization.

Information flow between nodes is virtually impossible in the first scenario (NO-EC) because the Spark-Cassandra interconnection promises information locality. Spark agent members can work with the air filtering algorithm on aspects found within their very own nodes, so the KB in each zone would achieve little to no decrease in delivery time. Nevertheless, concerning the number of aspects being sorted, the novice penalty on the air filtering algorithm decreases. Under this situation, a reduction in the zone’s size almost certainly leads to an enhancement in run-time.

There are considerably more movements between nodes in the next situation (EC), e.g., for modification of errors in adjacent zones, kept in numerous nodes must one maintenance spot be transferred. When adequate amounts of facts motions are found, the potential output advantage regarding the decrease of KB by zone begins to show up. This decrease in time connected with the transfer of information between nodes results in a total decrease in deployment times by compensating for a time boost related to startup sanctions.

Figure 15 demonstrates how execution occasions differ (using a logarithmic scale) because the places with EC and

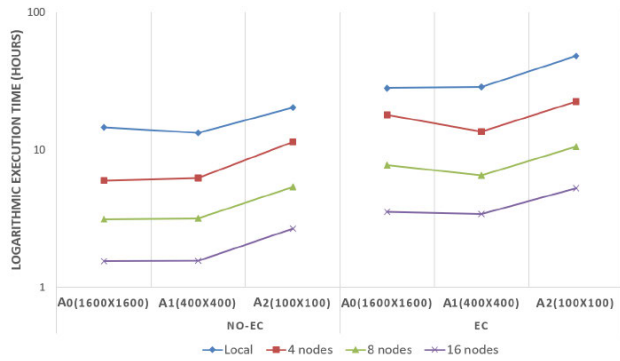


FIGURE 15. Execution times in different zones for NO-EC and EC.

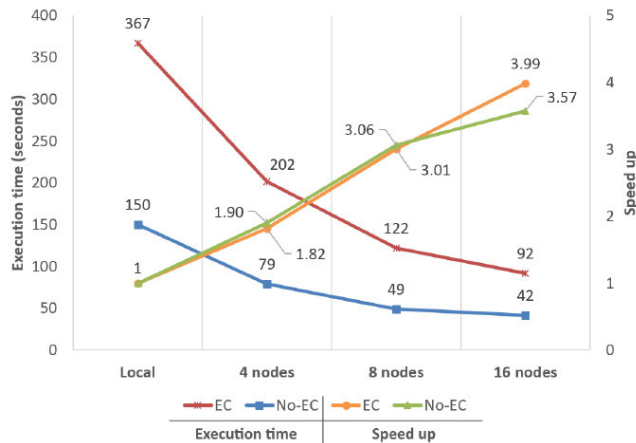


FIGURE 16. Performance comparison using A3 dataset using 4, 8 and 16 nodes.

NO-EC shift their sizes. The delivery time is reduced when the size of the region is reduced. The reduction in the number of aspects being filtered plays a role in improving the processing times, enabling one to account for the potential reduction in processing resulting from the increased bytes to pass across the system when boundary errors are fixed successfully. Nevertheless, the delivery times nearly end as they go from 400 to 1600 to 1600. There is no substantial change with NO-EC; thus, the volume of information is considerably large as the optimum errors are repaired, resulting in a fall of product output, causing negative times with EC.

The speedups acquired by the A3 (Figure 16) were 1.9x, 3.06x, 3.57x, 1.82x, 3.01x, and 3.99x. Though the overall performance gain of the big data technique was considerable, the results distant relative to the others were not as significant as the ones from the other information sets. Such results are clarified not just from the specific quantity of aspects being managed, unlike the functional cores, but by the wide scope of the zones (approximately 30 MB).

B. EVALUATION OF BOUNDARY ERROR CORRECTION

Three different methods are applied to evaluate the performance of the boundary error correction of the proposed method: an A3 considerable data obvious analysis and

two quantitative analyzes using several of the ISPRS [39] databases.

Figures 17 and 18 are essential for visible contrasts between EC and NO-EC. Figure 17 gives a restricted perspective of only ground points, including two filtered A3 rasters. Figure 17a reveals significant defects in cream zones that are omitted in Figure 17b. The vacant spots in this zone are vast areas of woodland or even small residential units. In Figure 18, two triangulated rasters from the prior raster series are shorter on the opposite side. The evaluation unit Global Mapper has triangulated areas from the rasters [49]. There is an error along the zone boundary in Figure 18a, but there is no error present in Figure 18b.

In Table 3, the standard Type I error (rejecting soil marks) is utilized to build a conceptual contrast between NO-EC. Note that raster points just influence type I errors; thus, each stage in a simple cloud point not incorporated into the raster gives rise to a raster error, resulting in rates observed in the graph. A comparison of these was made utilizing several datasets from ISPRS (first column) since the examination algorithms are viewed as normal. Because of the lower point intensity of the information sets, CS was placed into one for such measurements. The next column displays the errors of different types for every data set, using the whole cloud of areas as an entry. The next column is undivided. Just about all datasets had been split into four lesser, similarly broad regions (with the design of division in Fig. 2), accompanied by EC and NO-EC (fourth and third columns) processing throughout every area.

