

## Assessment of explainable tree-based ensemble algorithms for the enhancement of Copernicus digital elevation model in agricultural lands

Chukwuma Okolie, Adedayo Adeleke, Jon Mills, Julian Smit, Ikechukwu Maduako, Hossein Bagheri, Tom Komar & Shidong Wang

To cite this article: Chukwuma Okolie, Adedayo Adeleke, Jon Mills, Julian Smit, Ikechukwu Maduako, Hossein Bagheri, Tom Komar & Shidong Wang (12 Apr 2024): Assessment of explainable tree-based ensemble algorithms for the enhancement of Copernicus digital elevation model in agricultural lands, International Journal of Image and Data Fusion, DOI: [10.1080/19479832.2024.2329563](https://doi.org/10.1080/19479832.2024.2329563)

To link to this article: <https://doi.org/10.1080/19479832.2024.2329563>



© 2024 The Author(s). Published by Informa UK Limited, trading as Taylor & Francis Group.



Published online: 12 Apr 2024.



Submit your article to this journal [↗](#)



Article views: 276



View related articles [↗](#)



View Crossmark data [↗](#)

# Assessment of explainable tree-based ensemble algorithms for the enhancement of Copernicus digital elevation model in agricultural lands

Chukwuma Okolie<sup>a,b,c</sup>, Adedayo Adeleke<sup>d</sup>, Jon Mills<sup>c</sup>, Julian Smit<sup>e</sup>,  
Ikechukwu Maduako<sup>f</sup>, Hossein Bagheri<sup>g</sup>, Tom Komar<sup>c</sup> and Shidong Wang<sup>c</sup>

<sup>a</sup>Division of Geomatics, University of Cape Town, Cape Town, South Africa; <sup>b</sup>Department of Surveying and Geoinformatics, University of Lagos, Lagos, Nigeria; <sup>c</sup>School of Engineering, Newcastle University, Newcastle upon Tyne, UK; <sup>d</sup>Department of Geography, Geoinformatics and Meteorology, University of Pretoria, Pretoria, South Africa; <sup>e</sup>Department of Civil Engineering and Geomatics, Cape Peninsula University of Technology, Cape Town, South Africa; <sup>f</sup>Office of Global Innovation, UNICEF, New York, USA; <sup>g</sup>Faculty of Civil Engineering and Transportation, University of Isfahan, Isfahan, Iran

## ABSTRACT

There has been a rapid evolution of tree-based ensemble algorithms which have outperformed deep learning in several studies, thus emerging as a competitive solution for many applications. In this study, ten tree-based ensemble algorithms (random forest, bagging meta-estimator, adaptive boosting (AdaBoost), gradient boosting machine (GBM), extreme gradient boosting (XGBoost), light gradient boosting (LightGBM), histogram-based GBM, categorical boosting (CatBoost), natural gradient boosting (NGBoost), and the regularised greedy forest (RGF)) were comparatively evaluated for the enhancement of Copernicus digital elevation model (DEM) in an agricultural landscape. The enhancement methodology combines elevation and terrain parameters alignment, with feature-level fusion into a DEM enhancement workflow. The training dataset is comprised of eight DEM-derived predictor variables, and the target variable (elevation error). In terms of root mean square error (RMSE) reduction, the best enhancements were achieved by GBM, random forest and the regularised greedy forest at the first, second and third implementation sites respectively. The computational time for training LightGBM was nearly five-hundred times faster than NGBoost, and the speed of LightGBM was closely matched by the histogram-based GBM. Our results provide a knowledge base for other researchers to focus their optimisation strategies on the most promising algorithms.

## ARTICLE HISTORY

Received 13 August 2023  
Accepted 19 February 2024

## KEYWORDS

Copernicus; global digital elevation model; LiDAR; machine learning; tree-based ensembles; bagging; boosting; gradient boosting; explainability; partial dependence

## 1. Introduction

Several studies have been devoted to the enhancement of satellite-derived digital elevation models (DEMs). These wide-area DEMs cover entire states or continents and are usually supplied in the standard 2.5-dimension digital raster format (Schindler *et al.* 2011). They are

**CONTACT** Chukwuma Okolie  [oklchu002@myuct.ac.za](mailto:oklchu002@myuct.ac.za)  Division of Geomatics, University of Cape Town, Cape Town, South Africa

© 2024 The Author(s). Published by Informa UK Limited, trading as Taylor & Francis Group.

This is an Open Access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. The terms on which this article has been published allow the posting of the Accepted Manuscript in a repository by the author(s) or with their consent.

produced by a variety of methods (e.g. laser scanning, photogrammetric processing of aerial and satellite images, and synthetic aperture radar interferometry). DEMs have numerous applications in environmental modelling (Olusina and Okolie 2018), and their accuracies are influenced by several factors including the prevailing land cover and terrain irregularities, source data attributes, sensor distortions, and errors inherent in the DEM production methods (Okolie and Smit 2022). These errors compromise their quality and adequacy for hydrological and environmental applications (e.g. flood and watershed modelling) where precise and accurate terrain information is needed. Consequently, the development of methods for the enhancement of global DEMs has become an active area of research. By analysing the relationship between vertical accuracy and land cover or terrain parameters, the vertical error can be modelled and minimised. Consequently, several studies have emerged on DEM enhancement using a variety of methods (e.g. Pakoksung and Takagi 2016, Bagheri *et al.* 2018, Olajubu *et al.* 2021, Preety *et al.* 2022). Generally, these enhancements improve the DEMs through the reduction of the vertical bias or error.

Machine learning techniques are increasingly being applied for DEM enhancement. Table 1 presents an overview of previous studies that have applied machine learning for DEM enhancement. Wendi *et al.* (2016) used an artificial neural network (ANN) to enhance the accuracy of the 90 m Shuttle Radar Topography Mission (SRTM) DEM. The ANN exploited the interdependence between the DEM errors and satellite imagery spectral signatures for various land cover. When compared with a reference DEM, the enhancement led to a root mean square error (RMSE) reduction of ~68% (from 13.9 m to 4.4 m) and ~52% (from 14.2 m to 6.7 m) at two separate sites. In another study, Kulp and Strauss (2018) incorporated additional variables into an ANN such as slope, neighbourhood elevation values, vegetation cover indices, population density and local SRTM deviations. In their study, training, validation and testing data were built at sites with known actual SRTM error in coastal areas of the United States (US) and Australia. The performance assessment proceeded with US and Australia testing sets, as well as global ICESat measurements. The adjustment system reduced the mean vertical bias in the coastal US from 3.67 m to less than 0.01 m, and in Australia from 2.49 m to 0.11 m. RMSE was cut by roughly one-half at both locations, from 5.36 m to 2.39 m in the US, and from 4.15 m to 2.46 m in Australia. With reference to Ice, Cloud, and land Elevation Satellite (ICESat) data,

**Table 1.** Overview of previous machine learning methods used for DEM enhancement.

| Reference                   | Name of machine learning algorithm                                                                           |
|-----------------------------|--------------------------------------------------------------------------------------------------------------|
| Kasi <i>et al.</i> (2020)   | Genetic programming (GP), artificial neural network (ANN)                                                    |
| Chen <i>et al.</i> (2020)   | Generalised regression neural network (GRNN), random forest (RF), and back propagation neural network (BPNN) |
| Ma <i>et al.</i> (2020)     | BPNN                                                                                                         |
| Salah (2021)                | ANN                                                                                                          |
| Kim <i>et al.</i> (2019)    | ANN                                                                                                          |
| Hu and Ji (2022)            | BPNN, RF, gradient tree lifting and ensemble learning                                                        |
| Wendi <i>et al.</i> (2016)  | Feed forward neural network (FNN)                                                                            |
| Hawker <i>et al.</i> (2022) | RF                                                                                                           |
| Girohi and Bhardwaj (2022)  | FNN                                                                                                          |
| Kim <i>et al.</i> (2021)    | ANN                                                                                                          |
| Kim <i>et al.</i> (2020)    | ANN                                                                                                          |
| Liu <i>et al.</i> (2021)    | RF                                                                                                           |
| Kulp and Strauss (2018)     | ANN                                                                                                          |

they estimated the global bias of SRTM to fall from 1.88 m to  $-0.29$  m. Kulp and Strauss (2018) went on to posit that their method could be effectively applied to all land cover types, including dense urban development. In another study, Kim *et al.* (2020) attempted to cater for the peculiarities of urban areas in SRTM DEM improvement by incorporating building footprints along with multispectral imagery, SRTM DEM and a reference DEM into an ANN system. The performance of the DEM improvement scheme was tested over two dense urban cities in Nice (France) and Singapore. The improved SRTM DEM showed significantly better results than the original SRTM DEM, with about 38% RMSE reduction.

Tree-based ensemble learning algorithms have received significant attention as one of the most reliable and broadly applicable classes of machine learning approaches. These algorithms present several advantages such as interpretability, less data preparation, versatility, and the ability to handle non-linear and complex relationships. Decision trees provide a straightforward interpretation and understanding of the relationships between objects at different levels of detail (Miao *et al.* 2012). Thus, it is easier to interpret the logical rules followed by a decision tree than for example, the numeric weights of the connections between the nodes in a neural network (Kotsiantis 2013). Decision trees have a high tolerance for multicollinearity (Climent *et al.* 2019, Han *et al.* 2019, Pham and Ho 2021). If there are highly correlated features or variables, decision trees are inherently able to choose only one of the features when deciding upon a split (Climent *et al.* 2019). Other advantages of decision trees include: (i) automatic consideration of potential interactive effects among predictors, automatic exclusion of irrelevant features, and intuitive guidance for application of the results in decision making (e.g. cut-off points and ranked priorities for predictor variables) (Han *et al.* 2019). There has been a rapid adoption of tree-based supervised machine learning algorithms for addressing challenges in the remote sensing and geospatial science community, e.g. landslide susceptibility mapping (Kavzoglu and Teke 2022), mapping of glacial lakes, geological mapping (Albert and Ammar 2021), mapping tree canopy cover and aboveground biomass (Karlson *et al.* 2015), geoscience data analysis and modelling (Talebi *et al.* 2022), and wetland classification. Furthermore, gradient-boosted decision trees (GBDTs) have emerged as a winning solution and have outperformed deep learning algorithms in some studies (e.g. Kadra *et al.* 2021, Borisov *et al.* 2022, Shwartz-Ziv and Armon 2022). Dusseau *et al.* (2023) performed some initial testing of a GBDT (LightGBM) and found that it was superior to a neural network in reducing DEM vertical error.

A literature survey revealed a few studies using tree-based algorithms such as random forest for DEM enhancement, e.g. urban correction of MERIT DEM (Liu *et al.* 2021), vegetation correction of SRTM (Yanjun *et al.* 2015), enhancement of SRTM DEM for flood modelling (Meadows and Wilson 2021), and the production of a globally corrected version of Copernicus DEM, known as FABDEM (Hawker *et al.* 2022). Despite these modest achievements, the full potential of tree-based algorithms is yet to be fully tested or materialised in the field of remote sensing, specifically for DEM enhancement. Even the advanced gradient boosting algorithms such as XGBoost which outperformed deep learning in several challenges have not been fully exploited for this purpose. Moreover, the performance of algorithms for DEM enhancement in agricultural lands has not received extensive attention, since many of the previous studies were particularly focused on urban areas.

In this study, we compare the performance of ten tree-based ensemble machine learning algorithms (shown in Table 2) for the enhancement of Copernicus GLO-30 DEM in agricultural lands of Cape Town, South Africa. Most of the assessed algorithms are



either bagging or boosting ensembles. Bagging is “a method to train an ensemble where each constituent model trains on a random subset of training examples sampled with replacement (Google for Developers 2023). Essentially, multiple versions of a predictor are used for generating an aggregated predictor (Breiman 1996). This aggregation of multiple results into a single prediction can reduce the variance in the final results (Geurts *et al.* 2006, Zhang *et al.* 2022). Bagging is a generic concept and can also be applied on other machine learning algorithms. On the other hand, Boosting uses a forward stagewise approach to transform weak learners into strong learners by increasing the weights of training samples in successive iterations (Zhang *et al.* 2022). The final output is derived by synergising the results from all the iterations using a weighted sum (Elith *et al.* 2008, Zhang *et al.* 2022). Bagging tries to solve the overfitting problem while Boosting tries to reduce bias. The tree-based ensemble algorithms are explained further in Section 2.4.

Several studies have shown that topography affects the spatial distribution of soil types, biomass, organic matter, and affects crop yields (Yang *et al.* 2014, Nie *et al.* 2019, Ma *et al.* 2020). Thus, the improvement of DEM accuracy in agricultural lands is an important concern.

The unique strength of this study is that the comparison of these algorithms helps to simplify the challenge of model selection by other researchers faced with similar applications. Moreover, the study provides insights on the practical significance of the different algorithms based on their computational complexities. A comparative analysis of these tree-based algorithms is required to inform their knowledge-based application in remote sensing tasks and problems, especially DEM enhancement. The increasing availability and adoption of wide-area, multi-source and multi-sensor DEMs has provided tremendous opportunities for the application of machine learning in their enhancements.

To our knowledge, this is the first study to implement such a wide and comprehensive comparison of tree-based algorithms for DEM enhancement. The assessment is an impetus to stimulate further work aimed at achieving more significant accuracy improvements for open-access global DEMs. This paper is divided into four sections. Section 1 presents the introduction and aim of the study; section 2 discusses the materials and methods. The results are presented and analysed in section 3 while the discussion and conclusion follow in sections 4 and 5 respectively.

## 2. Materials and methods

The enhancement methodology combines elevation and terrain parameters alignment, with **feature-level fusion (ensemble learning)** into a DEM enhancement workflow.

