TRANSNAV
http://www.transnav.eu
the International Journal on Marine Navigation and Safety of Sea Transportation
Volume 19
Number 4
December 2025

DOI: 10.12716/1001.19.04.34

Use of Open Source for Bottom Model Interpolation – Single Beam Use Case in SONARMUS Software

P. Biernacik[1], W. Kazimierski[1] & M. Włodarczyk-Sielicka[2]
[1] *Maritime University of Szczecin, Szczecin, Poland*
[2] *InnoPM Ltd, Szczecin, Poland*

ABSTRACT: Single Beam Echosounders (SBES) continue to serve as a practical solution for hydrographic surveys. Despite their limitations in spatial coverage, SBES can provide reliable depth measurements when supported by solid post-processing techniques. One of the critical steps in generating usable bathymetric products from SBES data is interpolation – a method used to estimate the continuous bottom model from discrete depth soundings. This paper presents research on the use of open-source tools for bottom model interpolation in SBES datasets within customly developed SONARMUS software environment. Several commonly used interpolation techniques in geospatial and data science applications are evaluated, including Inverse Distance Weighting (IDW), Kriging, Natural Neighbor, and Radial Basis Functions (RBF). Emphasis is placed on the reproducibility and flexibility offered by Python-based open-source libraries such as SciPy, PyKrige, and GDAL. Real SBES datasets from various surveys are used for testing and validation. The study evaluates the accuracy of the interpolation by comparing the predicted depths with control points and assessing the root mean square error (RMSE), mean absolute error (MAE), and distribution analysis. The performance and suitability of each method is examined in terms of computational efficiency, sensitivity to data density and ease of integration into existing workflows. The results demonstrate that open-source tools, when appropriately configured, can effectively support high-quality bottom model generation from SBES data. The integration of these tools into SONARMUS enhances the software's capability to deliver reliable bathymetric outputs, particularly in low-cost and rapid survey scenarios.

1 INTRODUCTION

Development of accurate surfaces representing sea bottom, known as Digital Bottom Models (DBM), are nowadays fundamental in hydrography for sea and inland waters. These models are crucial for safe navigation, dredging works, offshore engineering or environmental monitoring. Based on discrete depth measurements, the models calculate a continuous surface, which reflects the shape of underwater area.

There are various methods and techniques used for bathymetric measurements. At the moment state-of-the-art Multibeam Echosounder Systems (MBES) are mostly used for hydrographic surveys, due to high data resolution and complex and wide bottom coverage they offer. However Single Beam Echosounders (SBES) still can be useful for many users in various applications. Their advantages include low pricing and small dimensions, as well as simplicity of measurement system. This makes them appropriate for small crafts and shallow waters. Therefore they are often mounted on Autonomous Surface Vehicles (ASV). These small crafts can be easily deployed in waters with restricted access for larger vehicles. In these waters ASV with SBES can be very useful although they do not offer full coverage, providing discrete measurements. To provide DBM, interpolation techniques have to be used, which can calculate the surface and grid from these measurements. Digital Bottom Models are usually calculated with the use of professional hydrographic or GIS software. However with rapid development of Data Science and Geodata Science fields, more and more processing methods are available in open source libraries. Such approach allow to provide DBMs created with interpolation techniques for the researchers and hydrographers without the need for purchasing specialized software. The libraries like SciPy or PyKrige offers flexible approach and wide access to the parameters of the methods. Many of commonly used methods are implemented, such as Inverse Distance Weighted (IDW), kriging, Natural Neighbor or Radial Basis Functions (RBF). They can be relatively easy integrated with self-prepared software.

Therefore the goal of this research was to examine interpolation methods offered by common open-source processing libraries and validate them for calculating Digital Bottom Model based on real acquired data.

This research was undertaken within the research project entitled Technology for intelligent processing of hydrographical data acquired with the use of imaging sonar and single beam echosounder, mounted on Autonomous Surface Vehicle, supported by the Foundation for Polish Science (FNP) in the FENG Proof of Concept programme under grant no. FENG.02.07-IP.05-0489/23. The goal of the entire project is to validate use of the AI for processing of underwater data. One of the tasks assumes processing of SBES data acquired in shallow waters with ASV. Part of the research is presented in this paper. In this paper we address the question how efficient is processing of relatively sparse SBES data with open-source interpolation techniques to obtain continuous surface. The Review of publications shows that researchers are interested in these subjects analyzing usage of various methods – deterministic, geostatistical and based on machine learning in different software for this purpose.

An example here is TFInterpy proposed in [1], showing how effective can be implementation of Kriging in Python with TensorFlow library. Simultaneously the tool called Grids, presented in [2] allows processing of spatiotemporal data for bathymetric interpolation and analysis, making use of Python implementation. Additionally, McEwen et al in [3] have shown that modern Python implementation can save up to 91% of processing time comparing to previous Fortran implementation. A set of interesting research, shows that in many applications use of Kriging shows many advantages over others. For example Wu et al. [4] shows that anisotropic Kriging or Thin-Plate Spline are significantly better in case of sparse SBES profiles, reducing RMSE error for about 20% compared to isotropic methods. Aykut et al. [5] have compared seven various interpolation methods for various shallow areas, showing Kriging as the most accurate method. In [6] Adediran et al. shows Spline interpolation as the most accurate and least uncertain method, however linear and IDW interpolation are also useful. Another open-source interpolation, based on IDW was proposed by Fleit in [7], providing also very good results. Very interesting approach is provided in Danish Depth Model, presented for example in Maseti et al. [8]. This publicly available model was created using open-source Python library and is available via OGC web services. In the research [9], very good interpolation accuracy was established with the use of SVR open source model for shallow areas. Additionally AI methods efficiency for this purpose is shown in [10,11] All of the above shows that open source libraries can be used for bathymetric data interpolation and that further research, such as proposed in this work, can show potential and further usage possibilities. Recent developments also include fully spaceborne and open-source workflows for regional bathymetry mapping, such as the ICESat-2 and Landsat-based approach proposed by [12] which eliminates the need for in-situ data collection. In contrast, this study emphasizes the practical value of ground-acquired SBES data and demonstrates how open-source interpolation techniques can effectively support bottom modeling in constrained operational environments. Small USV is proposed for data acquisition. This sector of vehicles is nowadays more often used for various measurements in shore areas. For example [13] shows usage of USV and AI for shore side data fusion.