Boundary errors develop within the four zones when sorting out the datasets, which can be seen when contrasting the third and second columns of Table 3. The fourth and third columns indicate that the portion of errors is reduced over the pre-divided point because of the proposed technique. The error rate with EC is considerably below the indexed dataset amount clarified for the two main elements. The first is simply because the rasters of the separated datasets have a small number of add-ons. The resulting raster has 81 points spread in a power grid of nine to nine rows, such as a raster out of the zone nine m on the zone nine m using CS = 1 m. After being split into four small ranges of 4.5 meters, the resulting four rasters will have a hundred points spread out more than four grids of 5 individual cells, since another half meters would imply that every power grid has an extra cell. Such additional information is likely to decrease recognition consistency and, consequently, the errors of group I. Another reason is related to the dimensions of the correction level, which extends outside of the boundaries on the regions; therefore, enabling the modification of some misclassified points from the boundaries.

A novice driver measurement is proposed to determine the misclassification of areas resulting from the sections improvement and the datasets in our automatic boundary error correction application.

$$\delta = \frac{(NGP - NMP)_{divided}}{(NGP - NMP)_{undivided}} \times 100 \tag{7}$$

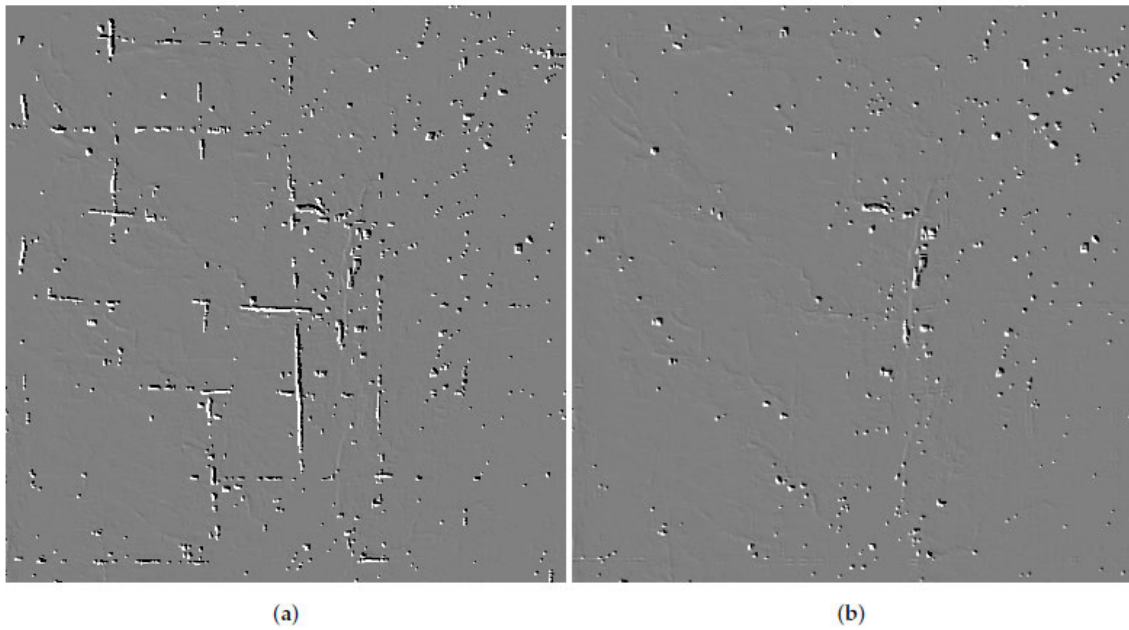


FIGURE 17. Filtered rasters: (a) NO-EC (b) EC.

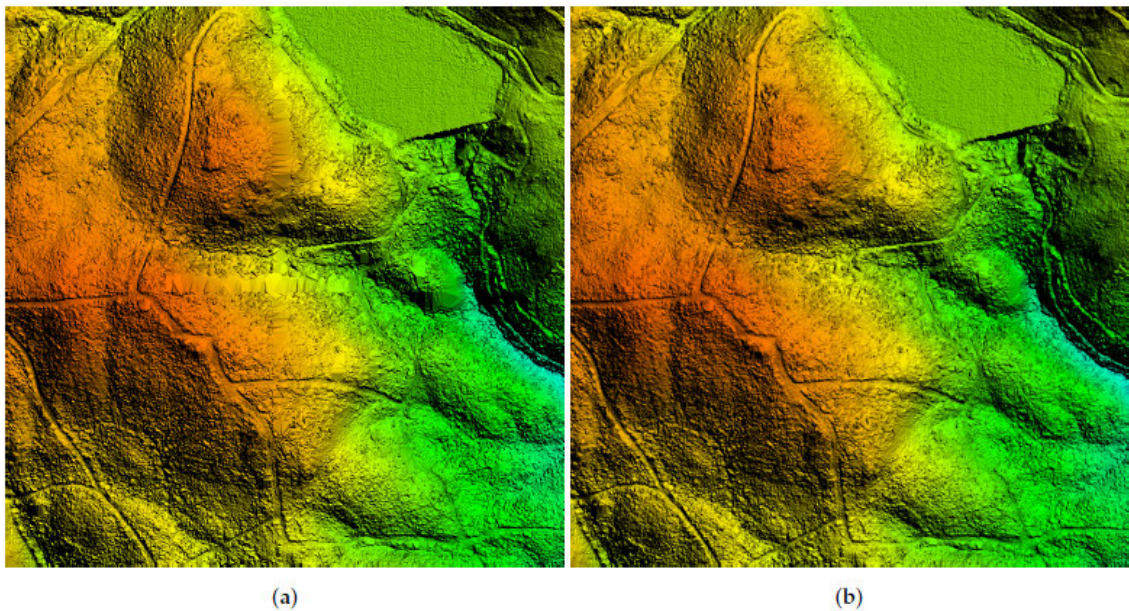


FIGURE 18. Fully triangulated rasters from the data set A3: (a) NO-EC (b) EC.

where NGP means the amount of terrain points and NMP is the amount of ground points that are mis-classified.