**Table 2.** Classification of the ten compared algorithms.

|                                                                                                                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                                                                                                                       |
|-----------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------|
| Bagging: <ul style="list-style-type: none"> <li>● Random Forest</li> <li>● Bagging</li> <li>Meta-estimator</li> </ul> | Boosting: <ul style="list-style-type: none"> <li>● Adaptive Boosting (AdaBoost)</li> <li>Gradient boosting:               <ul style="list-style-type: none"> <li>● Gradient Boosting Machine (GBM)</li> <li>● Extreme Gradient Boosting Machine (XGBoost)</li> <li>● Light Gradient Boosting Machine (LightGBM)</li> <li>● Histogram-based Gradient Boosting</li> <li>● Categorical Boosting (CatBoost)</li> </ul> </li> <li>Natural gradients:               <ul style="list-style-type: none"> <li>● Natural Gradient Boosting (NGBoost)</li> </ul> </li> </ul> | Improvement to gradient boosting: <ul style="list-style-type: none"> <li>● Regularised Greedy Forest (RGF)</li> </ul> |
|-----------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------|

Figure 1 shows the workflow diagram of the assessment methodology. The process starts with the selection of two open-access DEMs: Copernicus GLO-30 and the City of Cape Town (CCT) airborne Light Detection and Ranging (LiDAR)-derived DEMs. The DEMs are harmonised into the same coordinate system and a height error map is derived by calculating the differences between corresponding elevations in both DEMs. To characterise the influence of the terrain on the elevation error, the following input variables are incorporated into the ensemble learning framework: elevation, slope, aspect, surface roughness, topographic position index (TPI), terrain ruggedness index (TRI), terrain surface texture (TST), and vector ruggedness measure (VRM). The terrain parameters were generated within the QGIS 3.28.2 and SAGA GIS 7.8.2 software environments, while the tree-based algorithms were implemented in the Google Collaboratory (Colab) cloud computing environment. The workflow is explained in detail in the following sections.

## 2.1. Study area

Cape Town is the most south-western city in South Africa, with a land area of approximately 400 km<sup>2</sup> (Orimoloye *et al.* 2019). The city is situated in the Western Cape Province and has a high landscape-level diversity with rivers, coastal areas and wetlands (Goodness and Anderson 2013). The sites (shown in Figure 2) are selected from agricultural lands with few settlements located along the floodplain of River Diep. The Diep River is a sub-catchment within the Berg Catchment area and flows towards the Table Bay into the sea (Drakenstein River Environmental Management Plan 2008). The river catchment is low-lying, and surrounded by industries and factories, informal settlements and farms (DWS 2020, Gqomfa *et al.* 2022). There are several topographic changes within the low-lying river catchment as highlighted in Figures 2 and 3, and this further justifies the need for better DEM accuracy especially for agricultural site assessment studies. Table 3 shows the distribution of sample points in the agricultural lands. After the training and testing phase, an independent

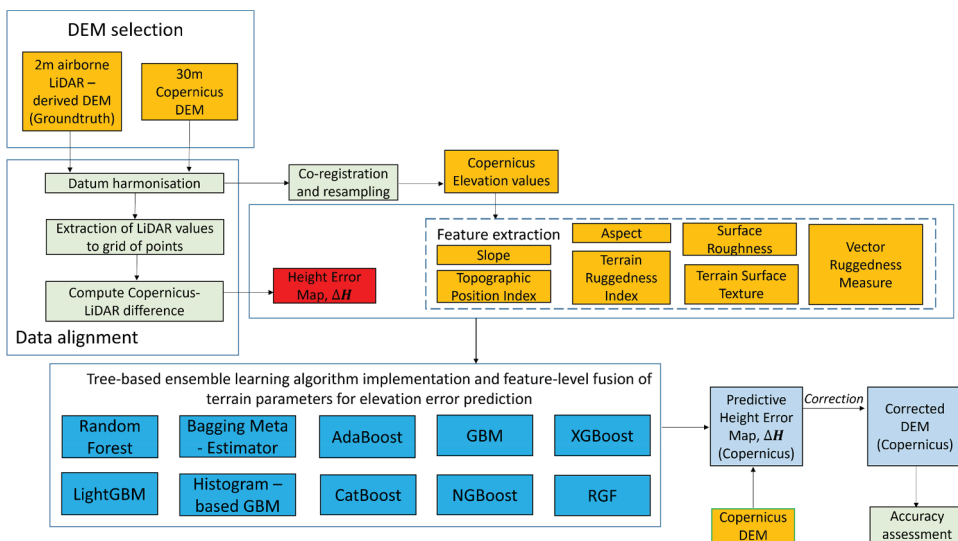


Figure 1. Workflow diagram of the assessment methodology.

evaluation was carried out at model implementation sites. All the selected sites (training/test and implementation) have similar elevation and terrain characteristics (see [Figure 3](#)).

## 2.2. Datasets and terrain parameters

### 2.2.1. Copernicus GLO-30 DEM

Copernicus DEM (released in 2020) is derived from the WorldDEM data. The WorldDEM data product is based on the radar satellite data which was acquired during the TanDEM-X Mission (ESA 2020b). The primary objective of the TanDEM-X mission was the generation of a global coverage DEM based on Interferometric Synthetic Aperture Radar (InSAR) in HRTI-3 standards. The duration of the TanDEM-X data acquisition was between December 2010 and January 2015. The Copernicus GLO-30 dataset has a grid spacing of one arc-second (30 m) and a standard extent of

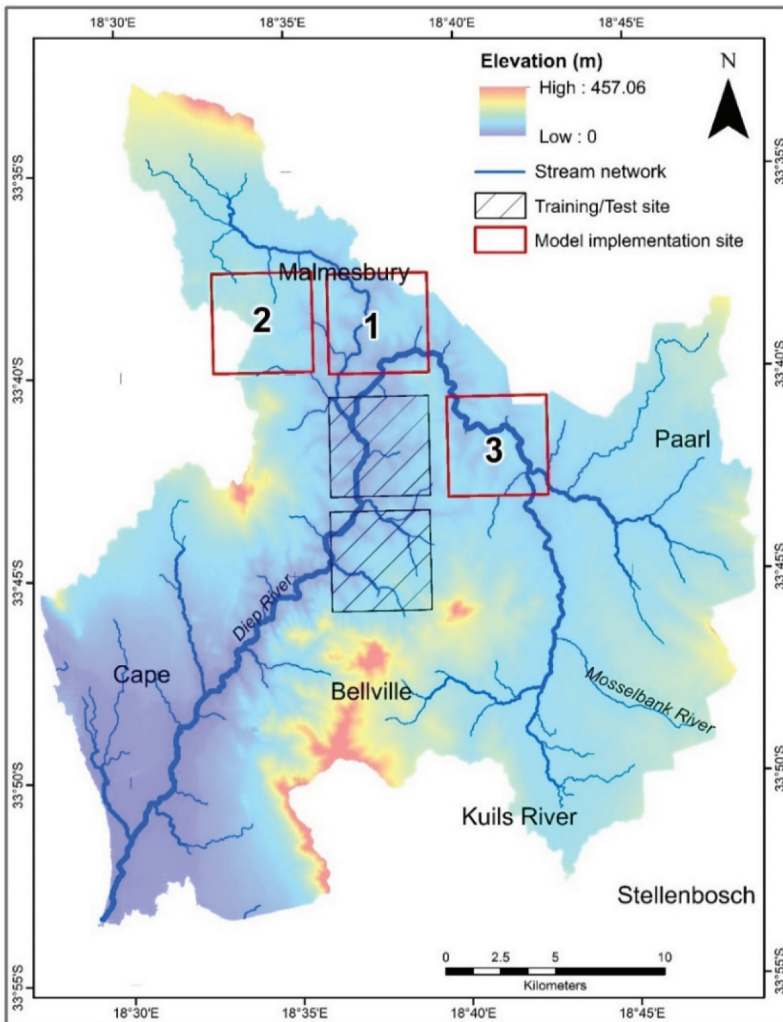
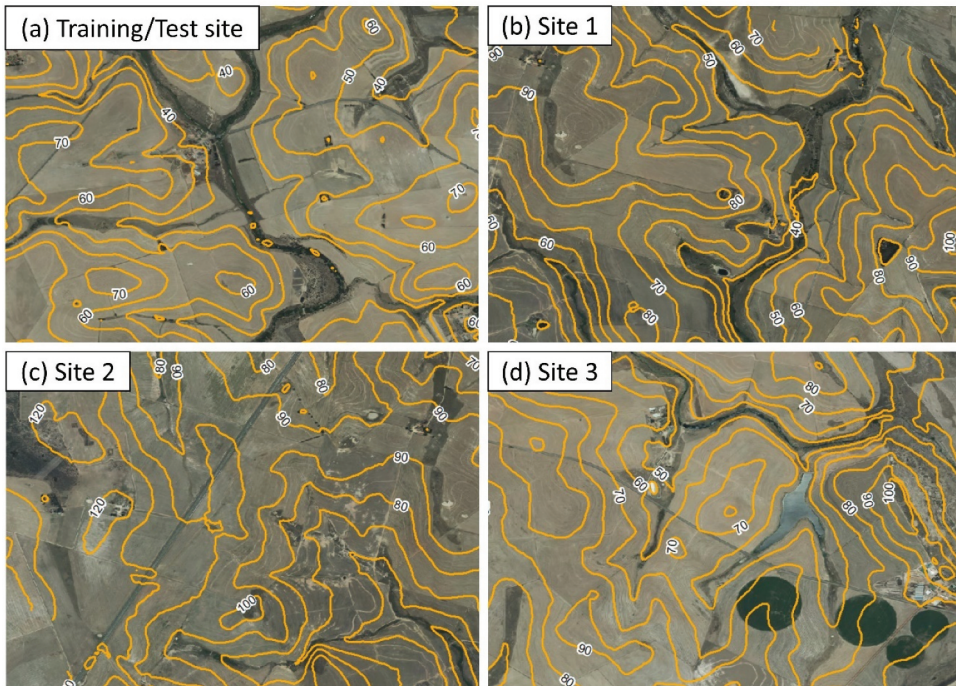


Figure 2. Map showing the selected sites in Cape Town, South Africa.



**Figure 3.** View of agricultural lands selected for assessment – (a) training/test site (b) 1st model implementation site (c) 2nd model implementation site (d) 3rd model implementation site (aerial imagery, January 2023; source: city of Cape Town). Contour interval: 10 m.

**Table 3.** The distribution of sample points at the selected sites.

| Site ID                 | Number of points | Area (km <sup>2</sup> ) |
|-------------------------|------------------|-------------------------|
| Training/testing        | 46177            | 41.4                    |
| 1st implementation site | 21539            | 20.7                    |
| 2nd implementation site | 23109            | 20.7                    |
| 3rd implementation site | 22366            | 20.7                    |

1° x 1°. For this study, the floating point Defense Gridded Elevation Data (DGED) format of the DEM was adopted. The Copernicus DEM has been assessed with ICESat-2 measurements, which indicate absolute vertical uncertainties of ~1–3 m (ESA 2020b). Some essential characteristics of the DEM are summarised in Table 4.

### 2.2.2. Cape Town LiDAR DEM

The City of Cape Town (CCT) airborne LiDAR-derived DEM was acquired from the Information and Knowledge Management Department of the City of Cape Town. The 2 m DEM is generated from the LiDAR point cloud, at a maintenance cycle of 3 years. The point density is 2–3 points/m<sup>2</sup>, and the point cloud has a height accuracy of 0.15 m. The data acquisition was conducted from 2018–2021, and the DEM is spatially referenced to the Hartebeesthoek94 horizontal co-ordinate system, while the height reference is the South Africa Land Levelling Datum (SAGEOID2010).

**Table 4.** Characteristics of Copernicus GLO-30 DEM.

| <b>Copernicus GLO-30 DEM characteristics</b> |                                                             |
|----------------------------------------------|-------------------------------------------------------------|
| Tiling                                       | 1 deg x 1 deg                                               |
| Projection                                   | Geographic Coordinates                                      |
| Coordinate reference system                  | WGS84-G1150 (Horizontal)<br>EGM2008 (Vertical)              |
| Absolute vertical accuracy                   | <4m (90% linear error)                                      |
| Relative vertical accuracy                   | <2m (slope ≤20%)<br>< 4m (slope > 20%)                      |
|                                              | (90% linear point-to-point error within an area of 1° x 1°) |
| Absolute horizontal accuracy                 | <6m (90% circular error)                                    |

Source: ESA (2020a).

### 2.2.3. Terrain parameters

The parameters include slope, aspect, surface roughness, topographic position index (TPI), terrain ruggedness index (TRI), terrain surface texture (TST), and vector ruggedness measure (VRM). The theoretical underpinnings of these parameters are available in the existing literature, therefore only a brief discussion is provided here. The slope function identifies the rate of maximum change in z-value from each cell of the DEM. Aspect identifies the downslope direction of the maximum rate of change in elevation value from each cell to its neighbours. Terrain roughness is the degree of variation of the z-axis across the terrain, in a defined area and at a defined scale (Department of Environment and Science, Queensland 2020). TPI compares the elevation of each cell in a DEM to the mean elevation of a specified neighbourhood around that cell (Vinod 2017). TRI deals with the degree of elevation difference between adjacent grid cells of a DEM (USNA 2022). The VRM measures terrain ruggedness as the variation in three-dimensional (3D) orientation of raster grid cells within a neighbourhood (Welty and Jeffries 2018). The elevation errors or differences ( $\Delta H$ ) between the Copernicus DEM and reference CCT LIDAR DEM was calculated as follows:

$$\Delta H = H_{\text{Copernicus}} - H_{\text{RefDEM}} \quad (1)$$

where,

$H_{\text{Copernicus}}$  = individual elevations from Copernicus GLO-30 DEM.

$H_{\text{RefDEM}}$  = individual elevations from CCT LiDAR-derived DEM.

### 2.3. Data preparation

The horizontal spatial reference of the CCT LiDAR and Copernicus DEMs were harmonised into the Universal Transverse Mercator (UTM) projection in WGS84, and the LiDAR elevations were transformed from the South Africa Land Levelling datum (LLD) to EGM2008, in conformity with the vertical datum of Copernicus DEM. A grid of points was generated and elevation values from the LiDAR and Copernicus DEMs that coincided with the points were extracted and recorded in an attribute table. Thus, the elevation error ( $\Delta H$ ) was calculated by subtracting the LiDAR elevations from the Copernicus elevations. Subsequently, the elevation error values were converted to a raster format. The elevation values, along with the values of the elevation error, and terrain parameters were extracted from the rasters to csv files. This resulted in the final set of points used for model training and testing.



