

Article

Optimization of Water Quality Monitoring Networks Using Metaheuristic Approaches: Moscow Region Use Case

Elizaveta Yudina ¹ , Anna Petrovskaia ^{1,2,*} , Dmitrii Shadrin ^{1,2} , Polina Tregubova ² , Elizaveta Chernova ²,
Mariia Pukalchik ^{1,2}  and Ivan Oseledets ^{1,3} 

- ¹ Center for Computational and Data-Intensive Science and Engineering, Skolkovo Institute of Science and Technology, 143026 Moscow, Russia; elizaveta.n.yudina@gmail.com (E.Y.); Dmitry.Shadrin@skolkovotech.ru (D.S.); m.pukalchik@skoltech.ru (M.P.); i.oseledets@skoltech.ru (I.O.)
- ² Digital Agriculture Laboratory, Skolkovo Institute of Science and Technology, 143026 Moscow, Russia; p.tregubova@skoltech.ru (P.T.); e.chernova@skoltech.ru (E.C.)
- ³ Marchuk Institute of Numerical Mathematics, RAS, 119333 Moscow, Russia
- * Correspondence: anna.petrovskaia@skoltech.ru

Abstract: Currently many countries are struggling to rationalize water quality monitoring stations which is caused by economic demand. Though this process is essential indeed, the exact elements of the system to be optimized without a subsequent quality and accuracy loss still remain obscure. Therefore, accurate historical data on groundwater pollution is required to detect and monitor considerable environmental impacts. To collect such data appropriate sampling and assessment methodologies with an optimum spatial distribution augmented should be exploited. Thus, the configuration of water monitoring sampling points and the number of the points required are now considered as a fundamental optimization challenge. The paper offers and tests metaheuristic approaches for optimization of monitoring procedure and multi-factors assessment of water quality in “New Moscow” area. It is shown that the considered algorithms allow us to reduce the size of the training sample set, so that the number of points for monitoring water quality in the area can be halved. Moreover, reducing the dataset size improved the quality of prediction by 20%. The obtained results convincingly demonstrate that the proposed algorithms dramatically decrease the total cost of analysis without dampening the quality of monitoring and could be recommended for optimization purposes.

Keywords: water quality network optimization; genetic algorithm; variable neighborhood search; water quality index; groundwater



Citation: Yudina, E.; Petrovskaia, A.; Shadrin, D.; Tregubova, P.; Chernova, E.; Pukalchik, M.; Oseledets, I. Optimization of Water Quality Monitoring Networks Using Metaheuristic Approaches: Moscow Region Use Case. *Water* **2021**, *13*, 888. <https://doi.org/10.3390/w13070888>

Academic Editor: George Besseris

Received: 18 February 2021

Accepted: 15 March 2021

Published: 24 March 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Water quality monitoring networks are essential for regulatory bodies including governments and policymakers for evaluating and managing water pollution. In the last fifty years, water quality monitoring networks of different spatial and temporal scales have been created by various environmental agencies and regulators in many countries [1]. However, taking into account the increasing rate of human industrial activity and growth of the population, we must admit that the monitoring networks established decades ago may be insufficient to meet the demands imposed by the current trends in pollution. Moreover, the system should be not only efficient but also cost-effective. One of the topical issues that should be addressed is that an average concentration over the monitoring sites is sensitive to biases in the monitoring network. Though it may seem to be effective to create a well-distributed network of locations exclusively, the actual selection of points for water quality monitoring depends on several site-specific factors: what the locations of the existing wells, springs, and rivers are; how suitable these sources could be by for water sampling; whether they are accessible or not; what budget and time are available. Merging and improving older networks not infrequently may result in subsequent streamlining,

rationalizing, and modifying their composition. Another approach to optimize current monitoring methodologies is harnessing new ecological monitoring data. This is especially useful for a water monitoring program on a regional scale. The aforementioned issues could be addressed using machine learning approaches and various optimization algorithms.