Table 4 shows the latest metric described above, using various ISPRS datasets. The first two columns give information on the first ISPRS datasets, while additional columns appear info regarding the obtain rasters together with the latest metric function.

It can be observed, later partition, the first sets of data $\delta < 100$ for most rasters with NO-EC because of so numerous ground areas accurately categorized in undivided rasters, no longer provided in the split ones. With EC, most of the correctly classified soil points were missed after the sections

are restored, acquiring $\delta \geq 100$. It must be mentioned here that, in many instances, $\delta > 100$, owing to a small number of additional points that are integrated on the raster with the proposed error correction technique.

C. SIGNIFICANCE OF PREPROCESSING OF THE CLOUD POINT

The conclusion provided in Sections 4.1 and 4.2 shows the significance of choices made for LiDAR preprocessing procedures, as the separation and storage of the clouds by Cassandra would substantially affect the effectiveness of the unit. Reducing byte dimensions in zones will improve the

TABLE 3. Type I error comparability between NO-EC and EC utilizing a number of the datasets provided by the ISPRS [49].

Image from datasets	undivided	(NO-EC) divided-in-4	(EC) divided-in-4
Image 19	46.22	54.29	48.23
Image 29	49.82	50.61	49.90
Image 59	21.80	25.52	23.58
Image 65	25.98	28.31	27.58
Image 81	11.28	13.91	11.38
Image 95	7.02	8.14	7.53
Image 102	5.81	7.54	6.07
Image 118	12.21	13.94	12.54
Image 128	15.20	17.27	15.83
Image 135	21.50	22.71	21.71

TABLE 4. Misclassification regions as an outcome of the areas on the datasets maintained by the ISPRS [49] and the improvements brought about by the proposed method.

Image from Datasets	Un-divided			(NO-EC) Divided-into-4			(EC) Divided-into-4		
	NGP	NGP	NMP	NGP	NMP	δ	NGP	NMP	δ
Image 19	20,142	11,126	342	10,152	387	94.1	11,249	487	98.82
Image 29	21,318	12,746	267	12,465	379	97.75	13,517	429	98.61
Image 59	19,249	17,417	153	16,218	163	95.23	17,521	172	98.12
Image 65	30,513	27,291	113	26,479	117	97.73	27,194	112	98.34
Image 81	3,249	3,491	127	3,427	139	98.13	3,418	143	99.29
Image 95	31,917	28,497	48	30,917	42	98.16	30,197	49	99.29
Image 102	18,567	17,693	279	17,683	300	98.27	17,734	283	99.59
Image 118	13,453	13,982	216	13,951	241	98.19	13,791	219	98.19
Image 128	2,786	2,813	105	2,791	109	97.16	2,913	114	99.17
Image 132	1,646	1,613	179	1,943	167	98.46	1,533	187	99.29

effectiveness of the system; however, after a specific amount is achieved, the output switch will stagnate or get more intense (as previously mentioned in Figure 15).

Cassandra performs best with a large amount of information transfer between nodes, reducing the zones. Nonetheless, the quantity of information is mainly determined by the type and timing of geospatial operations within the system. Therefore, though it may at first seem apparent that, in this particular analysis, for instance, in the situation of use, we could generate Spark's work by being forced to resolve all limit mistakes made by the divisions of time clouds. In the usage of Spark, for instance, it must be correctly determined in this analysis.

This study shows that the number of information exchanges between nodes should ideally be considered when deciding how point clouds are split and refined, and not merely the way the generation effectiveness of Spark's algorithms pertains to the byte size or maybe zone levels and just how these qualities will affect the work and functionality of Spark. Additionally, the vital importance of achieving the appropriate equilibrium between all the aspects, zone size consistency of the geospatial approach must be emphasized. It goes beyond the current research scope to produce an instant system to choose an optimum degree or a number of places, though it is a crucial topic that needs to be worked on.

Classification of full points to add full point classification, the kind of overall performance of the present air filtering algorithm will be lengthy. To ascertain the worth of this much more purpose, we have determined a naive approach. The particular filter efficiency is utilized for a brand-new stage to reference the category of all areas within the raw point cloud to obtain this function. Each issue from LiDAR zones has established a cell within the current dribble rasters based on its X and Y coordinates. The current unclassified factor

TABLE 5. Observation between entire issue cloud arrangement errors acquired through the proposed method, plus LAsTools.