## 2.4. Tree-based ensemble algorithms

### 2.4.1. Bagging algorithms

**Random forest (RF)** was proposed by L. Breiman in 2001. It aggregates the predictions of several decision trees with randomly allocated features (Biau and Scornet 2016). The previous bagging tree method was upgraded to the random forest algorithm by Breiman (2001), thus injecting more randomness in growing base trees and improving its predictive power and capability (Miao *et al.* 2012). The following steps are used in the **bagging regressor (bagging meta-estimator)** algorithm: (i) the original dataset is broken down into random subsets (ii) a base estimator is specified by the user and fitted on the subsets, and (iii) the predictions are integrated to generate the final result (Singh 2023). By default, the base estimator is a decision tree. Bagging methods are known to reduce overfitting, and are best applied to strong and complex models unlike boosting methods which work better with weak models (Scikit-Learn 2023).

### 2.4.2. Boosting algorithms

For a long time, **AdaBoost (Adaptive boosting)** was a well-known and widely applied boosting algorithm by practitioners and researchers (Schapire 2003, Barrow and Crone 2016). The inspiration for **AdaBoost** was a technique that combines the outputs of several weak classifiers to produce a strong learner (Hastie *et al.* 2009, Miao *et al.* 2012). Unlike bagging, AdaBoost grows an ensemble of trees through successive reweighting of training samples. It is resistant to overfitting but does not handle noise well (Bauer and Kohavi 1999, Miao *et al.* 2012).

For many years, gradient boosting has been the primary approach for learning problems with heterogeneous features, complex dependencies and noisy data (Roe *et al.* 2005, Caruana and Niculescu-Mizil 2006, Zhang and Haghani 2015, Prokhorenkova *et al.* 2018). Essentially, it involves the application of gradient descent in a functional space to construct an ensemble predictor (Prokhorenkova *et al.* 2018). It is backed by strong theoretical underpinnings that show how strong predictors can be constructed through the combination of base predictors (weaker models) in a greedy manner (Kearns and Valiant 1994, Prokhorenkova *et al.* 2018). GBDTs have excelled in a myriad of applications with state-of-the-art results (Chen and Guestrin 2016).

The **gradient boosting machine (GBM)** regressor adopts an additive model built in a forward stage-wise fashion; and enables the optimisation of differentiable loss functions (Scikit-Learn 2023). **Extreme Gradient Boosting (XGBoost)** is an end-to-end gradient booster that consecutively builds decision trees as each tree tries to improve the performance of the previous tree (Chen and Guestrin 2016, Safaei *et al.* 2022). It parallelises the training process of each tree and speeds up the training (Safaei *et al.* 2022). **Light Gradient Boosting Machine (LightGBM)** is an improved GBDT framework that was introduced by Microsoft (Ke *et al.* 2017) to overcome the scalability and efficiency limitations of previous GBDTs. LightGBM has less memory occupation and a faster training speed (Wang and Wang 2020, Microsoft Corporation 2022). The main features of LightGBM include the gradient-based one-side sampling (GOSS), exclusive feature bundling (EFB), and a histogram and leaf-wise growth strategy. The histogram-based gradient boosting regression tree (**histogram-based GBM**) is reputedly faster than the gradient boosting regressor for large datasets, and also supports missing values (Scikit-Learn 2023).

It can reduce the training time without losing accuracy (Padhi *et al.* 2021). **Categorical Boosting (CatBoost)** which debuted in 2018 is well-suited for problems involving heterogeneous and categorical data (Hancock and Khoshgoftaar 2020). It incorporates two important advances: ordered boosting (a permutation-driven substitute to the traditional gradient boosting), and a novel algorithm for the processing of categorical features (Prokhorenkova *et al.* 2018). **Natural Gradient Boosting (NGBoost)** was developed for generic probabilistic prediction. Instead of returning a point estimate, conditional on covariates, it returns a full probability distribution over the outcome space, conditional on the covariates (Duan *et al.* 2020). NGBoost has applications in regression, classification and survival prediction (Kavzoglu and Teke 2022).

### 2.4.3. Regularised greedy forest

The Regularized Greedy Forest (RGF) was proposed by Johnson and Zhang (2014). RGF learns a non-linear function through the adoption of an additive model over non-linear decision rules. It incorporates tree-structured regularisation into the learning and uses a fully corrective regularised greedy algorithm (Johnson and Zhang 2014). In several machine learning challenges, RGF has outperformed GBDTs (Joseph 2020).

## 2.5. Model implementation and DEM correction

The tree-based ensembles were implemented with their default hyperparameters. A hyperparameter is 'a parameter whose value is given by the user and used to control the learning process' (Mariani and Sipper 2022). Their values 'control the learning process and determine the values of model parameters that a learning algorithm ends up learning' (Nyuytiyimbii 2020). Default hyperparameters are adopted for a baseline performance comparison of the ten algorithms. This approach is foundational for future optimisation efforts. The training data includes the following input variables: elevation, slope, aspect, surface roughness, TPI, TRI, TST, and VRM; and the target variable, elevation error ( $\Delta H$ ). Incorporating this comprehensive set of input variables enables a more robust DEM enhancement framework. All the variables were converted from raster to csv format and split into 80% (training) and 20% (testing). The implementation was done within the Google Collaboratory (Colab) cloud computing environment using Python scripting, the scikit-learn machine learning library and other open-source libraries/packages. Google Colab enables the writing and execution of Python code through web browsers, and is well suited for machine learning (Google 2023). The default CPU for the Colab processing environment has the following specifications – Intel Xeon CPU @ 2.20 GHz, 13 GB RAM, Tesla K80 accelerator, and 12 GB GDDR5 VRAM (Das 2022). Utilising the Google Collaboratory cloud computing environment for implementation highlights a practical and accessible approach for computational tasks in remote sensing. A list of Python packages used, and their descriptions are presented in Table 5. Summarily, the data was passed into the model regressors for training and subsequently, the trained algorithms were evaluated at three implementation sites with very similar characteristics (shown in Figure 3). To derive the corrected elevations at the implementation sites, the predicted elevation errors were subtracted from the original elevations (i.e.  $DEM_{Corrected} = DEM_{Original} - \Delta H_{Predicted}$ ).



**Table 5.** List of python packages/libraries used.

| Package/library | Function                                                                                                  |
|-----------------|-----------------------------------------------------------------------------------------------------------|
| Scikit-learn    | Supervised and unsupervised learning, model fitting, model selection and model evaluation.                |
| Matplotlib      | Interactive visualisations.                                                                               |
| Pandas          | Data analysis and manipulation.                                                                           |
| Numpy           | Provides support for large, multi-dimensional arrays and matrices, and high-level mathematical functions. |
| Joblib          | Provides tools for lightweight pipelining.                                                                |

## 2.6. Model performance indicators

For a comprehensive evaluation of model accuracy and reliability, the following regression error metrics were adopted: root mean squared error (RMSE), mean absolute error (MAE), and median absolute error (MedAE). The RMSE is an excellent general-purpose error metric for assessing numerical predictions (Christie and Neill 2022). These error measurement metrics define the prediction accuracy and allow for the monitoring of outliers in predictions. However, depending on the type and volume of the data, as well as the nature of the predictions, different error metrics can be used to interpret the model results. The MAE is the average of the difference between the original values and the predicted values. It gives the measure of how far the predictions were from the actual output. It is quite robust to outliers, hence very useful where some of the variables are prone to outliers and biases. In terms of interpretability, it is the easiest to explain. However, it does not show the direction of the error i.e. under-prediction or over-prediction.

The RMSE is a standard regression measure that usually punishes larger errors more than smaller ones. The score ranges from 0 in a perfect match, to arbitrarily large values as the predictions become worse. The main difference is that the RMSE penalises more strongly the large errors. It is the square root of mean squared error (MSE) which is calculated as the average of the squared forecast error values. Squaring the forecast error values forces them to be positive hence putting more weight on large errors. In effect, the score gives worse performance to those algorithms that make large wrong forecasts. The MedAE sometimes can be used interchangeably with the MAE because the MedAE is also robust to outliers and suitable for use cases where some of the variables are prone to outliers and biases. The error is calculated by taking the median of all absolute differences between the original values and the predicted values. If  $\hat{y}_i$  is the predicted elevation error of the  $i$ -th sample and  $y_i$  is the corresponding true value of the elevation error for a total of  $n$  samples, the estimated metrics are defined as (Chai and Draxler 2014, Scikit-Learn 2023):

$$\sum_{i=1}^n (y_i - \hat{y}_i)^2 = \sum_{i=1}^n \epsilon_i^2 \quad (2)$$

$$\text{MSE}(y, \hat{y}) = \frac{1}{n} \sum_{i=0}^{n-1} (y_i - \hat{y}_i)^2 \quad (3)$$

$$\text{MAE}(y, \hat{y}) = \frac{1}{n} \sum_{i=0}^{n-1} |y_i - \hat{y}_i| \quad (4)$$

$$\text{MedAE}(y, \hat{y}) = \text{median}(|y_1 - \hat{y}_1|, \dots, |y_n - \hat{y}_n|) \quad (5)$$

The RMSE was further derived from the MSE function.

## 2.7. Evaluation of model computation time

Computational complexity is an important criterion in the analysis of machine learning algorithms. It is 'a computer science concept that focuses on the amount of computing resources needed for particular kinds of tasks' (Rouse 2019). Analysing the time complexity of machine learning algorithms could facilitate the selection and deployment of the most efficient (and appropriate) model for a particular dataset (Singh 2023). According to Pushp (2023), 'time complexity measures the time taken for the algorithm to execute.' An algorithm with faster training time is more efficient (Pushp, 2023). For insights on the time complexity of the algorithms, the computation time of the tree-based algorithms was compared.

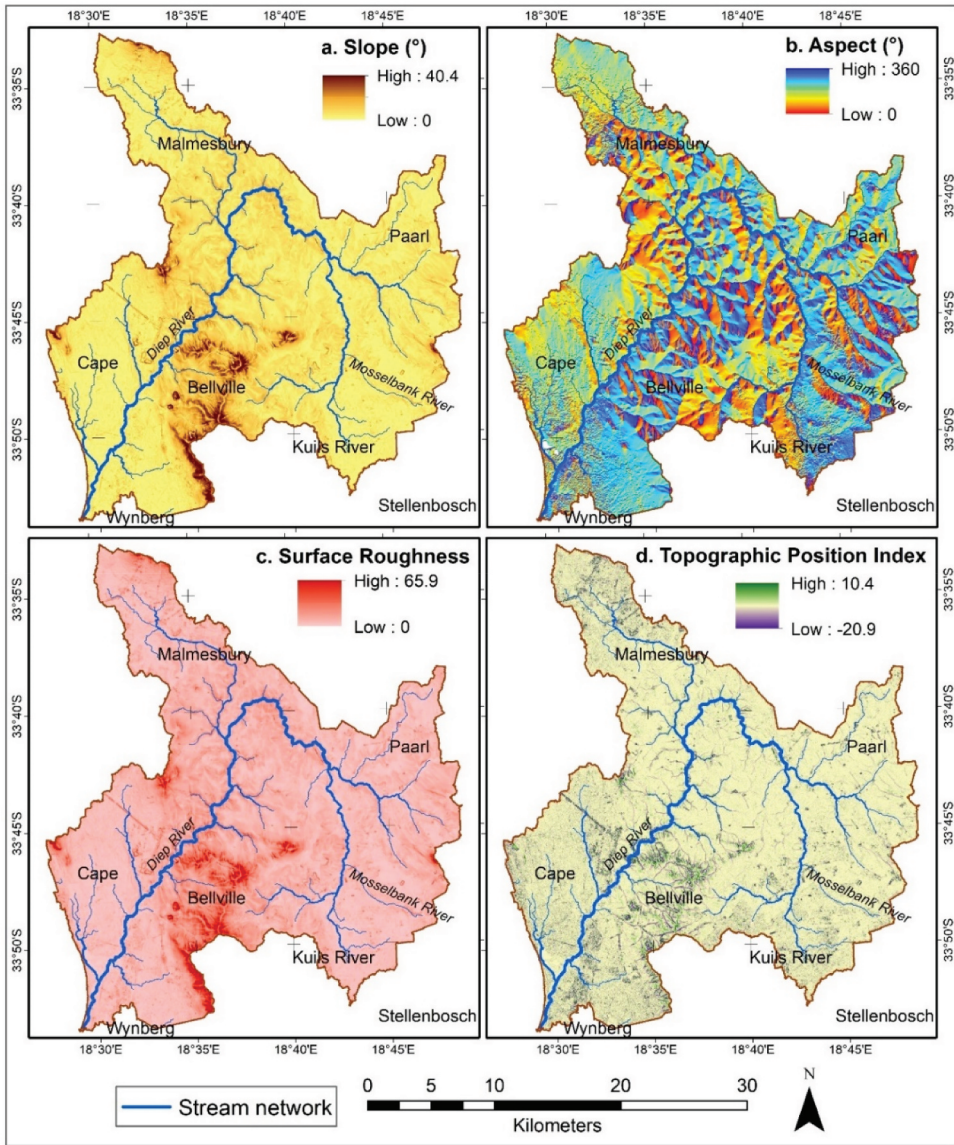
## 2.8. Model explainability

Explainability of machine learning algorithms has been a source of concern in the Artificial Intelligence (AI) community and is becoming a major requirement for their deployment. Explainability enables the identification of cause-and-effect relationships within the inputs and outputs of the system (Linardatos *et al.* 2021). In this study, feature importance and partial dependence plots (PDPs) are adopted to address the explainability of the tree-based algorithms. Feature importance plots are notably the most popular explainability technique (Saarela and Jauhiainen 2021). PDPs are model-agnostic plots for describing a predictor's contribution to the fitted model. To generate the feature importance plots, the attribute, 'feature\_importances\_' was fitted to the model regressors. The sklearn inspection module of Scikit-learn was used for creating one-way PDPs to show the interactions between the target responses and the input variables (e.g. elevation, slope, aspect, TPI, TRI etc.).