2 RESEARCH CONCEPT

The objective of this study was to evaluate the effectiveness of interpolation methods available in open-source libraries for generating Digital Bottom Models (DBMs) from single-beam echo sounder (SBES) data. The analysis was conducted across four test areas characterized by varying bottom morphologies. These areas were selected for their shallow and hard-to-access conditions, where the use of SBES is justified and efficient interpolation enables the transformation of measured data into operationally useful surfaces. Five interpolation methods, described in Section 2.1, were included in the study. They represent a spectrum of techniques available in open-source tools—from deterministic numerical methods, through geostatistical approaches, to artificial neural networks. All methods were implemented using Python-based open-source libraries, including SciPy, PyKrige, and scikit-learn. Accuracy was assessed using k-fold cross-validation and standard statistical indicators, as outlined in Section 2.3. In addition to predictive accuracy, computational performance was analyzed to evaluate the suitability of each method for operational hydrographic tasks conducted under constrained technical conditions. A qualitative, expert-driven assessment of the resulting surfaces was also carried out.

2.1 *Interpolation methods*

The purpose of interpolation is to construct a regular grid-based model (GRID) from irregularly distributed measurement data. This requires estimating the depth value at a point with known, arbitrarily defined planar coordinates based on the depths recorded at surrounding measurement points. Different estimation strategies were applied depending on the method used—some relying on local neighborhoods around the target point, while others operated globally using all available data points to represent the entire modeled surface. Previous reviews have systematically compared spatial interpolation methods in both environmental sciences and terrain modeling contexts, highlighting the trade-offs between model complexity, data density, and prediction accuracy [14], [15].

2.1.1 *Inverse Distance Weighting – IDW*

Inverse Distance Weighting (IDW) is a deterministic spatial interpolation technique widely used for constructing Digital Elevation Models (DEMs) [16, 17]. The core principle of IDW is based on the assumption that the interpolated value at any unsampled location is a weighted average of known values, where the influence of each known point diminishes with distance from the prediction location [16]. The IDW method estimates the unknown value $Z(x_0)$ at location x0 as:

$$Z(x_0) = \frac{\sum_{i=1}^{n} w_i(x_0) \cdot Z(x_i)}{\sum_{i=1}^{n} w_i(x_0)} \tag{1}$$

where:
$Z(x_i)$ are the known values at locations x_i,
$w_i(x0) = w_i(x_0) = \frac{1}{d(x_0, x_i)^p}$ are the weights assigned based on distance d,
p is the power parameter that controls the rate of decay of influence with distance.

Typically, a power parameter p=2 is used, giving more influence to closer points and producing smoother surfaces. Higher values of p lead to steeper gradients and less smoothing, which can be beneficial in highly variable seabed conditions, though they may also enhance noise [15]. IDW is especially well-suited for interpolating bathymetric surfaces from SBES data, where measurements are collected along narrow vessel transects, resulting in sparse and anisotropic point distributions [17]. In such cases, IDW provides a straightforward means to generate continuous gridded surfaces without requiring complex statistical modeling.

2.1.2 *Ordinary Kriging – OK*

Ordinary Kriging (OK) is a geostatistical interpolation technique that incorporates the spatial autocorrelation of measured variables to estimate unknown values. It is considered one of the most robust and statistically grounded methods for surface modeling, especially when irregular spatial distribution [18], [19]. It's efficiency for bathymetry interpolation was shown e.g. in [20]. Ordinary Kriging estimates the unknown value $Z(x_0)$ at a location x_0 as a weighted linear combination of surrounding observations:

$$Z(x_0) = \sum_{i=1}^{n} \lambda_i \cdot Z(x_i) \tag{2}$$

subject to:

$$\sum_{i=1}^{n} \lambda_i = 1 \tag{3}$$

where:
$Z(x_i)$ are known values at sampled locations,
λ_i are the Kriging weights derived from the spatial correlation structure,

the assumption is that the mean of the process is constant but unknown within the local neighborhood.

The weights λ_i are obtained by solving the Kriging system of equations, which depends on the variogram model $\gamma(h)$ a function quantifying how the data's similarity decreases with distance:

$$\gamma(h) = \frac{1}{2} E\left[\left(Z(x) - Z(x+h)\right)^2\right] \tag{4}$$

A fitted empirical variogram is essential for applying Kriging correctly and is often based on exploratory data analysis of the dataset [18]. As described by Kholghi and Hosseini (2008), this variogram modeling phase is critical in ensuring the reliability of predictions, particularly in environmental and hydrological applications [21] . Ordinary Kriging is particularly advantageous in interpolation due to its ability to provide statistically optimal predictions (best linear unbiased estimator, BLUE), quantify interpolation uncertainty through Kriging variance, and adapt weighting according to both distance and spatial correlation among samples [18, 21].

2.1.3 *Nearest Neighbor – NN*

The Nearest Neighbor (NN) method is a simple spatial interpolation technique in which the value at an unsampled location is assigned to be exactly equal to that of the geographically closest measured point. Despite its simplicity, the Nearest Neighbor method remains a practical solution under specific conditions—particularly for rapid surface generation or in applications where data generalization should be avoided [5]. The NN algorithm assigns the value of the nearest observed point to an unknown location x_0:

$$Z(x_0) = Z(x_i)\, where\, i = \arg mind(x_0, x_i) \tag{5}$$

where:
$Z(x_0)$ is the estimated value at location x_0,
$Z(x_i)$ is the known value at the closest observed location,
$d(x_0,x_i)$ is the Euclidean distance between the prediction point and the i-th sample.

No averaging, weighting, or smoothing is applied, making the method non-interpolative in the statistical sense but exact in terms of data fidelity [22]. Nearest Neighbor is frequently used for generating categorical maps where discrete classification is needed, quick visualizations, especially in preprocessing stages and establishing base rasters for comparison with other interpolation methods [5].

2.1.4 *Radial Basis Function – RBF*

Radial Basis Function (RBF) interpolation is a flexible and powerful method for reconstructing smooth surfaces from irregularly spaced data, particularly useful in geoscientific and hydrographic applications [23]. Unlike global polynomial regressions or deterministic methods like IDW, RBFs create surfaces by combining radially symmetric functions centered at each known data point. This allows RBFs to adapt well to varying terrain complexity and to interpolate across sparse datasets with high continuity [23], [24]. The RBF interpolation of an unknown value at location x_0 is expressed as:

$$Z(x_0) = \sum_{i=1}^{n} \lambda_i \cdot \Phi(x_0 - x_i) \tag{6}$$

where:
λ_i are coefficients computed by solving a system of equations that ensures $Z(x_i)=z_i$,
$\phi(r)$ is a radial basis function, depending only on the Euclidean distance $r = \|x0 - xi\|$,
ϕ can take many forms — e.g. Gaussian, multiquadric, inverse multiquadric, linear, or thin plate spline. The function shape and smoothing parameters control the tradeoff between exact fit and surface regularity, enabling adaptation to different dataset characteristics.