Image from Dataset	Error %		Error I %		Error II %	
	Proposed	LAsTools	Proposed	LAsTools	Proposed	LAsTools
Image 19	16.05	16.47	26.71	25.48	3.12	4.81
Image 29	5.03	6.18	7.18	11.81	2.02	1.87
Image 59	1.59	5.54	1.12	7.48	2.84	1.48
Image 65	13.89	13.94	13.99	14.18	8.14	3.13
Image 81	7.17	14.45	7.10	15.78	4.43	0.34
Image 95	9.11	12.13	10.42	14.48	4.15	2.17
Image 102	12.46	12.84	12.93	12.47	8.15	8.47
Image 118	1.50	6.04	1.30	4.81	4.14	2.15
Image 128	8.15	8.42	8.13	6.18	4.46	3.45
Image 132	5.48	5.78	6.07	6.46	2.46	2.02

is labeled with the same method as a raster thing if the gap between heights between the stage in the raster cell and the purpose of the area is much less than a parameter, defined as a height threshold (HT). Subsequently, the level of inconsistency with the other category is determined.

The ISPRS Filter Check datasets would be a contrast between this simple procedure and LAsTools [48]. The results of LAsTools are obtained from [49] as it is one of the most recent papers on the field of LiDAR stage clouds. Full errors (percentage of error areas), type I errors (refuse of terrain points), and type II errors (acceptance of non-land factors as land points) have been linked. Due to the lower density of the datasets, the CS and HT parameters were adjusted to one and 30 cm, respectively.

The results indicate that our strategy, on average, is much better compared to LAsTools, but marginally worse when considering type II errors. They are an anticipated consequence, as the primary purpose of the SC-091-12 algorithm was centered on obtaining ground dotted grids rather than determining points outside the ground. Of the ISPRS tests, Sample 11 was among the most difficult to control in different grading algorithms, mainly due to the steep incline of the surface and a great amount of format covering the whole exterior area of the landscape. As mentioned above, a simple approach was used to classify clouds in this 1st strategy, which obtains a higher amount of type I errors in the specific situation of Sample 11 than for others; nonetheless, the proportion of an error which is virtually the same as the one acquired with LAsTools is supplied. The utilization of the entire ingredient description as a possible investigation has been determined, and also the threat of errors is reduced.

V. CONCLUSION AND FUTURE WORK

In this article, large scale parallelization of geospatial tasks by big data systems was studied; in cases like this, Cassandra and Sparks are proven. With all the distributed computation methods created in this research, the effectiveness of the SC-091-12 algorithm was drastically enhanced. Aside from permutation, encryption is applied to the cover image. The transformation domain is used for data hiding. This key information is embedded in the higher frequency coefficients of the integer wavelet transform, through preprocessing and bit exchange. The excellent combined unit energy, the extremely programmable architecture of these two solutions, also supplies the capability, by basic incorporation of brand-new

computing phases, just like the error correction amount, to further create and increase the freedom of these geospatial tasks. This method was used to fix the classification errors identified independently by the air filtering algorithm, along the boundaries of adjacent zones.

The results of this study prove that the proposed method is powerful, with no error modification from 13.32 to 1.56 h, lowering the processing time of 21 billion points in one system from 28.57 to 3.41 h, and also of attaining a speed up from 8.4 a) in 16 node implementations (see Figure 13). The use associated with a big data strategy does not only consist of centralized processing for LiDAR information. Additionally, it carries with it the regular storage advantages related to this particular type of method, like scalability and good quality of data. The ideas suggested here would considerably benefit from GIS centers, federal agencies, or maybe other types of entities with vast data volumes; the entire approach to obtain total triangulation DTMs may be used for real-time processing with the necessary quantities of computing resources.

To finish the category clouds point in the far extra advanced and complicated algorithm outlined in Section 4.4, we shall personalize the process so that sometimes a filtered raster is from a certain problem cloud or maybe from a complete filtered point cloud. To be able to eliminate human intervention and to enhance the effectiveness of the distributed computing system, the setup of an automated tool to choose the perfect size and number of processing products can also be envisaged.

In tandem with the lower cycle times obtained, the independent nature of most processing phases gives rise to the possibility of monitoring the unit as a services-oriented strategy to the on-demand version of DTM/DSM, which will be exclusive and extremely valuable to various other uses of LiDAR. It can also be managed with ample real-time machine resources. A better method accounts for many geospatial tasks, such as the distributed processing framework mentioned in this study, to put on such processes over entire point clouds and just areas of concern displaying the people. The geospatial operations are included in this file.