## 3. Results and analysis

### 3.1. Terrain characteristics

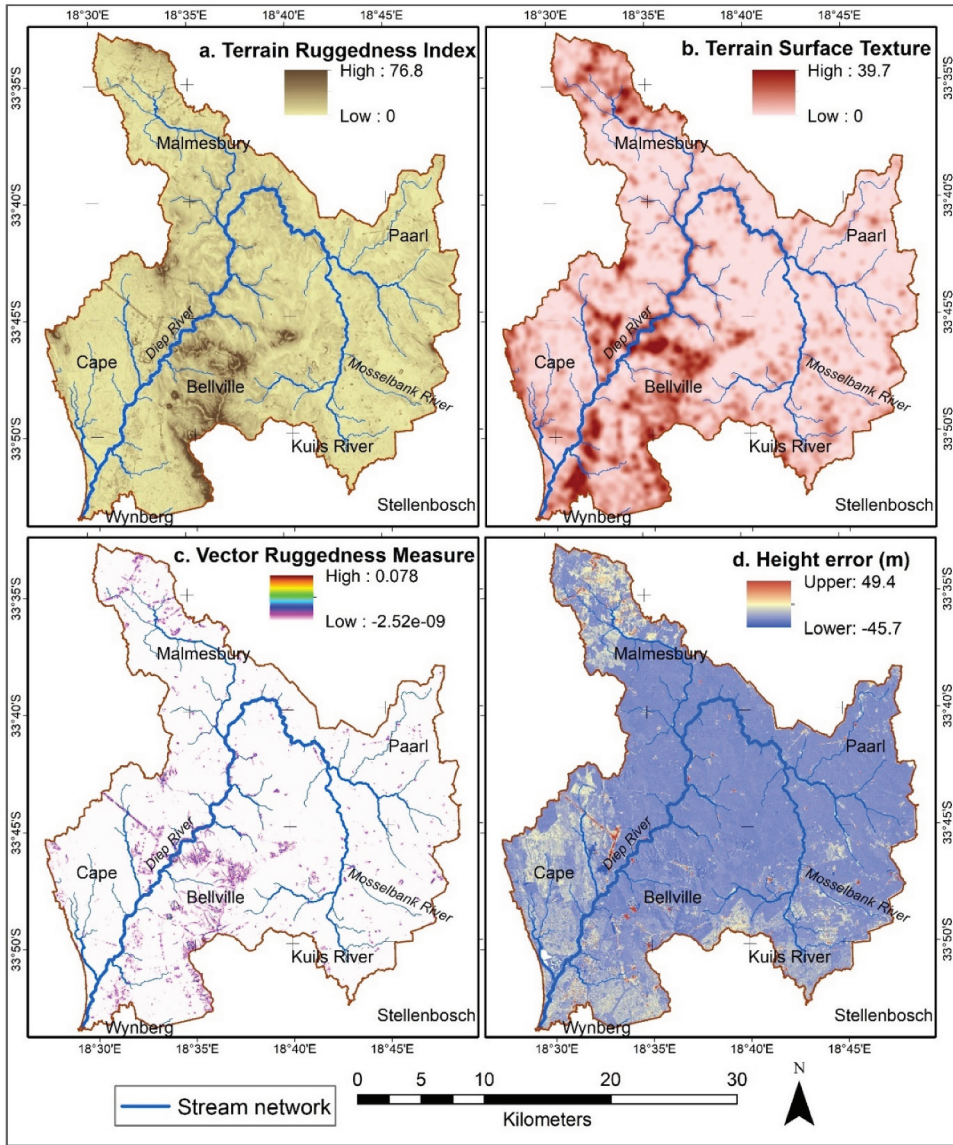
Figures 4 and 5 show maps of the terrain parameters. The streams flowing through the agricultural lands in the Diep River catchment exhibit a dendritic drainage pattern. Along the south-east, the slope ranges from gentle to moderately steep, and steep (Figure 4a). River Mosselbank drains into the Diep River which continues in a downward flow towards the south-west. The steeper slopes tend to exhibit higher TPI than their average surroundings, and vice versa. Table 6 shows descriptive statistics of the predictor variables used for training/testing. The similar terrain conditions at the training/test and implementation sites are evident in Figure 6. Most of the elevation errors at the selected sites are in the range of -5 to +5 m.



**Figure 4.** Maps of the terrain parameters, (a) slope (b) aspect (c) surface roughness, and (d) topographic position index.

### **3.2. Analysis of test error and model computation time**

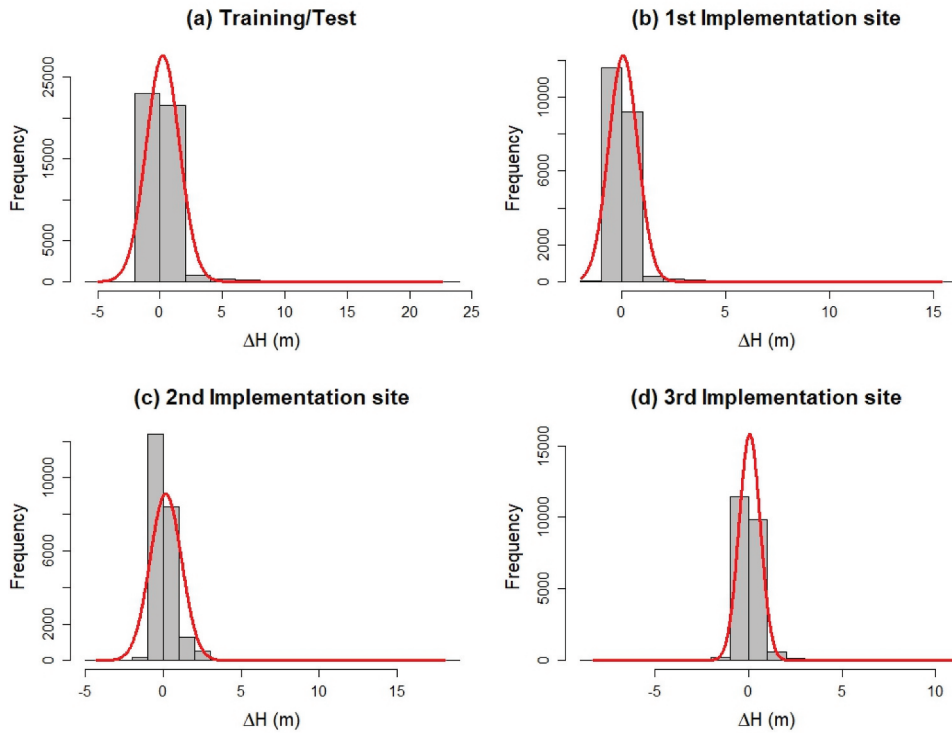
Table 7 presents the MAE, MedAE and RMSE comparisons of the model test errors. CatBoost attained the lowest test MAE (0.310 m) and RMSE (0.704 m). The low test MedAE of several algorithms, e.g. CatBoost (0.153 m), random forest (0.151 m), LightGBM (0.155 m) and XGBoost (0.154 m) suggests their robustness for modelling non-normally distributed data points. The absolute prediction errors are compared in Figure 7.



**Figure 5.** Maps of the terrain parameters – (a) terrain ruggedness index (b) terrain surface texture (c) vector ruggedness measure; and (d) height error map.

**Table 6.** Descriptive statistics of the input variables - training/test dataset.

|                            | Min   | 25th percentile | 50th percentile | 75th percentile | Max    | Mean   | St Dev |
|----------------------------|-------|-----------------|-----------------|-----------------|--------|--------|--------|
| Elevation (m)              | 16.52 | 49.68           | 62.37           | 76.35           | 154.31 | 64.63  | 21.66  |
| Slope (°)                  | 0.01  | 1.91            | 2.88            | 4.25            | 20.56  | 3.30   | 2.02   |
| Aspect (°)                 | 0.00  | 82.48           | 219.36          | 286.34          | 359.99 | 192.71 | 110.43 |
| Surface Roughness          | 0.06  | 2.69            | 4.01            | 5.89            | 26.06  | 4.62   | 2.76   |
| Topographic Position Index | -4.94 | -0.16           | 0.04            | 0.21            | 8.26   | 0.00   | 0.61   |
| Terrain Ruggedness Index   | 0.06  | 2.72            | 3.99            | 5.85            | 28.38  | 4.62   | 2.80   |
| Terrain Surface Texture    | 0.00  | 0.67            | 2.07            | 5.45            | 24.94  | 3.92   | 4.62   |
| Vector Ruggedness Measure  | 0.00  | 0.00            | 0.00            | 0.0002          | 0.0314 | 0.0003 | 0.0009 |



**Figure 6.** Histograms showing the elevation error ( $\Delta H$ ) distribution calculated from the full datasets, at the selected sites.

Possible reasons for the higher prediction error of AdaBoost (shown in Figure 7) are as follows: (i) Sequential learning process: AdaBoost builds an ensemble by adding models sequentially. Each new model focuses on the instances that were misclassified by the previous models. If there are instances that are particularly hard to classify correctly, the algorithm will increasingly focus on these, potentially leading to a biased model that does not perform well on the overall data; (ii) AdaBoost is particularly sensitive to noisy data and outliers. Since its focus is on correcting misclassified points by increasing their weights, it may place too much emphasis on outliers, resulting in poor model performance; (iii) The success of AdaBoost

**Table 7.** Comparison of model test error.

| Tree-based algorithm      | MAE (m)      | RMSE (m)     | MedAE (m)    |
|---------------------------|--------------|--------------|--------------|
| Random Forest             | 0.312        | 0.731        | <b>0.151</b> |
| Bagging Meta-estimator    | 0.327        | 0.757        | 0.160        |
| AdaBoost                  | 0.526        | 1.042        | 0.278        |
| GBM                       | 0.329        | 0.749        | 0.161        |
| XGBoost                   | 0.319        | 0.735        | 0.154        |
| LightGBM                  | 0.314        | 0.715        | 0.155        |
| Histogram-based GBM       | 0.317        | 0.724        | 0.157        |
| CatBoost                  | <b>0.310</b> | <b>0.704</b> | 0.153        |
| NGBoost                   | 0.332        | 0.754        | 0.162        |
| Regularised Greedy Forest | 0.338        | 0.776        | 0.162        |



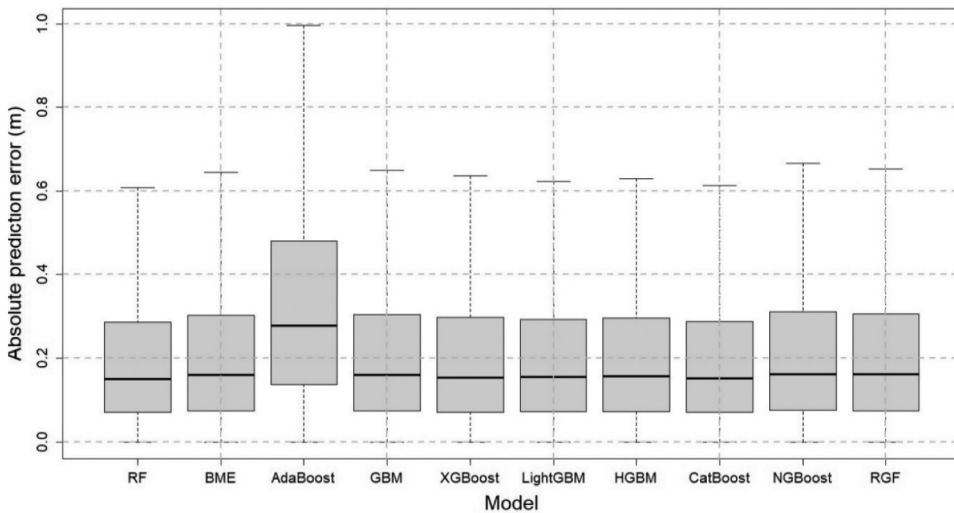


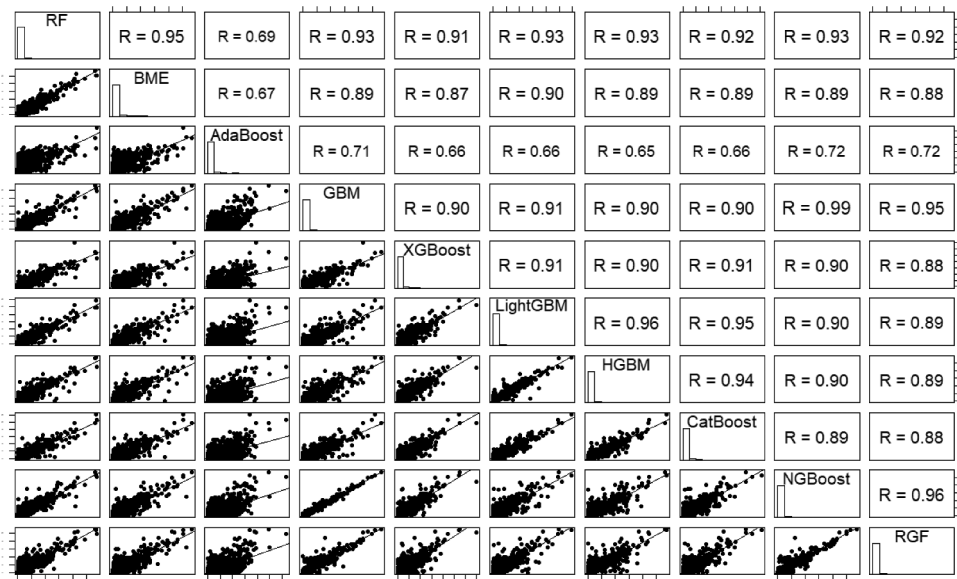
Figure 7. Comparison of model prediction error.