In the context of this study, the Thin Plate Spline variant of the RBF method was selected due to its balance between surface smoothness and interpolation fidelity, particularly suited for the terrain variability observed in the test areas [25]. Thin Plate Spline is a specific type of RBF where the radial kernel is defined as:

$$\phi(r) = r^2 \log(r) \tag{7}$$

This kernel leads to a surface that minimizes the bending energy — essentially modeling the smoothest possible surface that still fits all known data points. Thin Plate Spline is favored for its visual smoothness and ability to model soft, continuous surface gradients, making it particularly effective in representing gentle terrain variations in shallow, low-relief environments.

2.1.5 *General Regression Neural Network – GRNN*

General Regression Neural Network (GRNN) [26] is a non-linear, memory-based neural network model designed for function approximation and regression tasks. It is particularly useful for spatial interpolation in cases where traditional deterministic or geostatistical methods may struggle to capture complex relationships, such as highly non-linear or noisy terrain data. GRNN is a supervised learning algorithm and belongs to the family of radial basis function networks, though its architecture and training dynamics differ significantly from standard RBF implementations. A GRNN consists of four layers, each contributing to the network's regression capability [27]. First, the input layer receives the spatial coordinates (e.g., x,y) of the location to be interpolated. Second, the pattern layer computes the Euclidean distance between the input vector and every training sample in the dataset, effectively forming a radial response. Third, the summation layer accumulates the weighted outputs and computes both the numerator and denominator of the prediction function. Finally, the output layer calculates the estimated value by dividing the weighted sum of outputs by the sum of the weights, yielding the interpolated value at the target location. The estimated output $Z(x_0)$ at a new location x_0 is computed using:

$$\hat{Z}(x_0) = \frac{\sum_{i=1}^{n} Z(x_i) \cdot \exp\left(-\frac{x_0 - x_i^2}{2\sigma^2}\right)}{\sum_{i=1}^{n} \exp\left(-\frac{x_0 - x_i^2}{2\sigma^2}\right)} \tag{8}$$

where:
$Z(x_i)$ are known target values,
σ is the smoothing parameter controlling generalization versus overfitting,
the exponential kernel ensures that nearby observations influence the prediction more strongly.

This formulation is conceptually similar to kernel density estimation and can be viewed as a soft, data-driven version of inverse distance weighting. GRNN has recently emerged as a competitive technique for modeling spatial surfaces from bathymetric and topographic data. Its key strength lies in modeling complex or non-stationary datasets — especially where classical interpolators like Kriging or TPS may impose excessive smoothness or require strong statistical assumptions. Although this study focuses primarily on classical spatial interpolation methods, recent research has shown that selected machine learning approaches—such as MLP and SVM—can also achieve high accuracy in bathymetric modeling from satellite imagery [28]–[30]. These findings provide relevant context for the inclusion of GRNN in the present comparison, as it represents a computationally efficient and easily implementable nonlinear alternative within an otherwise traditional interpolation framework.

2.2 *Open-source libraries*

Nowadays spatial analyses are more often conducted outside of professional GIS software or even beyond traditional geoinformatics platforms such as QGIS. The development of a broad set of open-access libraries for spatial data processing has led to the emergence of Spatial Data Science as a specialized branch of Data Science. Python has become the most widely used programming language for this purpose. For example, [31] presented the Marine Geospatial Ecology Tools (MGET), which combine Python, R, MATLAB, and ArcGIS to enable non-programmers to conduct advanced ecological modeling workflows in an open and extensible environment. Similarly, [32] developed Nansat, a modular, Python-based framework tailored for scientists working with satellite and model-derived geospatial data. It emphasizes interoperability, metadata integration, and automation, reinforcing the trend toward highly adaptable open-source tools for environmental data processing. Another example of open, modular software for spatial analytics is PySAL — a Python-based library that supports scalable, interoperable geospatial workflows through desktop, web, and cloud platforms [33].

This study employed three key Python libraries to perform spatial interpolation and validation procedures: PyKrige, SciPy, and scikit-learn. Each of these libraries provides a distinct set of tools tailored respectively to geostatistical modeling, numerical methods, and machine learning-based regression.

PyKrige is a Python library designed specifically for geostatistical applications [34]. It provides comprehensive functionality for spatial interpolation using Kriging methods. The library includes implementations of Ordinary Kriging, Universal Kriging, and Kriging with external drift. It allows users to construct and fit variogram models either manually or automatically, with options including spherical, exponential, Gaussian, and linear models.

SciPy, while not dedicated to spatial statistics, includes a set of powerful interpolation tools that are frequently used in terrain modeling [35]. Within the scipy.interpolate module, the griddata() function was employed for deterministic interpolation methods such as Nearest Neighbor, Linear, and Cubic interpolation. The R() function, also from this module, enabled implementation of Radial Basis Function interpolation using kernels such as multiquadric, inverse multiquadric, linear, Gaussian, and thin-plate spline. The library also includes fast spatial indexing structures such as KD-Trees through the scipy.spatial module, which were useful for nearest-neighbor lookups during validation. Although SciPy does not provide dedicated tools for model validation, its outputs were easily passed to NumPy for calculating standard validation metrics such as RMSE and MAE.

Scikit-learn is a general-purpose machine learning library that provides robust tools for regression modeling, cross-validation, and error analysis [36]. Although it is not spatially aware by default, it can be adapted for interpolation tasks by treating them as regression problems. In this study, it was primarily used to implement the General Regression Neural Network (GRNN) approach, using multilayer perceptron regressors or custom kernel-based estimators. The library's KFold and cross_validate routines were used to systematically evaluate model performance across multiple subsets of the data, and standard metrics such as root mean square error, mean absolute error, and the coefficient of determination (R^2) were computed using the metrics module. While scikit-learn does not support variogram modeling or spatially weighted kernels natively, it offers excellent capabilities for parameter tuning and model optimization, including grid search procedures.

2.3 *Validation metrics*

Evaluating the effectiveness of interpolation methods is crucial in spatial modeling to ensure that the predicted surfaces reliably reflect the actual terrain values. In the context of Digital Bottom Models (DBMs), this is particularly important when interpolating from sparse or irregular SBES measurements. Model validation involves comparing predicted values against a set of known measurements that were not used during the interpolation process. Several quantitative metrics are routinely applied to assess interpolation accuracy, each providing insight into different aspects of model performance. They are usually related to reference values or surfaces. In case of the research in this paper the reference values were depths measured directly during campaign.