REFERENCES

- [1] Y.-Q. Shi, X. Li, X. Zhang, H.-T. Wu, and B. Ma, "Reversible data hiding: Advances in the past two decades," *IEEE Access*, vol. 4, pp. 3210–3237, 2016.
- [2] J. Fridrich, M. Goljan, and R. Du, "Invertible authentication, security, and watermarking of multimedia contents III," *Proc. SPIE*, vol. 4314, pp. 197–208, Aug. 2001.
- [3] M. U. Celik, G. Sharma, and A. M. Tekalp, "Lossless watermarking for image authentication: A new framework and an implementation," *IEEE Trans. Image Process.*, vol. 15, no. 3, p. 354–362, Oct. 2006.
- [4] J. Tian, "Wavelet-based reversible watermarking for authentication," *Proc. SPIE*, vol. 4675, pp. 679–690, Apr. 2002.
- [5] J. Tian, "Reversible data embedding using a difference expansion," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 8, pp. 890–896, Aug. 2003.
- [6] Z. Ni, Y.-Q. Shi, N. Ansari, and W. Su, "Reversible data hiding," in *Proc. IEEE Int. Symp. Circuits Syst.*, Dec. 2003, pp. 912–915.
- [7] R. Malik, R. Dhir, and S. K. Mittal, "Remote sensing and landsat image enhancement using multiobjective PSO based local detail enhancement," *J. Ambient Intell. Hum. Comput.*, vol. 10, no. 9, pp. 3563–3571, Sep. 2019.
- [8] Z. Ni, Y.-Q. Shi, N. Ansari, and W. Su, "Reversible data hiding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 3, pp. 354–362, Mar. 2006.
- [9] E. M. González-Ferreiro, D. Miranda, L. Barreiro-Fernandez, S. Bujan, J. Garcia-Gutierrez, and U. Dieguez-Aranda, "Modelling stand biomass fractions in galician eucalyptus globulus plantations by use of different LiDAR pulse densities," *Forest Syst.*, vol. 22, no. 3, p. 510, Nov. 2013.
- [10] A. Khamparia, A. Singh, D. Anand, D. Gupta, A. Khanna, N. Arun Kumar, and J. Tan, "A novel deep learning-based multi-model ensemble method for the prediction of neuromuscular disorders," *Neural Comput. Appl.*, vol. 32, no. 15, pp. 11083–11095, Aug. 2020.
- [11] G. Ventura, G. Vilardo, C. Terranova, and E. B. Sessa, "Tracking and evolution of complex active landslides by multi-temporal airborne LiDAR data: The montaguto landslide (Southern Italy)," *Remote Sens. Environ.*, vol. 115, no. 12, pp. 3237–3248, Dec. 2011.
- [12] M. S. Hossain, G. Muhammad, B. Song, M. M. Hassan, A. Alelaiwi, and A. Alamri, "Audio-Visual emotion-aware cloud gaming framework," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 12, pp. 2105–2118, Dec. 2015.
- [13] Q. Fang, J. Sang, C. Xu, and M. S. Hossain, "Relational user attribute inference in social media," *IEEE Trans. Multimedia*, vol. 17, no. 7, pp. 1031–1044, Jul. 2015.
- [14] M. S. Hossain, M. Moniruzzaman, G. Muhammad, A. Ghoneim, and A. Alamri, "Big data-driven service composition using parallel clustered particle swarm optimization in mobile environment," *IEEE Trans. Services Comput.*, vol. 9, no. 5, pp. 806–817, Sep. 2016.
- [15] C. Yang, M. Yu, F. Hu, Y. Jiang, and Y. Li, "Utilizing Cloud Computing to address big geospatial data challenges," *Comput. Environ. Urban Syst.*, vol. 61, no. 2017, p. 120–128.
- [16] S. Li, S. Dragicevic, F. A. Castro, M. Sester, S. Winter, A. Coltekin, C. Pettit, B. Jiang, J. Haworth, A. Stein, and T. Cheng, "Geospatial big data handling theory and methods: A review and research challenges," *ISPRS J. Photogramm. Remote Sens.*, vol. 115, pp. 119–133, May 2016.
- [17] M. S. Hossain and G. Muhammad, "Emotion-aware connected health-care big data towards 5G," *IEEE Internet Things J.*, vol. 5, no. 4, pp. 2399–2406, Aug. 2018.
- [18] R. Malik, R. Dhir, and S. K. Mittal, "Modified gain intervention refinement filter and multiobjective particle swarm optimization based local detail enhancement technique," *J. Eng. Technol.*, vol. 6, no. 2017, pp. 340–351, 2017.
- [19] A. Khamparia, B. Pandey, D. K. Pandey, D. Gupta, A. Khanna, V. H. C. de Albuquerque, "Comparison of RSM, ANN and fuzzy logic for extraction of oleonic acid from ocimum sanctum," *Comput. Ind.*, vol. 117, May 2020, Art. no. 103200, doi: 10.1016/j.compind.2020.103200.
- [20] L. Wang, Y. Ma, J. Yan, V. Chang, and A. Y. Zomaya, "PipsCloud: High performance cloud computing for remote sensing big data management and processing," *Future Gener. Comput. Syst.*, vol. 78, pp. 353–368, Jan. 2018.
- [21] Y. Ma, H. Wu, L. Wang, B. Huang, R. Ranjan, A. Zomaya, and W. Jie, "Remote sensing big data computing: Challenges and opportunities," *Future Gener. Comput. Syst.*, vol. 51, pp. 47–60, Oct. 2015.
- [22] J. Boehm, "File-centric Organization of large LiDAR Point Clouds in a Big Data context," in *Proc. 1st Workshop Process. Large Geospatial Data*, Cardiff, U.K., Jul. 2014, pp. 69–76.
- [23] M. Brédif, B. Vallet, and B. Ferrand, "Distributed dimensionality-based rendering of lidar point clouds," *Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. 3, pp. 559–564, Aug. 2015.
- [24] D. Deibe, M. Amor, and R. Doallo, "Big data storage technologies: A case study for Web-based LiDAR visualization," in *Proc. IEEE Int. Conf. Big Data*, Dec. 2018, p. 3831.
- [25] S. Garg, K. Kaur, N. Kumar, G. Kaddoum, A. Y. Zomaya, and R. Ranjan, "A hybrid deep learning-based model for anomaly detection in cloud datacenter networks," *IEEE Trans. Netw. Service Manage.*, vol. 16, no. 3, pp. 924–935, Sep. 2019.
- [26] H. Persson, J. Wallerman, H. Olsson, and J. E. S. Fransson, "Estimating forest biomass and height using optical stereo satellite data and a DTM from laser scanning data," *Can. J. Remote Sens.*, vol. 39, no. 3, pp. 251–262, Sep. 2013.
- [27] Y. Qian, M. Chen, J. Chen, M. S. Hossain, and A. Alamri, "Secure enforcement in cognitive Internet of vehicles," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 1242–1250, Apr. 2018.