Table 8. Ranking of the tree-based algorithms based on computational time for the training and test set.

| Tree-based algorithm      | Training     |            | Testing      |            |
|---------------------------|--------------|------------|--------------|------------|
|                           | Time (s)     | Rank       | Time (s)     | Rank       |
| Random Forest             | 49.697       | 9th        | 0.504        | 9th        |
| Bagging Meta-estimator    | 4.850        | 5th        | 0.049        | 5th        |
| AdaBoost                  | <b>1.814</b> | <b>3rd</b> | <b>0.026</b> | <b>2nd</b> |
| GBM                       | 13.777       | 8th        | <b>0.021</b> | <b>1st</b> |
| XGBoost                   | 2.937        | 4th        | <b>0.031</b> | <b>3rd</b> |
| LightGBM                  | <b>0.318</b> | <b>1st</b> | 0.050        | 6th        |
| Histogram-based GBM       | <b>0.671</b> | <b>2nd</b> | 0.074        | 7th        |
| CatBoost                  | 7.028        | 6th        | 0.035        | 4th        |
| NGBoost                   | 156.492      | 10th       | 0.636        | 10th       |
| Regularised Greedy Forest | 12.869       | 7th        | 0.179        | 8th        |

largely depends on the choice of weak learners. If this choice is inappropriate for the specific data at hand, the model's performance may suffer in a sequential learning manner.

In most cases, the prediction errors from the ten algorithms have a similar distribution, with a high or very high positive correlation (Figure 8). Computational efficiency is also an important criterion for evaluating the performance of algorithms. The training time of LightGBM (0.318 s) was the fastest among the evaluated algorithms. Previous studies and experiments have corroborated the fast speed and high accuracy of LightGBM (Ke *et al.* 2017). Following LightGBM, the histogram-based GBM had the second fastest training time. Inspired by LightGBM, the histogram-based GBM is reported to give high training speed when applied to large datasets, and it can reduce the training time without losing accuracy (Padhi *et al.* 2021). Although the accuracies of random forest and RGF were comparable with the other GBDTs, their training times were longer. NGBoost had the longest training time of 156.492 s. Notwithstanding, the developers of NGBoost assert that it requires far less expertise to implement (Duan *et al.* 2020). The training speed of the recent GBDTs (e.g. XGBoost, LightGBM and CatBoost) outperformed the GBM regressor



**Figure 8.** The absolute prediction error represented by histograms (diagonal panel), comparative scatter plots (lower panel) and correlations (upper panel).

and random forest. Thus, these recent GBDT implementations could provide highly efficient training speeds, especially for larger datasets.

### 3.3. Accuracy analysis at implementation sites

The implementation sites are small-scale trial sites where the trained algorithms were implemented to assess their feasibility, efficiency, and potential impact. Tables 9 and 10 show the accuracy comparisons of the original and corrected DEMs at the 1st, 2nd and 3rd implementation sites. At the 1st implementation site, there was generally a 6 - 12% reduction in the MAE of the original DEM, and a 15 - 28% reduction in the RMSE of the original DEM after correction. The most significant corrections were achieved by the GBM regressor (RMSE: 0.509 m) followed by XGBoost and NGBoost (RMSE: 0.511 m). With an RMSE of 0.517 m, RGF outperformed some of the popular GBDTs such as LightGBM and CatBoost. AdaBoost was the least accurate (RMSE: 0.596 m), and it caused the MAE of the original DEM to increase by 28% from 0.292 m to 0.372 m.

At the 2nd implementation site, there was a general improvement in DEM accuracy across all the algorithms evident in a 10 - 13% reduction in the MAE of the original DEM, and a 24-29% reduction in the RMSE of the original DEM after correction. However, AdaBoost was the only exception as it increased the MAE by 6% from 0.429 m to 0.454 m. The two bagging algorithms yielded the lowest RMSEs, i.e. random forest (RMSE: 0.735 m) and the Bagging Meta-estimator (RMSE: 0.744 m). While AdaBoost was the least performing algorithm (RMSE: 0.768 m), RGF and XGBoost had similar performance (RMSE: 0.758 m).

At the 3rd implementation site, there were little or no improvements in accuracy. While the MAEs of the original DEM reduced in several instances, the RMSEs unexpectedly



**Table 9.** Accuracy measures of the corrected DEMs at the 1st and 2nd implementation sites. The best performing algorithms are highlighted.

| Tree-based algorithm      | 1st implementation site |          |               |              | 2nd implementation site |          |               |              |
|---------------------------|-------------------------|----------|---------------|--------------|-------------------------|----------|---------------|--------------|
|                           | Original DEM            |          | Corrected DEM |              | Original DEM            |          | Corrected DEM |              |
|                           | MAE (m)                 | RMSE (m) | MAE (m)       | RMSE (m)     | MAE (m)                 | RMSE (m) | MAE (m)       | RMSE (m)     |
| Random Forest             | 0.292                   | 0.703    | 0.260         | 0.517        | 0.429                   | 1.024    | 0.374         | <b>0.735</b> |
| Bagging Meta-estimator    |                         |          | 0.274         | 0.534        |                         |          | 0.385         | 0.744        |
| AdaBoost                  |                         |          | 0.372         | 0.596        |                         |          | 0.454         | 0.768        |
| GBM                       |                         |          | 0.261         | <b>0.509</b> |                         |          | 0.376         | 0.757        |
| XGBoost                   |                         |          | 0.261         | 0.511        |                         |          | 0.377         | 0.758        |
| LightGBM                  |                         |          | 0.258         | 0.519        |                         |          | <b>0.372</b>  | 0.746        |
| Histogram-based GBM       |                         |          | <b>0.258</b>  | 0.519        |                         |          | 0.373         | 0.752        |
| CatBoost                  |                         |          | 0.258         | 0.522        |                         |          | 0.374         | 0.756        |
| NGBoost                   |                         |          | 0.262         | 0.511        |                         |          | 0.377         | 0.757        |
| Regularised Greedy Forest |                         |          | 0.263         | 0.517        |                         |          | 0.377         | 0.758        |

**Table 10.** Accuracy measures of the corrected DEMs at 3rd implementation site. The best performing algorithms are highlighted.

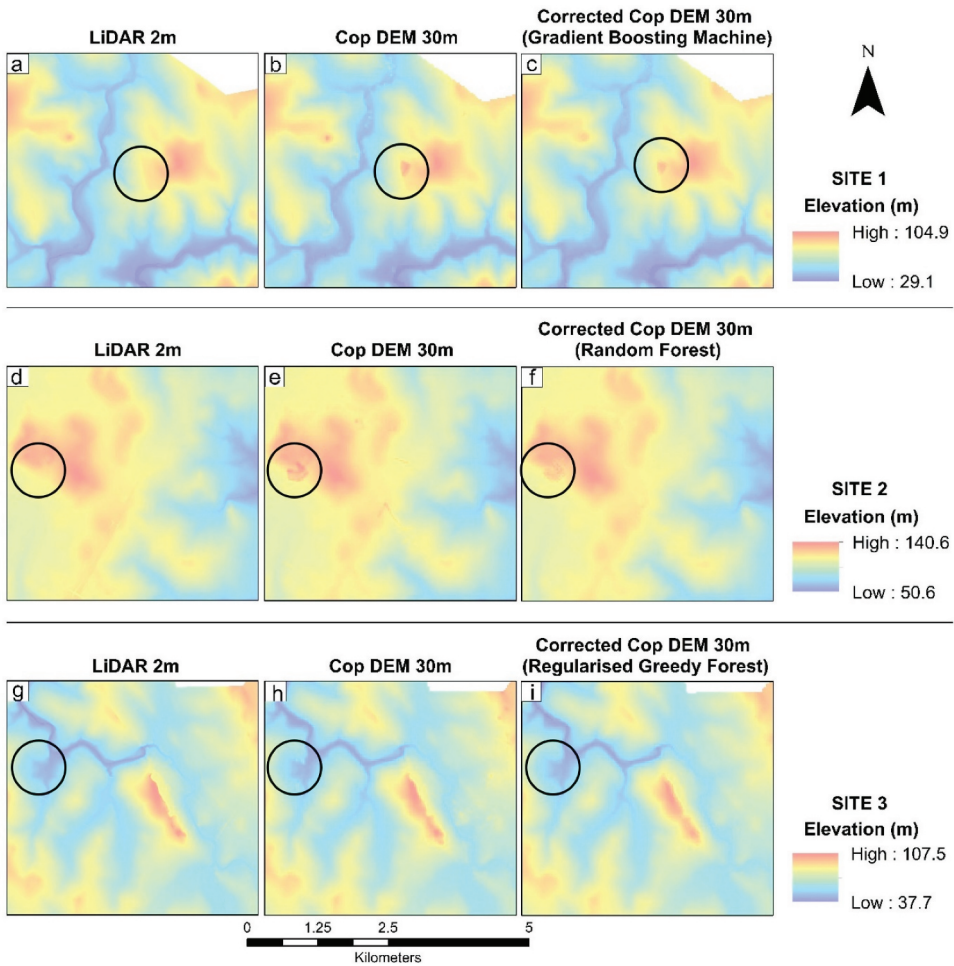
| Tree-based algorithm      | 3rd implementation site |          |               |              |
|---------------------------|-------------------------|----------|---------------|--------------|
|                           | Original DEM            |          | Corrected DEM |              |
|                           | MAE (m)                 | RMSE (m) | MAE (m)       | RMSE (m)     |
| Random Forest             | 0.294                   | 0.567    | 0.296         | 0.636        |
| Bagging Meta-estimator    |                         |          | 0.305         | 0.657        |
| AdaBoost                  |                         |          | 0.388         | 0.680        |
| GBM                       |                         |          | 0.290         | 0.617        |
| XGBoost                   |                         |          | <b>0.289</b>  | 0.614        |
| LightGBM                  |                         |          | 0.290         | 0.613        |
| Histogram-based GBM       |                         |          | 0.291         | 0.618        |
| CatBoost                  |                         |          | 0.294         | 0.627        |
| NGBoost                   |                         |          | <b>0.289</b>  | 0.612        |
| Regularised Greedy Forest |                         |          | 0.290         | <b>0.609</b> |

escalated. Notwithstanding, all the sites were chosen based on their similar terrain characteristics. RGF delivered the lowest RMSE of 0.609 m followed by NGBoost (0.612 m), LightGBM (0.613 m) and XGBoost (0.614 m). Summarily, the achieved accuracies from the models (with the exception of AdaBoost and the Bagging Meta-estimator) are comparable to each other. Nonetheless, the novel tree-structured regularisation, and fully corrective regularised greedy algorithm of RGF (Johnson and Zhang 2014), might have given it the upper hand in this scenario.

Figures 9 and 10 present visualisations of the enhanced DEMs and height error maps respectively, with visible improvements such as the smoothing of rough edges, better stream channel conditioning and refinement of grainy pixels. The elevation error range was also reduced in the enhanced DEMs.

### 3.4. Model explainability

The general notion is that machine learning algorithms are a 'black box', in other words, their internal working mechanisms and how they generate the predictions are most often not comprehensible. This is not absolutely true, especially for the tree-based algorithms. In this study, we have used the feature importance plots and the partial dependence plots



**Figure 9.** Visual comparison of the original and corrected DEMs at the three implementation sites. Areas for comparison are highlighted with the black circles.

(PDP) to explain how the algorithms arrive at certain results. This in turn, makes it easier for non-technical audiences to understand why the algorithms generate certain results and the deriving factors. The feature importance plots for selected tree-based algorithms are presented in [Figure 11](#).

Generally, slope and aspect had moderate or minimal influence in the elevation error prediction by random forest, GBM, XGBoost and RGF. The most influential predictor variables are TPI, TST and VRM especially by random forest, XGBoost CatBoost and RGF. The TPI and VRM have higher sensitivity for landform differentiation. For example, VRM incorporates 3D dispersion of vectors. There was very low importance allocated to elevation, slope, aspect, surface roughness and TRI in the predictions by RGF. The influences of TPI, VRM and TST are well exploited in GBM, whereas other features such as slope, surface roughness and TRI have reduced influence. The NGBoost-derived location and scale parameters are shown in [Figure 12](#). For normally distributed data, the