The approaches currently proposed to tackle the problem of optimizing water quality monitoring networks can be roughly classified into two categories: analytical methods and meta-heuristic algorithms. Among the former multi-criteria decision and multivariate statistical approaches seem to be most popular [2–4]. Some studies suggest clustering of water quality data based on a dynamic algorithm for choosing optimal water sampling locations [5]. The other one apply graph theory combined with a simulated annealing for optimized selection of river sampling sites [6]. There is a research proposal to use Principal Component Analysis [7] in order to find optimum monitoring positions and ultimately to reduce the monitoring costs.

Most studies focus on searching for and selecting an optimal position to install the equipment for observing large surface water bodies, such as rivers and lakes. Meanwhile, natural groundwater outlets, such as wells and springs, have yet received much less attention [8], since they normally appear to be among the least damaged freshwaters in the investigated area [9]. Additionally, at present, only few articles encompass the information about the integrated water quality index [10].

The problem of finding the best subset of training samples to optimize an existing network of water quality assessment could be considered as a combinatorial optimization problem. The methods for solving such problems can be roughly divided into two groups: the exact methods and approximate ones [11]. For solving NP-hard problems that are difficult to approach, metaheuristic algorithms are used [12,13]. However, these algorithms do not guarantee that an optimal solution would be found. Still they are widely used in practice as they allow users to obtain good solutions in reasonable time. Genetic algorithms also tend to be quite popular and are used for optimizing monitoring networks [14,15].

The present research aims to evaluate the effectiveness and robustness of genetic algorithms and compare them to variable neighborhood search family algorithms for solving problem of water quality assessment. We aimed to create an approach that would reduce the number of tested water-quality sites. One of the key feature of the proposed approach is that it allows user to find optimal sampling set of points capturing (i) the locations with the highest pollution rate and (ii) the locations representing the average regional level of pollution. To accomplish these goals, metaheuristic approaches were tested for optimization and multi-factors assessment of water quality monitoring in the New Moscow area. Namely, the dataset including 1182 samples from wells was analyzed. Our approach included five steps: selecting environmental factors, dividing the sampling grid, setting the initial stations, optimizing the sampling stations, and assessing reproducibility and efficiency of the proposed network.

2. Materials and Methods

2.1. Dataset for Case Study

Natural groundwater resources, such as springs, wells, and small rivers, are primary sources of fresh water in many countries, including Russia. They usually reflect human-made pollution sources, and environmental changes in soils and air [9,16]. To adjust the proposed approach, we used the dataset obtained in the course of the study by large-scale monitoring of small water bodies in New Moscow (see Figure 1), which is now part of Moscow, Russia.

The New Moscow region was chosen to be the case study because of its unusually great abundance of monitoring sites. We should mention that not all researches can boast such a wide and various range of samples, which should definitely give rise to the need for methods that evaluate the urbanization trend (i.e., the overall change in concentration across the network of monitoring sites) in the absence of long-term monitoring sites.