- [28] D. Deibe, M. Amor, R. Doallo, D. Miranda, and M. Cordero, "GVLiDAR: An interactive Web-based visualization framework to support geospatial measures on lidar data," *Int. J. Remote Sens.*, vol. 38, no. 3, pp. 827–849, Feb. 2017.
- [29] A. A. A. El-Latif, B. Abd-El-Atty, M. S. Hossain, S. Elmougy, and A. Ghoneim, "Secure quantum steganography protocol for fog cloud Internet of Things," *IEEE Access*, vol. 6, pp. 10332–10340, 2018.
- [30] *The Apache Software Foundation. Apache Cassandra*. Accessed: Jan. 8, 2018. [Online]. Available: <http://Cassandra.apache.org/>
- [31] *The Apache Software Foundation*. Accessed: Jan. 22, 2019. [Online]. Available: <http://spark.apache.org/>
- [32] R. Buyya, A. Dastjerdi, and R. Calheiros, *Big Data Principles and Paradigms*. Amsterdam, The Netherlands: Elsevier, 2016, p. 39–58c.
- [33] M. S. Hossain and G. Muhammad "Cloud-based collaborative media service framework for healthCare," *Int. J. Distrib. Sensor Netw.*, vol. 2014, Oct. 2014, Art. no. 858712.
- [34] K. Zhang, S.-C. Chen, D. Whitman, M.-L. Shyu, J. Yan, and C. Zhang, "A progressive morphological filter for removing nonground measurements from airborne LIDAR data," *IEEE Trans. Geosci. Remote Sens.*, vol. 41, no. 4, pp. 872–882, Apr. 2003.
- [35] R. Crecente. *LIDAR 2MDT Plus Generaci3n de Modelos Digitales de Terreno de Pendiente Variable a Partir de Datos LIDAR Mediante Filtro Morfol3gico Adaptativo y Computaci3n*. Accessed: Mar. 28, 2012. [Online]. Available: <http://www.ibader.gal/seccion/378/Outras-actividades.html>
- [36] M. S. Hossain, S. U. Amin, G. Muhammad, and M. Al Sulaiman, "Applying deep learning for epilepsy seizure detection and brain mapping visualization," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 15, no. 1, Feb. 2019, Art. no. 10.
- [37] V. C. Recarey, D. M. Barr3s, and D. A. A. Prado, "Optimizaci3n de los par3metros del algoritmo de filtrado LiDAR2MDTPlus, para la obtenci3n de Modelos Digitales del Terreno con LiDAR," Ph.D. dissertation, Higher Polytech. Eng. School, Univ. Santiago de Compostela (USC), Santiago de Compostela, Spain, Sep. 2014.
- [38] *The Apache Software Foundation*. Accessed: Jan. 8, 2018. [Online]. Available: <http://hadoop.apache.org/docs/r3.0.0/hadoop-project-dist/hadoop-hdfs/HdfsDesign.html>
- [39] *MongoDB*. Accessed: Jan. 8, 2018. [Online]. Available: <http://www.mongodb.com/>
- [40] *Redis Labs*. Accessed: Jan. 8, 2018. [Online]. Available: <http://redis.io/>
- [41] *DB-Engines Ranking Popularity Ranking of Database Management Systems*. Accessed: Jan. 8, 2018. [Online]. Available: <http://db-engines.com/en/ranking>
- [42] Datastax. *GitHub Datastax/Spark-Cassandra-Connector: DataStax Spark Cassandra Connector*. Accessed: Jan. 22, 2019. [Online]. Available: <https://github.com/datastax/spark-Cassandra-connector>
- [43] Q. Fang, C. Xu, J. Sang, M. S. Hossain, and A. Ghoneim, "Folksonomy-based visual ontology construction and its applications," *IEEE Trans. Multimedia*, vol. 18, no. 4, pp. 702–713, Apr. 2016.
- [44] *The Apache Software Foundation*. Accessed: Mar. 11, 2019. [Online]. Available: <http://storm.apache.org/>
- [45] The Apache Software Foundation. *Apache Flink: Stateful Computations over Data Streams*. Accessed: Mar. 11, 2019. [Online]. Available: <https://flink.apache.org>
- [46] The Apache Software Foundation. *Apache Hadoop*. Accessed: Mar. 11, 2019. [Online]. Available: <http://hadoop.apache.org/>
- [47] S. Lin, H. Chen, and F. Hu, "A workload-driven approach to dynamic data balancing in mongo DB," in *Proc. IEEE Int. Conf. Smart City/SocialCom/SustainCom (SmartCity)*, Chengdu, China, Dec. 2015, pp. 786–791.
- [48] A. Khamparia, S. Pande, D. Gupta, A. Khanna, and A. K. Sangaiah, "Multi level framework for Anomaly detection in Social Networking," Univ. Twente, Data Set, 2020.
- [49] University of Twente. *ISPRS Test Sites*. Accessed: Apr. 27, 2019. [Online]. Available: <https://www.itc.nl/isprs/wgIII-3/filtertest/downloadsites>
- [50] Blue Marble Geographics. *Global Mapper All-in-one GIS Software*. Accessed: May 10, 2019. [Online]. Available: <https://www.bluemarblegeo.com/products/global-mapper.php>
- [51] GmbH. *LAStools*. Accessed: May 15, 2019. [Online]. Available: <https://rapidlasso.com/lastools/>
- [52] X. Yang, T. Zhang, C. Xu, S. Yan, M. S. Hossain, and A. Ghoneim, "Deep relative attributes," *IEEE Trans. Multimedia*, vol. 18, no. 9, pp. 1832–1842, Sep. 2016.
- [53] Y. Hou, S. Garg, L. Hui, D. Nalin K. Jayakody, R. Jin, and M. S. Hossain, "A data security enhanced access control mechanism in mobile edge computing," *IEEE Access*, vol. 8, pp. 136119–136130, 2020.
- [54] W. Min, B.-K. Bao, C. Xu, and M. S. Hossain, "Cross-platform multi-modal topic modeling for personalized inter-platform recommendation," *IEEE Trans. Multimedia*, vol. 17, no. 10, pp. 1787–1801, Oct. 2015.
- [55] K. Kaur, S. Garg, G. Kaddoum, E. Bou-Harb, and K.-K.-R. Choo, "A big data-enabled consolidated framework for energy efficient software defined data centers in IoT setups," *IEEE Trans. Ind. Informat.*, vol. 16, no. 4, pp. 2687–2697, Apr. 2020.
- [56] M. Al-Qurishi, M. Al-Rakhami, A. Alamri, M. Alrubaian, S. M. M. Rahman, and M. S. Hossain, "Sybil defense techniques in online social networks: A survey," *IEEE Access*, vol. 5, pp. 1200–1219, 2017.
- [57] R. Abbasi, N. M. Faseeh Qureshi, H. Hassan, T. Saba, A. Rehman, B. Luo, and A. K. Bashir, "Generalized PVO based dynamic block reversible data hiding for secure transmission using firefly algorithm," *Trans. Emerg. Telecommun. Technol.*, vol. 2019, Jul. 2019, Art. no. e3680.
- [58] S. Garg, K. Kaur, N. Kumar, and J. J. P. C. Rodrigues, "Hybrid deep-learning-based anomaly detection scheme for suspicious flow detection in SDN: A social multimedia perspective," *IEEE Trans. Multimedia*, vol. 21, no. 3, pp. 566–578, Mar. 2019.
- [59] S. Garg, A. Singh, K. Kaur, G. S. Aujla, S. Batra, N. Kumar, and M. S. Obaidat, "Edge computing-based security framework for big data analytics in VANETs," *IEEE Netw.*, vol. 33, no. 2, pp. 72–81, Mar. 2019.
- [60] M. F. Alhamid, M. Rawasdeh, H. Al Osman, M. S. Hossain, and A. El Saddik, "Towards context-sensitive collaborative media recommender system," *Multimedia Tools Appl.*, vol. 74, no. 24, pp. 11399–11428, 2015.