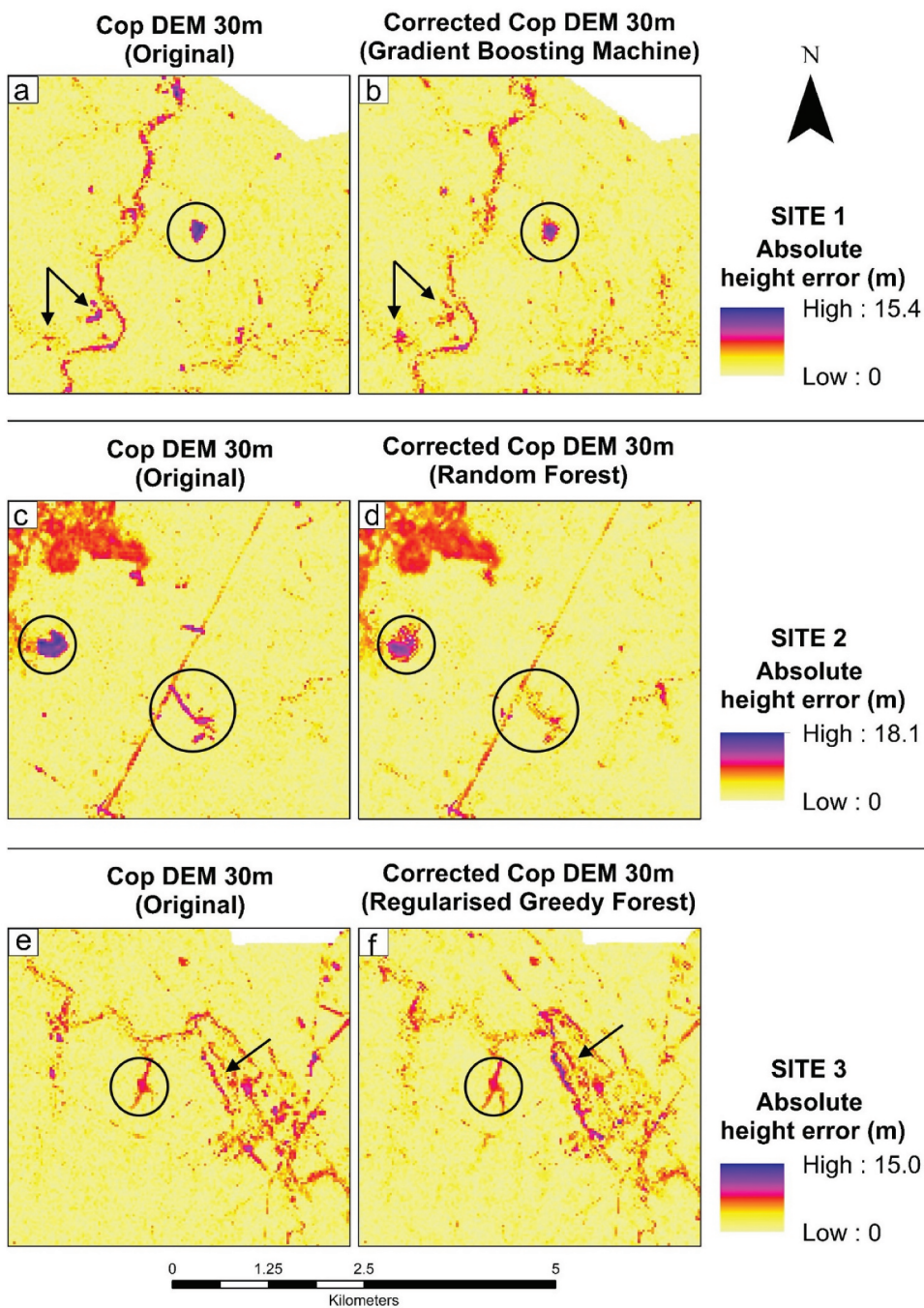
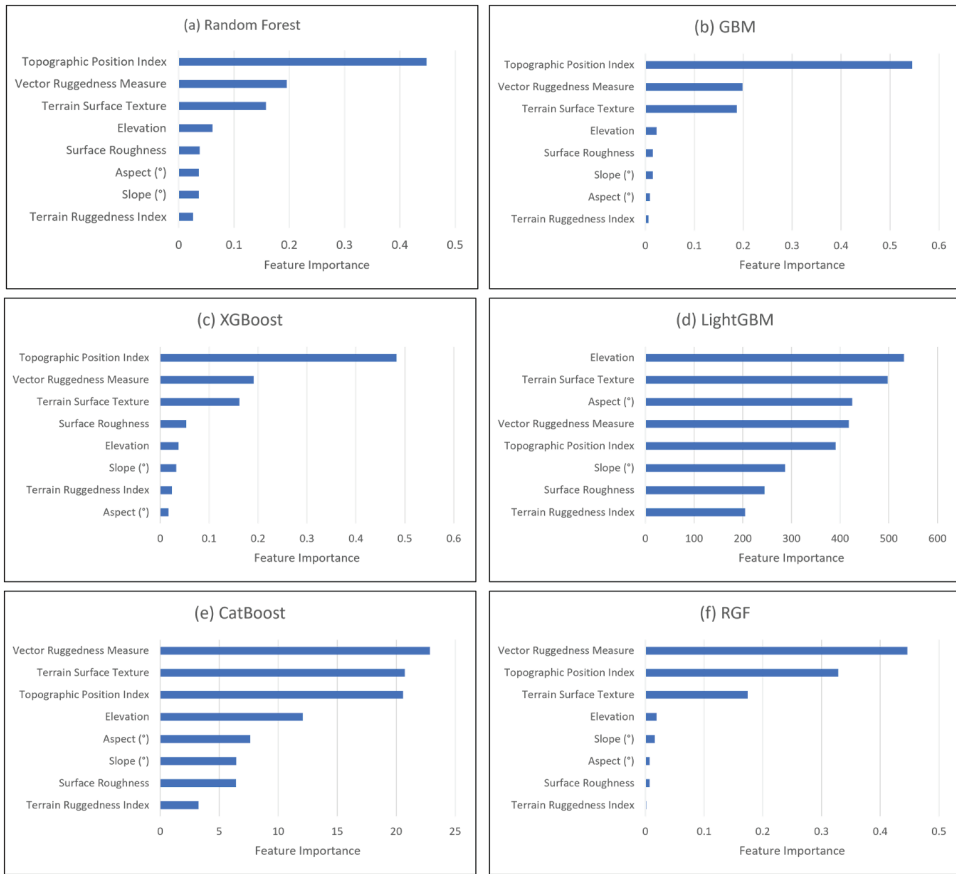


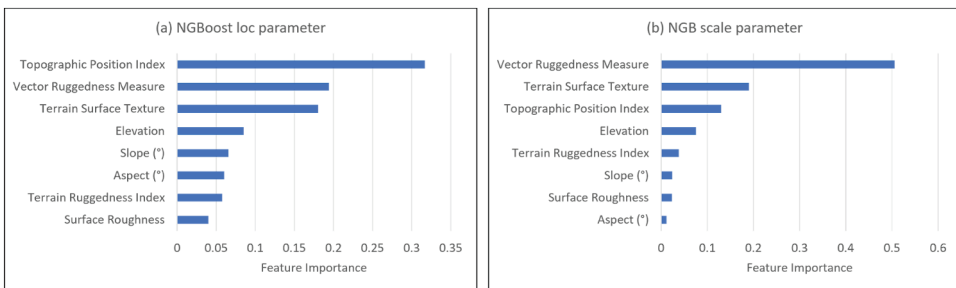
Figure 10. Visual comparison of the height error maps, calculated from the original and corrected DEMs at the three implementation sites. Areas for comparison are highlighted with the black arrows and circles.



**Figure 11.** Feature importance plots shown for some algorithms – (a) random forest (b) GBM (c) XGBoost (d) LightGBM (e) CatBoost (f) RGF.

location and scale parameters correspond to the mean and standard deviation, respectively.

The NGBoost algorithm was designed to extend the gradient boosting hypothesis to probabilistic regression problems. It does this by handling the parameters of the distribution (Gaussian distribution) as targets for a multiparameter boosting



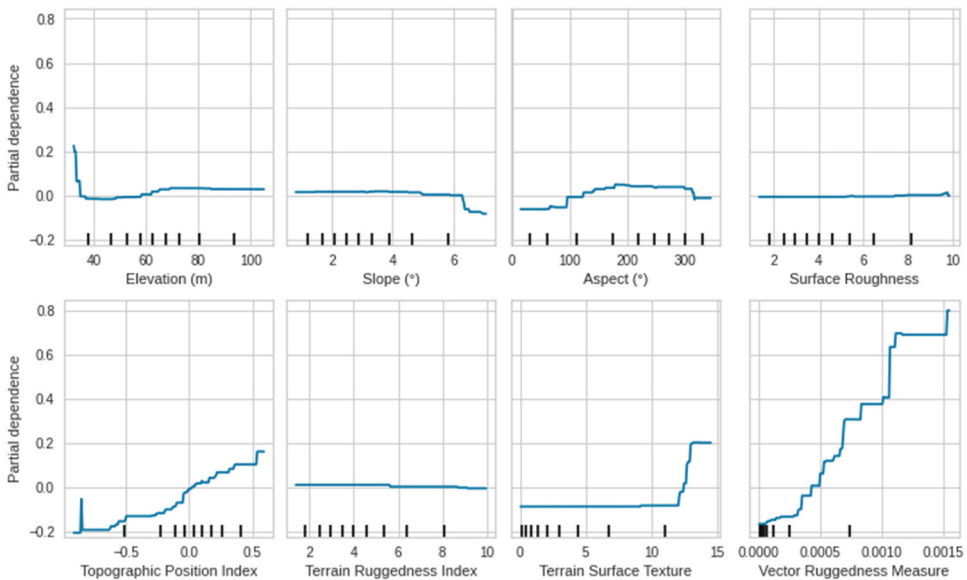
**Figure 12.** Ngboost feature importance plots for distribution parameters – (a) location parameter (b) scale parameter.

algorithm. As shown in Figure 12, the NGBoost model estimates both the location (mean) and scale (standard deviation) parameters of the Gaussian distribution instead of estimating only the conditional mean of the distribution. This approach according to the authors, significantly improved the performance of the model, the flexibility and scalability when compared to other probabilistic predictive algorithms. However, the process of estimating both the location and scale parameters of the Gaussian distribution affects the model's execution time. Table 8 shows that NGBoost took much longer than the other algorithms both in training and inferencing time.

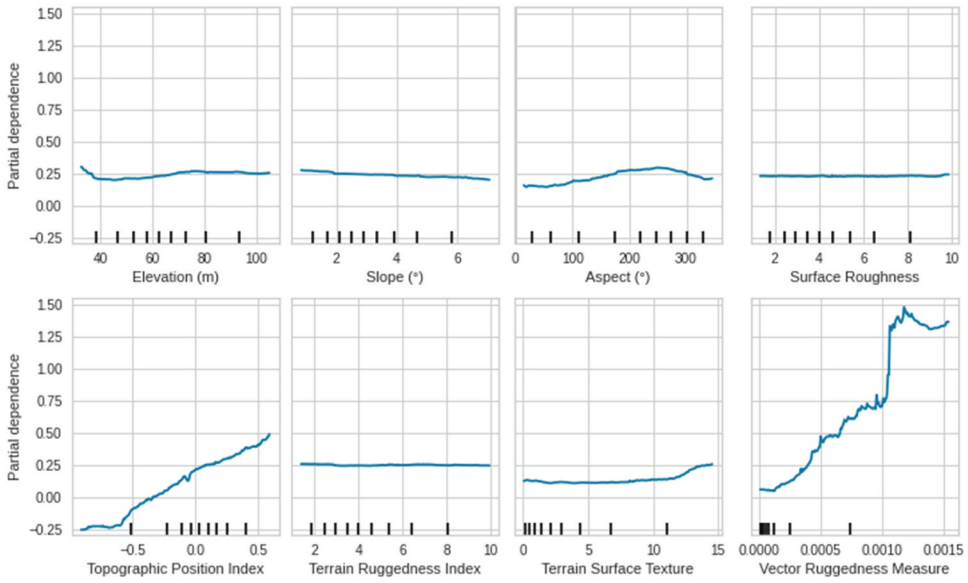
The partial dependence plot (PDP) depicts the functional relationship between a small number of input variables and predictions. It can show how the predictions partially depend on values of the input variables of interest. The PDPs in Figures 13–15 show the interdependence between the input variables (elevation, slope, aspect, surface roughness, TPI, TRI, TST and VRM) and the elevation errors. The trend line shows the changes in elevation errors in response to increasing feature values. In consonance with the feature importance plots, TPI, VRM and TST are found to influence the elevation error very significantly (e.g., in the case of GBM), whereas in random forest and GBM, the elevation errors are less influenced by variations in slope, surface roughness and TRI.

#### 4. Discussion

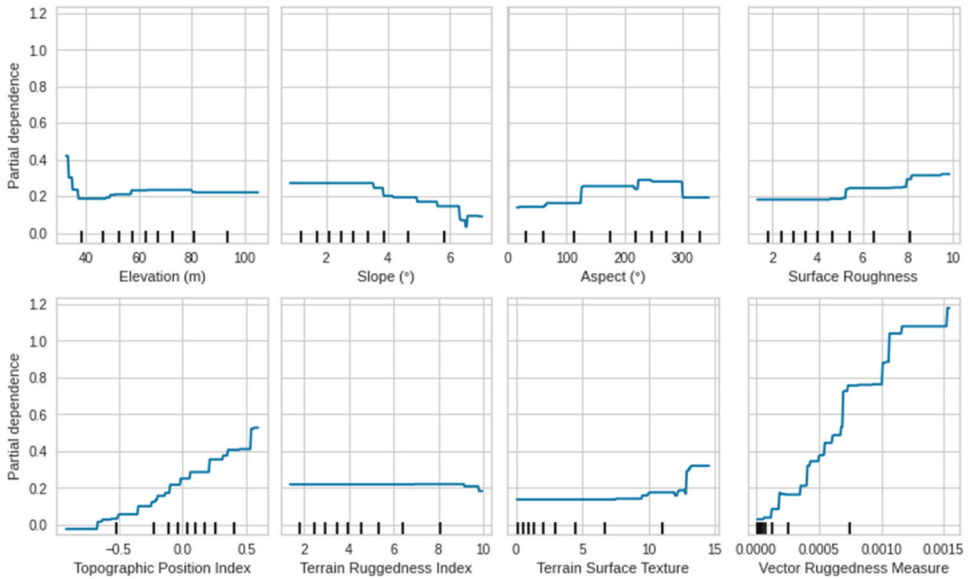
The reduction in DEM error (both MAE and RMSE) at the implementation sites indicates the effectiveness of tree-based algorithms for correcting elevation data. We quantified the



**Figure 13.** Partial dependence plot of the best-performing model (lowest RMSE) at site 1 – GBM. The y-axis shows the partial dependence while the x-axis shows the feature class values.



**Figure 14.** Partial dependence plot of the best-performing model (lowest RMSE) at site 2 – random forest. The y-axis shows the partial dependence while the x-axis shows the feature class values.



**Figure 15.** Partial dependence plot of the best-performing model (lowest RMSE) at site 3 – regularised greedy forest. The y-axis shows the partial dependence while the x-axis shows the feature class values.

improvements in DEM accuracy after model application. For example, at the 1st implementation site, there was a 6–12% reduction in the MAE of the original DEM, and a 15–28% reduction in the RMSE of the original DEM. The analysis in sections 3.2 and 3.3 improves the understanding that the prediction accuracies of different tree-based



algorithms especially on natural landscapes vary with location and topography. It is not correct to generalise that a given tree-based machine learning method does better than the others in DEM enhancement without taking into consideration, the locational characteristics of the site in question. For example, even though CatBoost had the lowest test error, LightGBM emerged with the best performance when applied for DEM error prediction and correction across the whole landscape. Moreover, LightGBM and several of the evaluated algorithms (e.g. Histogram-based GBM, XGBoost and CatBoost) have faster computation times and do not require extensive high-performance computing platforms to be used effectively at scale. This becomes very beneficial to the users who would like to select the most efficient tree-based algorithms for DEM enhancement irrespective of site and location.