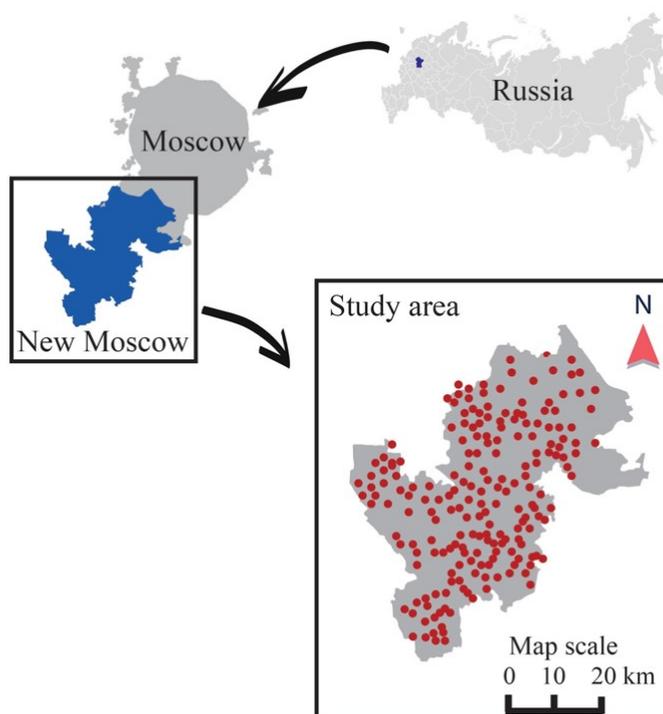


Figure 1. Study area: territory of New Moscow. Red points correspond to the water sampling locations considered in the research. The dataset consists of 1182 water sampling points allocated in wells. For each point 25 parameters of chemical properties are available.

This dataset under discussion includes three types of water sources: wells (1215 samples), springs (225 samples), small rivers (160 samples) monitored in 2017–2018—1600 samples in total [17]. Wells were selected to be an object for the research since they comprise the largest part of the sample set, 1182 samples remained on the list after outliers have been excluded. The outliers were discarded based on DBSCAN clustering algorithm (this methodology was described in detail in [18]). Each sample was evaluated according to 25 parameters of chemical properties. In order to use the model, we scaled all coordinates from 0 to 10. The Universal Transverse Mercator coordinate system [19] was used to ensure a correct conversion of geographical coordinates. The conversion was carried out using the *utm* library for python [20].

2.2. Integral Water Quality Index

Water quality assessment is usually based on a large number of parameters, i.e., physical and chemical characteristics (25 items in our case). One of the challenging task was to integrate the measured parameters within a single meaningful composite value (Water Quality Index—WQI). WQI was supposed to be a function of the measured parameters and represent the water quality in the particular sampling point. Principal component analysis (PCA) has already been proposed as a means to aggregate all measurements into the final WQI without overestimating parameters [21,22]. In this study, we used the WQI geospatial modeling approach proposed by Shadrin et al. [18]. This approach includes two steps: (i) calculating PCA-weighted WQI for each tested sample, (ii) applying Gaussian Process Regression (GPR) with optimal kernel structure search using Bayesian informational criteria in order to build a precise map of water quality distribution. It was successfully tested on a similar dataset and helped reveal the actual situation with water pollution.

A parameter selection work-flow for the further WQI construction, in brief, was as follows. According to previous study, from the whole data-set twenty-one parameters presented in water samples in significant concentrations were included in the PCA. From the output only components which corresponding eigenvalue was higher than or equal

to 1 following *Varimax* rotation, and PCs that explained at least 5% of the observed data variation were considered for further calculations. Those parameters that were correlated with other significant parameters (correlation was more than 0.6) were eliminated if they had the smallest loadings among correlated parameters. After the above-mentioned data analysis procedures, only 12 parameters were left for constructing the water quality index.

Finally, water quality index was calculated using the following equation taken from the work [18]:

$$\begin{aligned} \text{WQI} = & 0.2912 \cdot (\text{Cl}) + 0.0979 \cdot (\text{pH} + \text{Alkalinity}) + \\ & + 0.0884 \cdot (\text{NH}_4 + \text{PO}_4) + 0.0735 \cdot (\text{Cr} + \text{Fe} + \text{Mn}) + \\ & + 0.0589 \cdot (\text{Cu} + \text{SO}_4 + \text{K} + \text{NO}_3) \end{aligned} \quad (1)$$

The values of WQI were also scaled from 0 to 1. The general workflow of our research is presented on Figure 2.