RAHUL MALIK (Member, IEEE) received the M.Tech. degree from NIT Nagpur and the Ph.D. degree from NIT, Jalandhar, Punjab, in 2019. He has expertise in teaching and research and development of seven years. He has more than ten research articles along with book chapters, including more than five articles in SCI indexed journals. His research interests include machine learning, deep learning, image processing, and soft computing.



ADITYA KHAMPARIA received the B.Tech. degree from RGPV, Bhopal, the M.Tech. degree from VIT University, and the Ph.D. degree from Lovely Professional University, Punjab, in May 2018. He did his postdoctoral research at the University of Fortaleza, Brazil, in 2019. He has expertise in teaching, entrepreneurship, and research and development of eight years. He has around 73 research articles along with book chapters, including more than 20 articles in SCI indexed journals with a cumulative impact factor of above 50 to his credit. Additionally, he has authored, edited, and edited six books. His research interests include machine learning, deep learning, educational technologies, and soft computing. He is the Convener and an Organizer of the ICCR Lab Associated Conference Series. Furthermore, he has served in the research field as a keynote speaker/session chair/reviewer/TPC member/guest editor, and many more positions in various conferences and journals. He is an Honorary Editor of the *ICSES Transactions on Image Processing and Pattern Recognition*. He is also associated with various professional bodies, such as ISTE, IAENG, CSI, IET, and ACM.



SAHIL GARG (Member, IEEE) received the Ph.D. degree from the Thapar Institute of Engineering and Technology, Patiala, India, in 2018. He is currently a Postdoctoral Research Fellow at the École de technologie supérieure, Université du Québec, Montréal, Canada. He has many research contributions in the area of machine learning, big data analytics, security and privacy, the Internet of Things, and cloud computing. He has over 50 publications in high-ranked journals and conferences,

including over 25 IEEE TRANSACTIONS/journal articles. He is a member of ACM. He received the IEEE ICC Best Paper Award, in 2018, in Kansas City, MS, USA. He serves/served as the Workshop Chair/Publicity Co-Chair for several IEEE/ACM conferences, including the IEEE INFOCOM, the IEEE GLOBECOM, the IEEE ICC, and the ACM MobiCom. In addition, he also serves as the Workshops and Symposia Officer of the IEEE ComSoc Emerging Technology Initiative on Aerial Communications. He serves as the Managing Editor for the *Human-centric Computing and Information Sciences* journal (Springer). He is also an Associate Editor of the IEEE NETWORK, the IEEE SYSTEM JOURNAL, *Applied Soft Computing* (Elsevier), *Future Generation Computer Systems*, and the *International Journal of Communication Systems* (Wiley). He has guest-edited a number of Special Issues in top-cited journals, including the IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, the IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, the IEEE INTERNET OF THINGS JOURNAL, the IEEE NETWORK, and *Future Generation Computer Systems*.