2.3.1 *Prediction errors*

Mean Absolute Error (MAE) represents the average absolute difference between observed and predicted values. Unlike RMSE, it gives equal weight to all errors, making it more robust in the presence of extreme values [37].

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|Z(x_i) - \hat{Z}(x_i)\right| \tag{9}$$

Root Mean Square Error (RMSE) is one of the most widely used metrics and indicates the average magnitude of prediction errors. It penalizes large deviations more heavily than small ones and is sensitive to outliers. Lower RMSE values indicate a better fit [37].

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(Z(x_i) - \hat{Z}(x_i)\right)^2} \tag{10}$$

The coefficient of determination, denoted as R^2, is used statistical metric in spatial interpolation and regression analysis to quantify the proportion of variance in the observed data that is explained by the model. It provides a normalized measure of model accuracy, ranging from 0 to 1, where a value closer to 1 indicates a better fit between predicted and observed values [37]. Mathematically, R^2 is defined as:

$$R^2 = \frac{\left(\sum_{i=1}^{n}\left(Z_i - \bar{Z}\right)\cdot\left(\hat{Z}_i - \bar{\hat{Z}}\right)\right)^2}{\sum_{i=1}^{n}\left(Z_i - \bar{Z}\right)^2 \cdot \sum_{i=1}^{n}\left(\hat{Z}_i - \bar{\hat{Z}}\right)^2} \tag{11}$$

where:
Z_i is the observed value at point i,
$\hat{Z}_i$ is the estimated (interpolated) value at the same point,
$\hat{Z}_i$ and $\bar{\hat{Z}}$ are the means of observed and estimated values, respectively,
n is the number of validation points.

In this study, interpolation accuracy was assessed using k-fold cross-validation, applied exclusively to the SBES dataset. The dataset was partitioned into k equally sized subsets. For each fold, one subset was withheld as a validation set, while the remaining k − 1 subsets were used to train the interpolation model. This procedure was repeated k times so that each data point was used for validation exactly once. Predicted values at validation locations were compared to their corresponding measured depths, and the differences were aggregated using standard error metrics, including Root Mean Square Error (RMSE), Mean Absolute Error (MAE) and Coefficient of determination (R^2).

In addition to traditional accuracy metrics, the validation framework incorporated assessments of execution time and memory usage to quantify the computational efficiency of each interpolation method. These metrics were measured independently of interpolation accuracy and applied uniformly across all test scenarios. Execution time was recorded using Python's built-in time module. The total duration was computed as the difference in seconds and subsequently averaged across all folds to yield a representative measure of computational demand per method. Memory usage was monitored using the memory_profiler Python package. For consistency, memory profiling was conducted under identical hardware conditions and for each method operating on the full training dataset within a given fold.

3 RESEARCH

The research aims at comparison of various interpolation methods, based on the real data gathered on Unmanned Surface Vehicle and processed with open-source libraries. Thus measurement equipment and software test environment are crucial. The data was acquired in 4 various test areas to validate performance of methods for different bottom characteristics.

3.1 *Description of the test areas, equipment and characteristics of the data collected*

Test data for interpolation were collected using an unmanned surface vehicle (USV) integrated with a single beam echosounder (SBES) (Figure 1). The Echologger EU400 was used in the survey. Table 1 shows the parameters of the vehicle and the echosounder themselves.

Figure 1. Autonomous surface vehicles (USV) prepared for bathymetric survey on the Odra River, Szczecin.

Table 1. USV survey system parameters [38]

Single-beam echosounder		Unmanned survey vehicle		Positioning system	
Frequency	450 kHz	Length	1200 mm	Static accuracy	Static horizontal 4 mm + 0.5 ppm Static vertical 8 mm + 1 ppm
Beam width	5° Conical (-3 dB)	Width	1000 mm	Kinematic accuracy	Kinematic vertical 14 mm + 1 ppm Kinematic horizontal 7 mm + 1 ppm
Transmit pulse width	10~200 μsec (10 μsec step)	Height	360 mm (without mast)	Signals tracked simultaneously	GPS/QZSS, GLONASS, BeiDou, Galileo
Ranges	0.15 ~100 m	Survey speed	1.2 m/s		
Data output format	ASCII TXT, NMEA0183	Thrusters	2 x T200 Blue Robotics		

The data were saved in XYZ format, where XY values specify geographical coordinates and Z values specify depth. The data were collected in the WGS 1984 UTM 33N coordinate system (EPSG code: 32633). The surveys were conducted in four test areas located in north-western Poland. These areas include sections of the Odra River and east side of Lake Dabie. The first area, referred to as CEOP, is located along the Odra River, adjacent to the Marine Facilities Operation Centre of the Maritime University in Szczecin. This site exhibits moderate bathymetric variability, with recorded depths ranging from –0.15 m to 6.44 m. A total of 3 222 valid depth measurements were collected. The mean depth was 3.51 m, while the median was slightly higher at 3.68 m. The second area, OSRM, is also situated along the Odra River, near the Maritime Rescue Training Centre. Depths in this section ranged from 2.26 m to 5.73 m, based on 3 141 collected measurements. The average depth was 3.83 m, with a median of 3.88 m, indicating a relatively uniform depth distribution. The third test site, Czarna Laka, is located within the Bystrzanska Bay and represents a significantly shallower environment compared to the river sections. Depths ranged from 0.34 m to 4.03 m, based on 10 706 measurements. The mean depth was just 0.94 m, while the median was 0.69 m, indicating a right-skewed distribution dominated by very shallow areas. The fourth and largest dataset was acquired in Lubczyna, a part of Lake Dąbie. Here, the USV collected 113 068 individual depth measurements—by far the most extensive dataset among the four areas. Depths ranged from 0.87 m to 3.83 m, with a mean depth of 2.78 m and a median of 2.88 m, suggesting a relatively smooth and uniform bathymetric profile.

3.2 *Testing software SonarMUS*

The implementation and testing of bottom interpolation methods were carried out using proprietary software called SonarMUS, developed as part of the project titled "Technology for intelligent processing of hydrographical data acquired with the use of imaging sonar and single beam echosounder, mounted on Autonomous Surface Vehicle," funded by the Foundation for Polish Science under grant number FENG.02.07-IP.05-0489/23. The project aims to develop AI-based methods for processing sonar and single-beam echosounder data. As part of this effort, the SonarMUS visual and testing environment was created to enable qualitative evaluation of implemented methods and system functionalities. In the context of SBES data processing, SonarMUS includes a dedicated preprocessing module that handles data filtering, reduction techniques, and the generation of isobaths, along with the interpolation methods described in the previous section. This software environment served as the basis for conducting the experimental research presented in this article.