Scientists, researchers and industry practitioners are usually interested in the computational efficiency of machine learning algorithms before deploying them on large scale. Table 11 compares the accuracy measures of the corrected DEMs in the whole area. It is shown that LightGBM delivered the best correction when the accuracy measures were averaged for the whole area (MAE: 0.308 m; RMSE: 0.635 m). LightGBM had the fastest training time, followed by the Histogram-based GBM, and AdaBoost. Despite the relatively fast training time of AdaBoost, its poor performance in accuracy is a serious concern that could limit its adoption by users. In terms of the MAE, LightGBM, Histogram-based GBM, and CatBoost emerged as the top three algorithms with the best accuracies in the DEM correction. However, in terms of the RMSE, LightGBM, NGBoost, and Random Forest emerged on top. Overall, LightGBM outperformed all the other algorithms, emerging with the shortest training time (fastest training speed) and it delivered the best accuracies in DEM correction at the implementation sites. The appeal of LightGBM is shown in its very recent adoption by counterpart research groups for DEM correction. For example, Ouyang *et al.* (2023) integrated LightGBM as one of the base models in an ensemble DEM correction framework which they referred to as the 'Stacking Fusion Correction Model.' More recently, Dusseau *et al.* (2023) adopted LightGBM in the development of DiluviumDEM, a new global coastal DEM derived from Copernicus DEM. The developers of DiluviumDEM (i.e. Dusseau *et al.* 2023) highlighted the advantage of LightGBM for global DEM correction involving large datasets. This assertion is corroborated by the performance of LightGBM in the present study where it achieved the fastest training

**Table 11.** Ranking of the tree-based algorithms based on the achieved accuracies for the DEM correction in the whole area. The best performing algorithms are highlighted.

| Tree-based algorithm      | Whole area (1st site + 2nd site + 3rd site) |          |               |            |              |            |
|---------------------------|---------------------------------------------|----------|---------------|------------|--------------|------------|
|                           | Original DEM                                |          | Corrected DEM |            |              |            |
|                           | MAE (m)                                     | RMSE (m) | MAE (m)       | Rank       | RMSE (m)     | Rank       |
| Random Forest             | 0.340                                       | 0.792    | 0.311         | 7th        | 0.638        | 3rd        |
| Bagging Meta-estimator    |                                             |          | 0.322         | 9th        | 0.653        | 9th        |
| AdaBoost                  |                                             |          | 0.405         | 10th       | 0.687        | 10th       |
| GBM                       |                                             |          | 0.311         | 4th        | 0.639        | 5th        |
| XGBoost                   |                                             |          | 0.311         | 6th        | 0.638        | 4th        |
| LightGBM                  |                                             |          | <b>0.308</b>  | <b>1st</b> | <b>0.635</b> | <b>1st</b> |
| Histogram-based GBM       |                                             |          | 0.309         | 2nd        | 0.640        | 7th        |
| CatBoost                  |                                             |          | 0.310         | 3rd        | 0.645        | 8th        |
| NGBoost                   |                                             |          | 0.311         | 5th        | 0.637        | 2nd        |
| Regularised Greedy Forest |                                             |          | 0.312         | 8th        | 0.639        | 6th        |



speed in the whole area of study, thus proving its potential for application to global remote sensing datasets and big geospatial data. Notwithstanding, it is important to note that performance assessments of machine learning algorithms in remote sensing use cases are site-specific and cannot be generalised for other landscapes. For example, XGBoost and CatBoost have surpassed LightGBM in some other assessments. Moreover, the algorithm configurations and terrain dynamics are different in separate assessments.

In section 3.4, we used the model feature importance plots and the feature partial dependence plots to show how each predictor interacts with the algorithms to produce regression values. It is noteworthy that TPI, TST, VRM, and elevation are generally the most influential in the predictability power of all the algorithms tested in this experiment. This becomes a very important and useful analysis for the readers and subsequently users of these algorithms for DEM enhancement because it suggests they could get almost similar results using only TPI, TST, VRM, and elevation in situations where there is less capacity to include all the features used in this experiment. Also, for the purposes of speed and scale, other researchers can reduce the number of predictors to the top 4 or 5 based on this explainability analysis.

## 5. Conclusion

Machine learning has presented an effective method to model complex terrain parameters. Tree-based ensembles are very powerful for reducing the uncertainty in digital elevation datasets and enhancing DEM quality. This low-cost approach can be adopted by national mapping organisations with budgetary constraints to enhance wide-area DEMs for producing more accurate topographic maps and cartographic products. Topographic position index, terrain surface texture and vector ruggedness measure were revealed as very influential terrain parameters for the elevation error prediction in the studied agricultural landscapes, whereas slope, aspect, surface roughness, and TRI had less influence on the predictions. This specific finding is a valuable contribution to understanding the interdependencies and influence of terrain parameters in modelling the topography of landscapes. By addressing the explainability of the compared machine learning algorithms, this study demystifies the complexities behind the effective deployment and analysis of terrain features for prediction and modelling of DEM error. Understanding the impact of these variables can help practitioners focus on the most relevant data, improve data collection practices, and refine model inputs for better accuracy. This knowledge is also important for tailoring models to specific applications where certain types of prediction sensitivity are required.

We have also tested the probabilistic regression algorithm NGBoost for predicting point estimates of the elevation error. Thus, it has been showcased that natural gradients are effective for use in DEM enhancement and remote sensing tasks. However, the learning task parameters of NGBoost have not been fully developed to the level of more advanced GBDTs such as XGBoost, LightGBM, and CatBoost. Summarily, all the tested algorithms (except for AdaBoost) provide satisfactory results in terms of the achievable accuracy. This comparative analysis serves as an invaluable source of knowledge on the performances of tree-based ensembles for handling remote sensing tasks. Overall, this research presents a comprehensive approach to enhancing DEM accuracy using machine learning. More importantly, we have advanced the understanding of

model explainability in the context of terrain analysis and remote sensing. This challenges the common perception of machine learning algorithms as incomprehensible ‘black boxes’. By using tools like feature importance and PDPs, the study demonstrates that it is possible to gain insights into how certain machine learning models, especially tree-based models, make their predictions.

Tree-based ensembles are advantageous in terms of time complexity, and this is a significant advantage when deploying models on a wider scale. The computational burden of many deep learning implementations is a confounding factor and a serious limitation for researchers and industry practitioners, especially when computational resources are limited or sparse. The choice of algorithm depends on available computing resources and user requirements. Both Bagging and Boosting ensembles provide competitive accuracy and finally, according to the No-Free-Lunch (NFL) theorem, there is no universal algorithm that can solve all types of problems.

## Acknowledgement

We are grateful to the Chief Directorate: National Geospatial Information (CD: NGI), South Africa for providing us with information on the vertical datum for South Africa. LIDAR data for the City of Cape Town was provided by the Information and Knowledge Management Department, City of Cape Town. We appreciate the support of Professor Jennifer Whittal. Lastly, we thank the journal editors and reviewers for their valuable feedback which improved the quality of the article.

## Code availability statement

Code written in support of this publication is publicly available at <https://github.com/mrjohnokolie/dem-enhancement>

## Disclosure statement

No potential conflict of interest was reported by the author(s).

## Funding

The research reported here was funded by the (i) Commonwealth Scholarship Commission and the Foreign, Commonwealth and Development Office in the UK (CSC ID: NGCN-2021-239) (ii) University of Cape Town. We are grateful for their support. All views expressed here are those of the author(s) not the funding bodies.

## Data availability statement

On reasonable request, the corresponding author will provide data that support the findings of this study.

## Author contributions

- Chukwuma Okolie: conceptualisation, writing – original draft, review and editing, methodology, data curation, software, investigation, visualisation, formal analysis.

- Adedayo Adeleke: conceptualisation, writing – review and editing, supervision.
- Jon Mills: conceptualisation, writing – review and editing, resources, supervision.
- Julian Smit: conceptualisation, writing – review and editing, resources, supervision.
- Ikechukwu Maduako: writing – review and editing, resources.
- Hossein Bagheri: writing – review and editing, resources.
- Tom Komar: writing – review and editing, resources.
- Shidong Wang: writing – review and editing, resources.

## References

- Albert, G. and Ammar, S., 2021. Application of random forest classification and remotely sensed data in geological mapping on the Jebel Meloussi area (Tunisia). *Arabian Journal of Geosciences*, 14 (21), 2240. doi:10.1007/s12517-021-08509-x
- Bagheri, H., Schmitt, M., and Xiang Zhu, X., 2018. Fusion of TanDEM-X and cartosat-1 elevation data supported by neural network-predicted weight maps. *Isprs Journal of Photogrammetry and Remote Sensing*, 144 (August), 285–297. doi:10.1016/j.isprs.2018.07.007
- Barrow, D.K. and Crone, S.F., 2016. A comparison of AdaBoost algorithms for time series forecast combination. *International Journal of Forecasting*, 32 (4), 1103–1119. doi:10.1016/J.IJFORECAST.2016.01.006
- Bauer, E. and Kohavi, R., 1999. Empirical comparison of voting classification algorithms: bagging, boosting, and variants. *Machine Learning*, 36 (1), 105–139. doi:10.1023/A:1007515423169
- Biau, G. and Scornet, E., 2016. A random forest guided tour. *Test*, 25 (2), 197–227. doi:10.1007/s11749-016-0481-7
- Borisov, V., et al. 2022. Deep neural networks and Tabular Data: a survey, in *IEEE Transactions on Neural Networks and Learning Systems*, doi: 10.1109/TNNLS.2022.3229161.
- Breiman, L., 1996. Bagging Predictors. *Machine Learning*, 24 (2), 123–140. doi:10.1007/BF00058655
- Breiman, L., 2001. Random forests. *Machine Learning*, 45 (1), 5–32. doi:10.1023/A:1010933404324
- Caruana, R. and Niculescu-Mizil, A., 2006. An empirical comparison of supervised learning algorithms. *ACM International Conference Proceeding Series* 148: 161–168. 10.1145/1143844.1143865.
- Chai, T. and Draxler, R.R., 2014. Root mean square error (RMSE) or mean absolute error (MAE)? – Arguments against avoiding RMSE in the literature. *Geoscientific Model Development*, 7 (3), 1247–1250. doi:10.5194/GMD-7-1247-2014
- Chen, T.Q. and Guestrin, C. 2016. XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd Acm Sigkdd International Conference on Knowledge Discovery and Data Mining*, San Francisco, CA, USA, 13–17 August 2016. 785–794.
- Chen, C., Yang, S., and Li, Y., 2020. Accuracy assessment and correction of SRTM DEM using ICESat/GLAS data under data coregistration. *Remote Sensing*, 12 (20), 3435. doi:10.3390/RS12203435
- Christie, D. and Neill, S.P., 2022. Measuring and observing the Ocean Renewable Energy Resource. *Comprehensive Renewable Energy*, 149–175. doi:10.1016/B978-0-12-819727-1.00083-2.
- Climent, F., Momparler, A., and Carmona, P., 2019. Anticipating bank distress in the Eurozone: An Extreme Gradient Boosting approach. *Journal of Business Research*, 101, 885–896. doi:10.1016/j.jbusres.2018.11.015
- Das, T., 2022. *Google Colab: everything you need to know*. Geekflare. <https://geekflare.com/google-colab/>.
- Department of Environment and Science, Queensland (2020) *Terrain Roughness, WetlandInfo* website. Available from: <https://wetlandinfo.des.qld.gov.au/wetlands/ecology/aquatic-ecosystems-natural/estuarine-marine/itst/terrain-roughness/> [Accessed 4 October 2022].
- Drakenstein River Environmental Management Plan (2008). Available from: <http://www.drakenstein.gov.za/docs/Documents/River%20EMP%20Part%201%20Report%20Structure.pdf> [accessed 11 October 2022]
- Duan, T., et al. (2020). NGBoost: Natural Gradient Boosting for Probabilistic Prediction *Proceedings of the 37 th International Conference on Machine Learning*, Vienna, Austria, PMLR 108.