DEEPAK GUPTA (Member, IEEE) is an eminent Academician and plays versatile roles and responsibilities juggling between lectures, research, publications, consultancy, community service, and Ph.D. and postdoctorate supervision. With 12 years of rich expertise in teaching and two years in the industry, he focuses on rational and practical learning. He has contributed massive literature in the fields of human-computer interaction, intelligent data analysis, nature-inspired computing,

machine learning, and soft computing. He has actively been an organizing end of various reputed international conferences. He has completed his postdoctoral research at Inatel, Brazil, and the University of Valladolid, Spain. He has authored/edited 38 books with national-/international-level publishers (IEEE, Elsevier, Springer, Wiley, and Katson). He has published 131 scientific research publications in reputed international journals and conferences, including 62 SCI indexed journals of IEEE, Elsevier, Springer, and Wiley. He is the Convener and an Organizer of the ICICC Springer Conference Series. He has served as the Editor-in-Chief, a Guest Editor, and an Associate Editor in SCI and various other reputed journals (IEEE, Elsevier, Springer, Wiley, and MDPI). He is the Editor-in-Chief of the OA Journal—*Computers and Quantum Computing and Applications (QCAA)*, an Associate Editor of the *Expert Systems* (Wiley), *Intelligent Decision Technologies* (IOS Press), and the *Journal of Computational and Theoretical Nosciences*, and an Honorary Editor of the *ICES Transactions on Image Processing and Pattern Recognition*. He is also a Book Series Editor of *Smart Sensors Technology for BioMedical Engineering* (Elsevier), *Intelligent Biomedical Data Analysis* (De Gruyter, Germany), and *Computational Intelligence for Data Analysis* (Bentham Science). He is also associated with various professional bodies, such as ISTE, IAENG, IACSIT, SCIEI, ICSES, UACEE, Internet Society, SMEI, IAOP, and IAOP.



BONG JUN CHOI (Senior Member, IEEE) received the B.Sc. and M.Sc. degrees in electrical and electronics engineering from Yonsei University, South Korea, and the Ph.D. degree in electrical and computer engineering from the University of Waterloo, Canada. He was an Assistant Professor at the Department of Computer Science, The State University of New York Korea, South Korea, and concurrently a Research Assistant Professor at the Department of Computer Science, Stony

Brook University, USA. He is currently an Associate Professor at the School of Computer Science and Engineering and at the School of Electronic Engineering, Soongsil University, Seoul, South Korea. His current research interests include energy-efficient networks, distributed mobile wireless networks, smart grid communications, and network security. He is a member of the ACM.

M. SHAMIM HOSSAIN (Senior Member, IEEE) received the Ph.D. degree in electrical and computer engineering from the University of Ottawa, Canada, in 2009. He is currently a Professor at the Department of Software Engineering, College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia. He is also an Adjunct Professor at the School of Electrical Engineering and Computer Science, University of Ottawa. He has authored or coauthored more than 275 publications, including refereed journals, conference papers, books, and book chapters. Recently, he co-edited a book *Connected Health in Smart Cities* (Springer). His research interests include cloud networking, smart environment (smart city and smart health), AI, deep learning, edge computing, the Internet of Things (IoT), multimedia for health care, and multimedia big data. He is a Senior Member of the ACM. He has served as the Co-Chair, the General Chair, the Workshop Chair, the Publication Chair, and a TPC Member for over 12 IEEE and ACM conferences and workshops. He is currently the Co-Chair of the 3rd IEEE ICME Workshop on Multimedia Services and Tools for Smart-Health (MUST-SH 2020). He was a recipient of a number of awards, including the Best Conference Paper Award and the 2016 *ACM Transactions on Multimedia Computing, Communications and Applications (TOMM)* Nicolas D. Georganas Best Paper Award. He is on the Editorial Board of the IEEE TRANSACTIONS ON MULTIMEDIA, the IEEE MULTIMEDIA, the IEEE NETWORK, the IEEE WIRELESS COMMUNICATIONS, IEEE ACCESS, the *Journal of Network and Computer Applications* (Elsevier), and the *International Journal of Multimedia Tools and Applications* (Springer). He also serves as a Lead Guest Editor for the IEEE NETWORK, the *ACM Transactions on Internet Technology*, the *ACM TOMM*, and the *Multimedia Systems Journal*. He serves/served as a Guest Editor for *IEEE Communications Magazine*, the IEEE NETWORK, the IEEE TRANSACTIONS ON INFORMATION TECHNOLOGY IN BIOMEDICINE (currently JBHI), the IEEE TRANSACTIONS ON CLOUD COMPUTING, *Future Generation Computer Systems* (Elsevier), the *International Journal of Multimedia Tools and Applications* (Springer), and *Cluster Computing* (Springer).

...