4 RESULTS

A comprehensive evaluation of the effectiveness of interpolation methods implemented using open-source libraries requires both quantitative and qualitative assessment. It is worth noting that the study was conducted using data from real-world water bodies rather than simulated datasets, which prevents direct comparison with a externally modelled reference surface.

4.1 *Quantitative evaluation*

For validation purposes, a 10-fold cross-validation procedure was applied, in which the dataset was divided into ten equal subsets, and models were iteratively trained on 90% of the data and tested on the remaining 10%. Prediction accuracy was evaluated using RMSE, MAE, and R^2 metrics, allowing direct comparison of all methods. Validation results for all interpolation methods and test areas are presented in Figures 2–6.

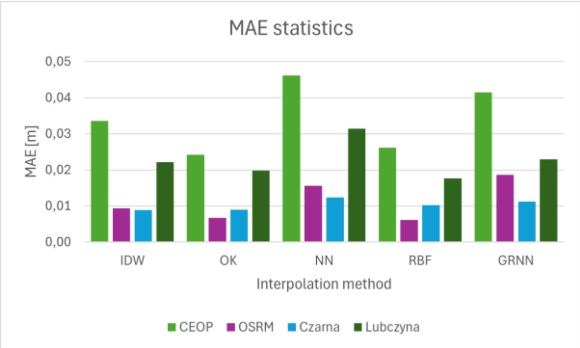

Figure 2. MAE statistics for all methods and test areas

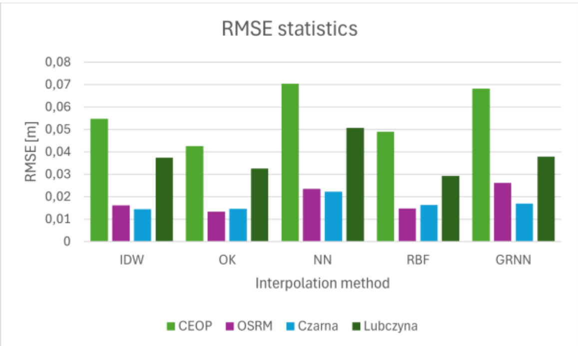

Figure 3. RMSE statistics for all methods and test areas

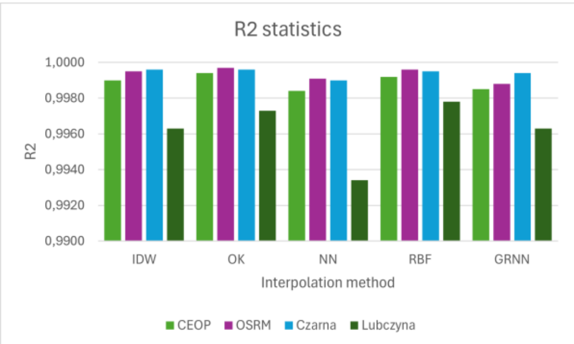

Figure 4. R^2 statistics for all methods and test areas

Figures 2-4 show that all tested interpolation methods performed generally well, with high R^2 values across most test areas, typically exceeding 0.9. This indicates a strong fit between predicted and observed depths. However, the Lubczyna site revealed more pronounced differences between methods. Here, Ordinary Kriging (OK) stood out as the most accurate in all metrics, while GRNN, despite maintaining a high R^2, showed higher RMSE and MAE values than both RBF and IDW. These differences are likely due to the sparse and uneven distribution of measurement points in Lubczyna, where methods that explicitly account for spatial relationships, such as Ordinary Kriging, retained higher precision. In contrast, data-driven approaches like GRNN were more sensitive to noise and prone to overfitting under these conditions. When considering RMSE and MAE more broadly, OK consistently achieved the lowest errors across all areas, followed by RBF-TPS and IDW, which demonstrated stable performance. GRNN showed less favorable results in error magnitude, outperforming only the Nearest Neighbor (NN) method, which, lacking spatial averaging, yielded the highest error values. It is also notable that error values were higher in the CEOP area. This site is characterized by a greater depth range, which may have increased interpolation difficulty—particularly for methods sensitive to depth variability or local anomalies.

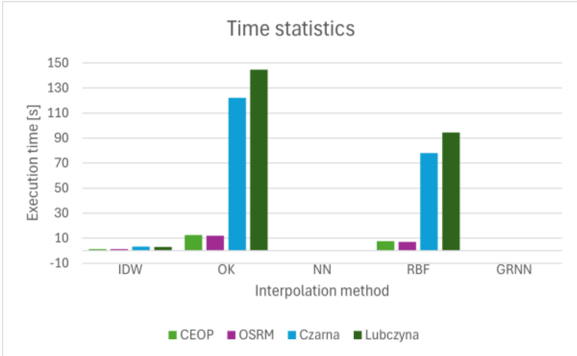

Figure 5. Interpolation execution time statistics for all methods and test areas

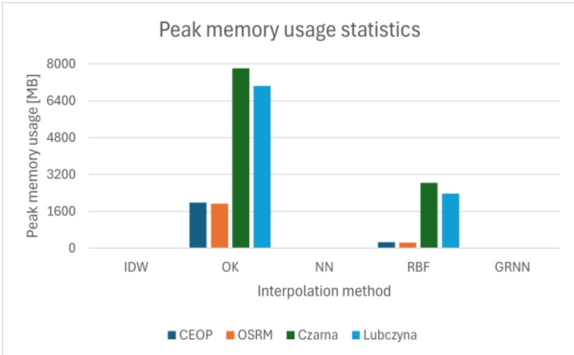

Figure 6. Peak memory usage statistics during interpolation for all methods and test areas

In terms of execution time, the fastest methods were NN, IDW, and GRNN, due to their algorithmic simplicity or limited model training demands. RBF-TPS exhibited moderate computation time, while Ordinary Kriging (OK) was consistently the slowest—particularly for larger datasets such as Lubczyna, where variogram fitting incurred significant processing overhead. On the Figures 5 and 6, some bars are missing for certain methods and test areas. This is not due to lack of data, but rather because the recorded values were extremely low, often below the plotting resolution. Their minimal computational footprint caused their bars to effectively disappear from the visualizations. In terms of memory usage, the lowest consumption was observed for NN, IDW, and GRNN. GRNN, despite involving non-linear regression operations, exhibited a compact memory profile due to its localized computation model. In contrast, OK demonstrated the highest peak memory usage, driven by the need for variogram modeling and solving large kriging systems, especially when applied to dense or high-resolution datasets.