- Dusseau, D., Zobel, Z., and Schwalm, C.R., 2023. DiluviumDEM: Enhanced Accuracy in Global Coastal Digital Elevation Models. *Remote Sensing of Environment*, 298 (September), 113812. doi:10.1016/j.rse.2023.113812
- DWS (Department of Water and Sanitation) (2020). South Africa: part 1 of 3. Government Gazette, p 1–292
- Elith, J., Leathwick, J.R., and Hastie, T., 2008. A working guide to boosted regression trees. *The Journal of Animal Ecology*, 77 (4), 802–813. doi:10.1111/j.1365-2656.2008.01390.x
- ESA (2020a). Copernicus Digital Elevation Model Product Handbook. Tech. Rep. GEO.2018-1988-2, AIRBUS: [https://spacedata.copernicus.eu/documents/20126/0/GEO1988-CopernicusDEM-RP-001\\_ValidationReport\\_I3.0.pdf](https://spacedata.copernicus.eu/documents/20126/0/GEO1988-CopernicusDEM-RP-001_ValidationReport_I3.0.pdf)
- ESA (2020b). Copernicus Digital Elevation Model Validation Report. Tech. Rep. GEO.2018-1988-2, AIRBUS: [https://spacedata.copernicus.eu/documents/20126/0/GEO1988-CopernicusDEM-SPE-002\\_ProductHandbook\\_I1.00.pdf](https://spacedata.copernicus.eu/documents/20126/0/GEO1988-CopernicusDEM-SPE-002_ProductHandbook_I1.00.pdf)
- Geurts, P., Ernst, D., and Wehenkel, L., 2006. Extremely randomized trees. *Machine Learning*, 63 (1), 3–42. doi:10.1007/s10994-006-6226-1
- Girohi, P. and Bhardwaj, A., 2022. A neural network-based fusion approach for improvement of SAR Interferometry-based Digital Elevation Models in plain and hilly regions of India. *AI*, 3 (4), 820–843. <http://doi.org/10.3390/AI3040050>.
- Goodness, J. and Anderson, P.M.L., 2013. Local assessment of Cape Town: navigating the management complexities of urbanization, biodiversity, and Ecosystem Services in the Cape Floristic Region. In: *Urbanization, Biodiversity and Ecosystem Services: challenges and opportunities*. Springer: Dordrecht. doi:10.1007/978-94-007-7088-1\_24
- Google. 2023. “Google Colab”. <https://research.google.com/colaboratory/faq.html>.
- Google for Developers (2023). Machine learning glossary. Available from: <https://developers.google.com/machine-learning/glossary/df> [accessed 12 August 2023].
- Gqomfa, B., Maphanga, T., and Shale, K., 2022. The impact of informal settlement on water quality of Diep River in Dunoon. *Sustainable Water Resources Management*, 8 (1), 27. doi:10.1007/s40899-022-00629-w
- Han, J., et al., 2019. Using decision tree to predict response rates of consumer satisfaction, attitude, and loyalty surveys. *Sustainability*, 11 (8), 2306. <http://doi.org/10.3390/su11082306>.
- Hancock, J.T. and Khoshgoftaar, T.M., 2020. CatBoost for big data: an interdisciplinary review. *Journal of Big Data*, 7 (1), 1–45. <http://doi.org/10.1186/s40537-020-00369-8>
- Hastie, T., Tibshirani, R., and Friedman, J., 2009. *The elements of statistical learning*. Second Edition [https://hastie.su.domains/ElemStatLearn/printings/ESLII\\_print12\\_toc.pdf](https://hastie.su.domains/ElemStatLearn/printings/ESLII_print12_toc.pdf).
- Hawker, L., et al., 2022. A 30 m global map of elevation with forests and buildings removed. *Environmental Research Letters*, 17 (2), 024016. <http://doi.org/10.1088/1748-9326/AC4D4F>.
- Hu, M. and Ji, S., 2022. Accuracy evaluation and improvement of common DEM in Hubei region based on ICESat/GLAS data. *Earth Science Informatics*, 15 (1), 221–231. <http://doi.org/10.1007/s12145-021-00721-3>
- Johnson, R. and Zhang, T., 2014. Learning nonlinear functions using regularized greedy forest. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 36 (5), 942–954. <http://doi.org/10.1109/TPAMI.2013.159>
- Joseph, M. (2020). The gradient boosters II: regularized greedy forest. Available from: <https://deep-and-shallow.com/2020/02/09/the-gradient-boosters-ii-regularized-greedy-forest/> [accessed 31 July 2023]
- Kadra, A., Lindauer, M., Hutter, F., Grabocka, J. (2021). Well-tuned Simple Nets Excel on Tabular Datasets. *Advances in Neural Information Processing Systems* 34 (NeurIPS 2021), Virtual, December 7–10, 2021. <https://proceedings.neurips.cc/paper/2021/file/c902b497eb972281fb5b4e206db38ee6-Paper.pdf>
- Karlson, M., et al., 2015. Mapping tree canopy cover and aboveground biomass in Sudano-Sahelian Woodlands using Landsat 8 and random forest. *Remote Sensing*, 7 (8), 10017–10041. <http://doi.org/10.3390/rs70810017>.

- Kasi, V., et al., 2020. A novel method to improve vertical accuracy of CARTOSAT DEM using machine learning models. *Earth Science Informatics*, 13 (4), 1139–1150. <http://doi.org/10.1007/s12145-020-00494-1>
- Kavzoglu, T. and Teke, A., 2022. Predictive performances of ensemble machine learning algorithms in landslide susceptibility mapping using random forest, extreme gradient boosting (XGBoost) and natural gradient boosting (NGBoost). *Arabian Journal for Science & Engineering*, 47 (6), 7367–7385. <http://doi.org/10.1007/s13369-022-06560-8>
- Ke, G., et al. 2017. LightGBM: a highly efficient gradient boosting decision tree. 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA. <https://proceedings.neurips.cc/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf>.
- Kearns, M. and Valiant, L., 1994. Cryptographic limitations on learning boolean formulae and finite automata. *Journal of the ACM (JACM)*, 41 (1), 67–95. <http://doi.org/10.1145/174644.174647>
- Kim, D.E., et al., 2020. Simple-yet-effective SRTM DEM Improvement Scheme for Dense Urban Cities Using ANN and remote sensing data: application to flood modeling. *Water*, 12 (3), 816. <http://doi.org/10.3390/W12030816>
- Kim, D.E., et al., 2020a. Simple-yet-effective SRTM DEM improvement scheme for Dense Urban Cities Using ANN and remote sensing data: application to flood modeling. *Water*, 12 (3), 816.
- Kim, D.E., et al., 2021. Satellite DEM Improvement Using Multispectral Imagery and an Artificial Neural Network. *Water*, 13 (11), 1551. doi:10.3390/W13111551
- Kim, D.-E., Gourbesville, P., and Liong, S.-Y., 2019. Overcoming data scarcity in flood hazard assessment using remote sensing and artificial neural network. *Smart Water*, 4 (1), 1–15. <http://doi.org/10.1186/S40713-018-0014-5>
- Kotsiantis, S.B., 2013. Decision trees: a recent overview. *Artificial Intelligence Review*, 39 (4), 261–283. <http://doi.org/10.1007/s10462-011-9272-4>
- Kulp, S. and Strauss, B., 2018. CoastalDEM: A Global Coastal Digital Elevation Model Improved from SRTM Using a Neural Network. *Remote Sensing of Environment*, 206, 231–239. <http://doi.org/10.1016/J.RSE.2017.12.026>
- Linardatos, P., Papastefanopoulos, V., and Kotsiantis, S., 2021. Explainable AI: a review of machine learning interpretability methods. *Entropy*, 23 (1), 18. <http://doi.org/10.3390/e23010018>
- Liu, Y., et al., 2021. Bare-Earth DEM Generation in Urban Areas for Flood Inundation Simulation Using Global Digital Elevation Models. *Water Resources Research*, 57 (4), e2020WR028516. <http://doi.org/10.1029/2020WR028516>.
- Ma, Y., et al., 2020. An innovative approach for improving the accuracy of digital elevation models for cultivated land. *Remote Sensing*, 12 (20), 3401. <http://doi.org/10.3390/rs12203401>
- Mariani, S. and Sipper, M., 2022. High per parameter: a large-scale study of hyperparameter tuning for machine learning algorithms. *Algorithms*, 15 (9), 315. <http://doi.org/10.3390/A15090315>
- Meadows, M. and Wilson, M., 2021. A comparison of machine learning approaches to improve free topography data for flood modelling. *Remote Sensing 2021*, 13 (2), 275. <http://doi.org/10.3390/RS13020275>
- Miao, X., et al., 2012. Applying tree-based ensemble algorithms to the classification of ecological zones using multi-temporal multi-source remote-sensing data. *International Journal of Remote Sensing*, 33 (6), 1823–1849. <http://doi.org/10.1080/01431161.2011.602651>
- Microsoft Corporation (2022). LightGBM documentation. Available from: <https://lightgbm.readthedocs.io/en/v3.3.3/> [accessed: 11 November 2022]
- Nie, X., et al., 2019. Effects of soil properties, topography and landform on the understory biomass of a pine forest in a subtropical hilly region. *Catena*, 176, 104–111. <http://doi.org/10.1016/j.catena.2019.01.007>
- Nyuytiybiy, K., 2020. Parameters and hyperparameters in machine learning and deep learning. *Towards Data Science*. <https://towardsdatascience.com/parameters-and-hyperparameters-aa609601a9ac>.
- Okolie, C.J. and Smit, J.L., 2022. A systematic review and meta-analysis of Digital Elevation Model (DEM) fusion: pre-processing, methods and applications. *ISPRS Journal of Photogrammetry and Remote Sensing*, 188, 1–29. <http://doi.org/10.1016/j.isprsjprs.2022.03.016>

- Olajubu, V., et al., 2021. Urban correction of global DEMs using building density for Nairobi, Kenya. *Earth Science Informatics*, 14 (3), 1383–1398. <http://doi.org/10.1007/s12145-021-00647-w>
- Olusina, J.O. and Okolie, C.J., 2018. Visualisation of Uncertainty in 30m Resolution Global Digital Elevation Models: SRTM v3.0 and ASTER V2. *Nigerian Journal of Technological Development*, 15 (3), 77. <http://doi.org/10.4314/njtd.v15i3.2>
- Orimoloye, I., et al., 2019. Spatial assessment of drought severity in Cape Town area, South Africa. *Heliyon*, 5 (7), e02148. <http://doi.org/10.1016/j.heliyon.2019.e02148>
- Ouyang, Z., et al., 2023. SRTM DEM correction using ensemble machine learning algorithm. *Remote Sensing*, 15 (16), 3946. <http://doi.org/10.3390/rs15163946>
- Padhi, D.K., et al., 2021. A fusion framework for forecasting financial market direction using enhanced ensemble models and technical indicators. *Mathematics*, 9 (21), 2646. <http://doi.org/10.3390/MATH9212646>
- Pakoksung, K. and Takagi, M., 2016. Digital elevation models on accuracy validation and bias correction in vertical. *Modeling Earth Systems and Environment*, 2 (1), 11. <http://doi.org/10.1007/s40808-015-0069-3>
- Pham, X.T.T. and Ho, T.H., 2021. Using boosting algorithms to predict bank failure: an untold story. *International Review of Economics & Finance*, 76 (November 2021), 40–54. <http://doi.org/10.1016/j.iref.2021.05.005>
- Preety, K., et al., 2022. Accuracy assessment, comparative performance, and Enhancement of Public Domain Digital Elevation Models (ASTER 30 m, SRTM 30 m, CARTOSAT 30 m, SRTM 90 m, MERIT 90 m, and TanDEM-X 90 m) using DGPS. *Remote Sensing*, 14 (6), 1334. <http://doi.org/10.3390/rs14061334>
- Prokhorenkova, L., et al. 2018. CatBoost: unbiased boosting with categorical features. *Advances in Neural Information Processing Systems*, 31. <https://github.com/catboost/catboost>
- Pushp, P., 2023. #7 beyond accuracy: understanding and measuring efficiency in machine learning models. Available from: <https://www.linkedin.com/pulse/7-beyond-accuracy-understanding-measuring-efficiency-machine-pushp/> [Accessed 3 January 2023].
- Roe, B.P., et al., 2005. Boosted decision trees as an alternative to artificial neural networks for particle identification. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 543 (2–3), 577–584. <http://doi.org/10.1016/J.NIMA.2004.12.018>.
- Rouse, M., 2019. Computational Complexity. Techopedia article. Available from: <https://www.techopedia.com/definition/18466/computational-complexity> [Accessed 3 January 2024].
- Saarela, M. and Jauhiainen, S., 2021. Comparison of feature importance measures as explanations for classification models. *SN Applied Sciences*, 3 (2), 1–12. doi:10.1007/s42452-021-04148-9
- Safaei, N., et al., 2022. E-CatBoost: an efficient machine learning framework for predicting ICU mortality using the eICU collaborative research database. *Public Library of Science ONE*, 17 (5), e0262895. <http://doi.org/10.1371/journal.pone.0262895>
- Salah, M., 2021. SRTM DEM correction over dense urban areas using inverse probability weighted interpolation and sentinel-2 multispectral imagery. *Arabian Journal of Geosciences*, 14 (9), 1–16. <http://doi.org/10.1007/s12517-021-07148-6>
- Schapiro, R.E. (2003). The boosting approach to machine learning an overview. Nonlinear estimation and classification. [www.research.att.com/](http://www.research.att.com/).
- Schindler, K., et al., 2011. Improving wide-area DEMs through data fusion – chances and limits. *Photogrammetric Week*, 11, 159–170.
- Scikit-learn 2023. Scikit-learn - machine learning in python. Available from: <https://scikit-learn.org/stable/> (Accessed 31 July 2023)
- Shwartz-Ziv, R. and Armon, A., 2022. Tabular data: deep learning is not all you need. *Information Fusion*, 81, 84–90. <http://doi.org/10.1016/j.inffus.2021.11.011>
- Singh, A. (2023). A comprehensive Guide to ensemble learning (with python codes) Available from: <https://www.analyticsvidhya.com/blog/2018/06/comprehensive-guide-for-ensemble-models/> [Accessed 22 January, 2024]
- Talebi, H., et al., 2022. A truly spatial random forests algorithm for geoscience data analysis and modelling. *Mathematical Geosciences*, 54 (1), 1–22. <http://doi.org/10.1007/s11004-021-09946-w>



- USNA (2022). Topographic Ruggedness Index (TRI). Available from: [https://www.usna.edu/Users/oceano/pguth/md\\_help/html/topo\\_rugged\\_index.htm](https://www.usna.edu/Users/oceano/pguth/md_help/html/topo_rugged_index.htm) [Accessed 3 October 2022]
- Vinod, P.G., 2017. Development of topographic position index based on Jenness algorithm for precision agriculture at Kerala, India. *Spatial Information Research*, 25 (3), 381–388. doi:10.1007/s41324-017-0104-8
- Wang, Y. and Wang, T., 2020. Application of Improved LightGBM Model in Blood Glucose Prediction. *Applied Sciences*, 10 (9), 3227. <http://doi.org/10.3390/app10093227>
- Welty, J.L. and Jeffries, M.I. 2018 Western United States ruggedness raw values: U.S. Geological survey data release, <https://www.sciencebase.gov/catalog/item/5ab296d2e4b081f61ab4601a>
- Wendi, D., et al., 2016. An innovative approach to improve SRTM DEM using multispectral imagery and artificial neural network. *Journal of Advances in Modeling Earth Systems*, 8 (2), 691–702. <http://doi.org/10.1002/2015MS000536>.
- Yang, Q.-Y., et al., 2014. Prediction of soil organic matter in peak-cluster depression region using kriging and terrain indices. *Soil and Tillage Research*, 144, 126–132. doi:10.1016/j.still.2014.07.011
- Yanjun, S., et al., 2015. SRTM DEM correction in vegetated mountain areas through the Integration of Spaceborne LiDAR, airborne LiDAR, and optical imagery. *Remote Sensing*, 7 (9), 11202–11225. <http://doi.org/10.3390/rs70911202>
- Zhang, Y. and Haghani, A., 2015. A gradient boosting method to improve travel time prediction. *Transportation Research Part C: Emerging Technologies*, 58 (September), 308–324. <http://doi.org/10.1016/J.TRC.2015.02.019>
- Zhang, Y., Liu, J., and Shen, W.A., 2022. Review of ensemble learning algorithms used in remote sensing applications. *Applied Sciences*, 12 (17), 8654. <http://doi.org/10.3390/app12178654>