4.2 *Qualitative evaluation*

In addition to the quantitative verification of interpolation performance, qualitative evaluation of the generated surfaces is equally important from a practical standpoint. While statistical measures such as MAE, RMSE, and R^2 provide insight into overall prediction accuracy at test points, they do not capture the full picture of the visual quality and spatial coherence of the resulting Digital Bottom Models (DBMs). Figures 7–11 show the interpolation results for each method applied to the CEOP test area. These visualizations support the qualitative analysis by allowing direct comparison of surface smoothness, continuity, and visible artifacts.

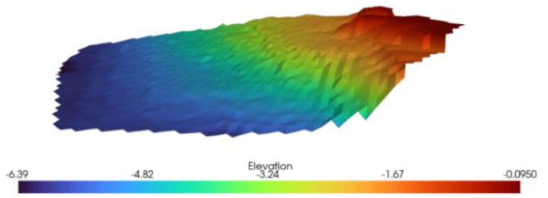

Figure 7. IDW-generated interpolation for the CEOP area

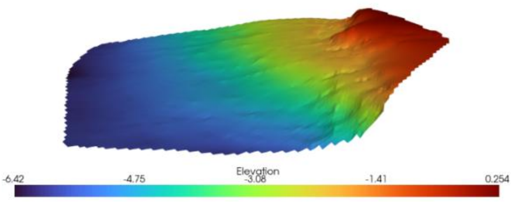

Figure 8. OK-generated interpolation for the CEOP area

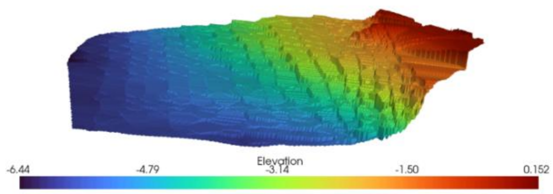

Figure 9. NN-generated interpolation for the CEOP area

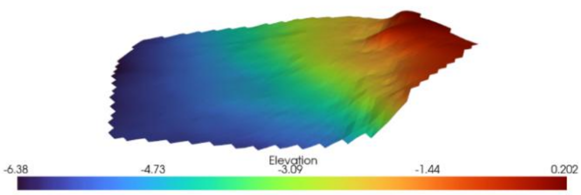

Figure 10. RBF TPS-generated interpolation for the CEOP area

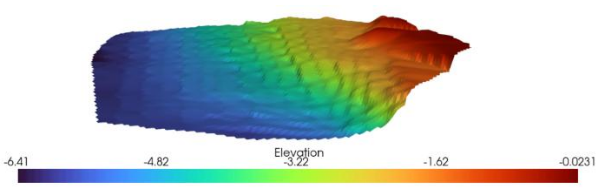

Figure 11. GRNN-generated interpolation for the CEOP area

The interpolation results using thin plate spline radial basis functions (RBF-TPS) are characterized by a high level of smoothing and visually appealing surface continuity. The RBF model effectively suppresses noise and ensures natural depth transitions, resulting in a clear representation of bottom morphology. Despite a slight tendency to oversmooth local details, the method maintains a realistic depiction and is particularly useful in applications requiring consistent and continuous surfaces. In contrast, the Nearest Neighbor (NN) method produces a model with very low spatial continuity. The visualization reveals sharp, irregular edges and block-like artifacts. The lack of smoothing and true interpolation results in a surface of low visual quality and limited practical value, except perhaps as a reference layer or for quick previews. The IDW method provides a moderate level of smoothness, but the resulting surface exhibits noticeable local inconsistencies. This is due to the excessive influence of nearby measurement points and the method's inability to adapt to the broader spatial structure of the dataset. Consequently, the surface appears fragmented and uneven, especially in areas with more complex depth variation. The best visual results were achieved using Ordinary Kriging (OK). This method demonstrated excellent capability in representing the actual shape of the bottom, preserving local gradients and maintaining high spatial coherence. The resulting surface is smooth and artifact-free, while avoiding excessive generalization. The use of a fitted variogram model allows for precise adaptation to the data distribution, yielding a realistic and reliable bathymetric model. The resulting surface is highly continuous and accurately reflects bottom structures, comparable to the output from Kriging. Some minor blurring of local features is visible, likely due to the nature of the kernel functions used in the GRNN architecture. In summary, from a visual quality perspective, the most favorable results were obtained with OK, followed by GRNN and RBF. All three methods provide coherent and realistic bottom representations suitable for bathymetric analysis. In contrast, while IDW and NN offer simplicity and low computational demands, they are limited in surface quality and prone to artifacts, which significantly restricts their applicability in more demanding hydrographic tasks.The interpolation results using thin plate spline radial basis functions (RBF-TPS) are characterized by a high degree of surface continuity and effective noise suppression. In the CEOP area, the method produced a visually smooth surface, though with a tendency to oversimplify local variations. This makes RBF-TPS well-suited for gently varying terrains but potentially less precise in areas where sharp depth gradients are present. The Nearest Neighbor (NN) method produced a surface with minimal spatial coherence. As visible in the CEOP test area, the result exhibits abrupt transitions and block-like artifacts, reflecting the method's lack of smoothing and interpolation. Despite its low visual quality, NN may still be useful for quick previews or for visualizations that strictly reflect the influence of the nearest measured values without modification. The IDW method yielded intermediate visual quality. It preserved more local variability than NN and performed reasonably well in CEOP. However, it sometimes exaggerated the influence of nearby points, particularly in areas with irregular data density, resulting in surfaces that appeared locally distorted or fragmented. Ordinary Kriging (OK) provided the most visually convincing results. In the CEOP area, it preserved depth gradients and localized features while maintaining high smoothness and spatial consistency. The fitted variogram enabled flexible adaptation to the spatial structure of the data, yielding a realistic and coherent representation of bottom morphology. The Digital Bottom Model generated using the General Regression Neural Network (GRNN) for the CEOP area displays a surface that is neither well-smoothed nor morphologically accurate. While GRNN avoids the sharp, blocky transitions typical of the Nearest Neighbor method, it also fails to capture the complexity of the bottom terrain. The result is a visually coarse surface with a gridded appearance and poor representation of local features. This can be attributed to inadequate parameter tuning, particularly the smoothing factor in the kernel function, which leads to underfitting and spatial generalization. In the CEOP area, where significant depth variation and sharp gradients are present, the GRNN output lacks resolution and realism. Although the method has potential in smoother basins, its performance in complex environments like CEOP is unsatisfactory, producing results more similar in structure to NN than to advanced interpolators such as Ordinary Kriging or RBF. In summary, Ordinary Kriging yielded the most visually accurate and spatially consistent results, followed by RBF-TPS. GRNN offered a general, but structurally weak representation that underperformed in complex areas. IDW and NN were clearly limited in surface quality, though their computational simplicity may justify their use in rapid or exploratory analyses.

5 CONCLUSIONS

This article presents a study aimed at analyzing the applicability of a data processing methodology derived from Data Science—based on open-source computational libraries—for the interpolation of Digital Bottom Models using single-beam echosounder data. The study was based on real-world data acquired using a single-beam echosounder mounted on an unmanned surface vehicle owned by the Maritime University of Szczecin. The implementation was carried out within a proprietary visual and testing environment – SONARMUS – developed as part of a dedicated research project. The conducted study demonstrated that open-source interpolation methods can effectively generate accurate Digital Bottom Models (DBMs) from single-beam echosounder (SBES) data. Among the five tested methods, Ordinary Kriging consistently achieved the highest interpolation accuracy, yielding the lowest RMSE and MAE values across all test areas and producing spatially coherent surfaces suitable for detailed bathymetric analysis. The RBF-TPS and GRNN methods also showed strong performance, particularly in areas with gradual depth transitions, making them valuable alternatives in workflows that require smooth surface generation or data-driven flexibility. In contrast, IDW and Nearest Neighbor methods exhibited higher error levels and limited adaptability to local terrain complexity, although they offered advantages in computational efficiency and ease of implementation—beneficial for rapid assessments or operations with limited resources.

The results confirm that open-source tools such as PyKrige, SciPy, and scikit-learn provide a viable and cost-effective foundation for hydrographic data processing workflows. They enable small teams and budget-constrained institutions to conduct professional-grade bathymetric modeling without relying on commercial software. With appropriate configuration and validation, these tools can not only complement but, in many cases, effectively replace proprietary solutions—especially in hydrographic operations carried out under budgetary constraints, where transparency, reproducibility, and flexibility are as critical as accuracy.

Future research could focus on hybrid approaches that combine the adaptive, non-linear learning capabilities of GRNN with the statistical rigor of classical geostatistical methods such as Kriging. This integration could enable more robust interpolation performance in complex or noisy environments, where GRNN alone may underperform. Specifically, coupling GRNN with variogram-informed weighting schemes or using Kriging-derived residuals as input features may offer an effective compromise between data-driven flexibility and spatial coherence.

FUNDING

This work was supported by the Foundation for Polish Science (FNP) in the FENG Proof of Concept programme under grant no. FENG.02.07-IP.05-0489/23, entitled Technology for intelligent processing of hydrographical data acquired with the use of imaging sonar and single beam echosounder, mounted on Autonomous Surface Vehicle.

REFERENCES

[1] J. Chen and W. Zhong, "TFInterpy: A high-performance spatial interpolation Python package," SoftwareX, vol. 18, p. 101076, 2022.
[2] R. C. Hales, E. J. Nelson, G. P. Williams, N. Jones, D. P. Ames, and J. E. Jones, "The Grids Python Tool for Querying Spatiotemporal Multidimensional Water Data," Water, vol. 13, no. 15, p. 2066, Jul. 2021, doi: 10.3390/w13152066.
[3] T. McEwen, D. Pothina, and S. Negusse, "Improving efficiency and repeatability of lake volume estimates using Python," Proc. 10th Python in Science Conf. (SciPy), 2011, pp. 114–118.
[4] C.-Y. Wu, J. Mossa, L. Mao, and M. Almulla, "Comparison of different spatial interpolation methods for historical hydrographic data of the lowermost Mississippi River," Annals of GIS, vol. 25, no. 2, pp. 133–151, 2019, doi: 10.1080/19475683.2019.1588781.
[5] N. O. Aykut, B. Akpınar, and Ö. Aydın, "Hydrographic data modeling methods for determining precise seafloor topography," Comput. Geosci., vol. 17, pp. 661–669, 2013, doi: 10.1007/s10596-013-9347-1.
[6] E. Adediran, C. Kastrisios, K. Lowell, G. Rice, and Q. Zhang, "Toward Quantifying Interpolation Uncertainty in Set-Line Spacing Hydrographic Surveys," ISPRS Int. J. Geo-Inf., vol. 14, no. 2, p. 90, Feb. 2025, doi: 10.3390/ijgi14020090.
[7] G. Fleit, "Windowed anisotropic local inverse distance-weighted (WALID) interpolation method for riverbed mapping," Acta Geophysica, vol. 73, pp. 2819–2833, 2025, doi: 10.1007/s11600-024-01510-4.
[8] G. Masetti et al., "Denmark's Depth Model: Compilation of Bathymetric Data within the Danish Waters,"

Geomatics, vol. 2, no. 4, pp. 486–498, 2022, doi: 10.3390/geomatics2040026.
[9] M. Specht et al., "Analysis of Methods for Determining Shallow Waterbody Depths Based on Images Taken by Unmanned Aerial Vehicles," Sensors, vol. 22, no. 5, Art. no. 1844, Feb. 2022, doi: 10.3390/s22051844.
[10] A. Jaszcz, M. Włodarczyk-Sielicka, A. Stateczny, D. Połap, and I. Garczyńska, "Automated shoreline segmentation in satellite imagery using USV measurements," Remote Sensing, vol. 16, no. 24, p. 4457, 2024, doi: 10.3390/rs16234457.
[11] K. Prokop, D. Połap, M. Włodarczyk-Sielicka, K. Połap, A. Jaszcz, and A. Stateczny, "Automated shoreline extraction process for unmanned vehicles via U-net with heuristic algorithm," Alexandria Engineering Journal, vol. 102, pp. 108–118, 2024, doi: 10.1016/j.aej.2024.05.104. K. Prokop, D. Połap, M. Włodarczyk-Sielicka, K. Połap, A. Jaszcz, and A. Stateczny, "Automated shoreline extraction process for unmanned vehicles via U-net with heuristic algorithm," Alexandria Engineering Journal, vol. 102, pp. 108–118, 2024, doi: 10.1016/j.aej.2024.05.104.
[12] N. Thomas, B. Lee, O. Coutts, P. Bunting, D. Lagomasino, and L. Fatoyinbo, "A purely spaceborne open source approach for regional bathymetry mapping," IEEE Transactions on Geoscience and Remote Sensing, vol. 60, pp. 1–13, 2022, Art. no. 4708109, doi: 10.1109/TGRS.2022.3192825.
[13] I. Garczyńska-Cyprysiak, W. Kazimierski, and M. Włodarczyk-Sielicka, "Neural approach to coordinate transformation for LiDAR–camera data fusion in coastal observation," Sensors, vol. 24, no. 20, p. 6766, 2024, doi: 10.3390/s24206766.
[14] J. Li and A. D. Heap, "Spatial interpolation methods applied in the environmental sciences: A review," Environmental Modelling & Software, vol. 53, pp. 173–189, 2014, doi: 10.1016/j.envsoft.2013.12.008.
[15] R. Gradka and A. Kwinta, "A short review of interpolation methods used for terrain modeling," Geomatics, Landmanagement and Landscape, vol. 4, pp. 29–47, 2018, doi: 10.15576/GLL/2018.4.29.
[16] W. Lu and D. W. S. Wong, "An Adaptive Inverse-Distance Weighting Spatial Interpolation Technique," Computers & Geosciences, vol. 34, no. 9, pp. 1044–1055, Sept. 2008. doi: 10.1016/j.cageo.2007.07.010
[17] S. Salekin, J. H. Burgess, J. Morgenroth, E. G. Mason, and D. F. Meason, "A Comparative Study of Three Non-Geostatistical Methods for Optimising Digital Elevation Model Interpolation," ISPRS International Journal of Geo-Information, vol. 7, no. 8, p. 300, Jul. 2018. doi: 10.3390/ijgi7080300
[18] N. A. C. Cressie, "The Origins of Kriging," Mathematical Geology, vol. 22, no. 3, pp. 239–252, Apr. 1990, doi: 10.1007/BF00889887.
[19] A. G. Journel and C. J. Huijbregts, Mining Geostatistics, London: Academic Press, 1978.
[20] P. Biernacik, W. Kazimierski, and M. Włodarczyk-Sielicka, "Comparative analysis of selected geostatistical methods for bottom surface modeling," Sensors, vol. 23, no. 8, p. 3941, 2023, doi: 10.3390/s23083941.
[21] M. Kholghi and S. M. Hosseini, "Comparison of Groundwater Level Estimation Using Neuro-fuzzy and Ordinary Kriging," Environmental Modeling & Assessment, vol. 14, no. 6, pp. 729–737, Dec. 2009, doi: 10.1007/s10666-008-9174-2.
[22] C. Du, "An interpolation method for grid-based terrain modeling," The Computer Journal, vol. 39, no. 10, pp. 837–843, 1996, doi: 10.1093/comjnl/39.10.837.
[23] R. Freedman, "New approach for solving inverse problems encountered in well-logging and geophysical applications," Petrophysics, vol. 47, no. 2, pp. 93–111, Apr. 2006.
[24] Y.-L. Zou, F.-L. Hu, C.-C. Zhou, C.-L. Li, and K.-J. Dunn, "Analysis of radial basis function interpolation approach," Appl. Geophys., vol. 10, no. 4, pp. 397–410, Dec. 2013, doi: 10.1007/s11770-013-0407-z.
[25] W. Keller and A. Borkowski, "Thin plate spline interpolation," J. Geod., vol. 93, no. 8, pp. 1251–1269, Aug. 2019, doi: 10.1007/s00190-019-01240-2.
[26] D. F. Specht, "A general regression neural network," IEEE Transactions on Neural Networks, vol. 2, no. 6, pp. 568–576, 1991.
[27] H. K. Ghritlahre and R. K. Prasad, "Exergetic performance prediction of solar air heater using MLP, GRNN and RBF models of artificial neural network technique," Journal of Environmental Management, vol. 223, pp. 566–575, 2018, doi: 10.1016/j.jenvman.2018.06.033.
[28] S. Liu, Y. Gao, W. Zheng, and X. Li, "Performance of two neural network models in bathymetry," Remote Sensing Letters, vol. 6, no. 4, pp. 321–330, 2015, doi: 10.1080/2150704X.2015.1034885.
[29] Z. Duan, S. Chu, L. Cheng, C. Ji, M. Li, and W. Shen, "Satellite-derived bathymetry using Landsat-8 and Sentinel-2A images: assessment of atmospheric correction algorithms and depth derivation models in shallow waters," Optics Express, vol. 30, no. 3, pp. 3238–3261, 2022, doi: 10.1364/OE.444557.
[30] A. Misra, Z. Vojinovic, B. Ramakrishnan, A. Luijendijk, and R. Ranasinghe, "Shallow water bathymetry mapping using Support Vector Machine (SVM) technique and multispectral imagery," International Journal of Remote Sensing, vol. 39, no. 14, pp. 4739–4765, 2018, doi: 10.1080/01431161.2017.1421796.
[31] J. J. Roberts, B. D. Best, D. C. Dunn, E. A. Treml, and P. N. Halpin, "Marine Geospatial Ecology Tools: An integrated framework for ecological geoprocessing with ArcGIS, Python, R, MATLAB, and C++," Environmental Modelling & Software, vol. 25, no. 10, pp. 1197–1207, 2010, doi: 10.1016/j.envsoft.2010.03.029.
[32] A. A. Korosov, M. W. Hansen, K.-F. Dagestad, A. Yamakawa, A. Vines, and M. Riechert, "Nansat: a scientist-orientated Python package for geospatial data processing," Journal of Open Research Software, vol. 4, 2016, Art. no. e39, doi: 10.5334/jors.120.
[33] S. J. Rey, L. Anselin, X. Li, R. Pahle, J. Laura, W. Li, and J. Koschinsky, "Open geospatial analytics with PySAL," ISPRS Int. J. Geo-Inf., vol. 4, no. 2, pp. 815–836, 2015, doi: 10.3390/ijgi4020815.
[34] B. Murphy, S. Müller, and R. Yurchak, GeoStat-framework/PyKrige v1.5.1 (Version v1.5.1) [Computer software], Zenodo, Aug. 2020. [Online]. Available: https://doi.org/10.5281/zenodo.3991907
[35] P. Virtanen et al., "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python," Nature Methods, vol. 17, pp. 261–272, 2020. [Online]. Available: https://www.nature.com/articles/s41592-019-0686-2
[36] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011. [Online]. Available: https://www.jmlr.org/papers/volume12/pedregosa11a/pedregosa11a.pdf
[37] H. Junninen, H. Niska, K. Tuppurainen, J. Ruuskanen, and M. Kolehmainen, "Methods for imputation of missing values in air quality data sets," Atmospheric Environment, vol. 38, no. 18, pp. 2895–2907, Jun. 2004, doi: 10.1016/j.atmosenv.2004.02.026.
[38] J. Lubczonek, W. Kazimierski, G. Zaniewicz, and M. Lacka, "Methodology for combining data acquired by unmanned surface and aerial vehicles to create digital bathymetric models in shallow and ultra-shallow waters," Remote Sensing, vol. 14, no. 1, p. 105, 2022, doi: 10.3390/rs14010105.