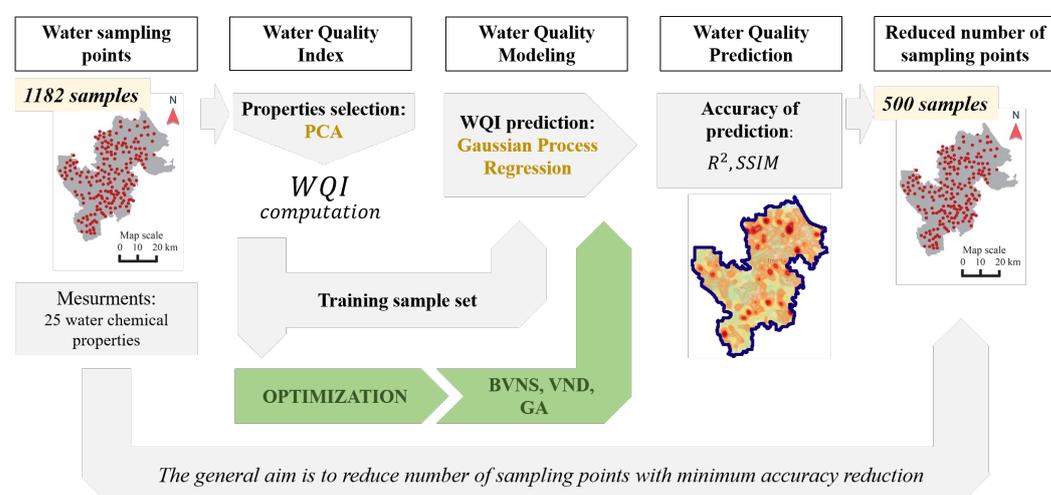


Figure 2. Outlines of the research.

2.3. Optimization Algorithms

The optimization criteria for monitoring networks has certain limits. In the algorithms, the main priority was given to those sampling points which have the most diverse neighbors in terms of WQI values and which represented the regional average values. All algorithms are implemented in the Python programming language.

2.3.1. Variable Neighborhood Search Algorithms

Two algorithms from the Variable Neighborhood Search (VNS) family were applied to the following cases: Variable Neighborhood Descent (VND) and basic VNS proposed by Hansen and Mladenovic in 2003 [23] and 2007 [24], respectively. The main idea of these methods is to change the neighborhoods and apply the local search procedure. Since the performance of VNS depends significantly on the choice of a neighborhood at each iteration [25], we used the neighborhood structure based on geographical similarity and k-mean clustering. The initial solution was randomly selected. It had to contain at least one point from each cluster and, at the same time, to exclude points from the similar area; as a result, the solution divided all points into clusters according to their coordinate.

One of the most popular algorithms of the VNS family is Variable Neighborhood Descent (VND). In order to apply this algorithm for solving the problem, system of neighborhoods and the initial solution had to be specified. In this algorithm the change of neighborhood was applied in a deterministic way. The proceeding search was performed using the selected neighborhood that had been obtained using the best improvement, which tended to be the most effective strategy. If an improved solution was obtained, all neighborhood structures on the list would become available for the next iteration. The

procedure ended when every enlisted structure had been exhaustively explored, and no other improvement could be achieved. The final solution is a local minimum with respect to all neighborhoods.

Algorithm Basic Variable Neighborhood Search (BVNS) is based on VND with an additional Shake function [26]. The idea of the Shake function is to generate a new solution that will not be close that would significantly differ from the current solution, which allows the optimizer to get away from the local optima.

VNS and BVNS implemented as randomized versions of these algorithms by searching only in part of the neighbor points. In each experiment the algorithms explore 10% of neighbors.

For each run of the VND algorithm, the change of neighborhoods is specified by a random sequence of numbers; as for the BVNS algorithm, the order is defined both at the beginning and after each shake. Randomization of algorithms helps to avoid the local optima and reduce the running time of algorithms, and at the same time increases the variance of solutions.

2.3.2. Genetic Algorithm

Genetic algorithm (GA), an evolutionary algorithm, was inspired by processes of biological evolution observed in the nature, such as crossover, mutation, and selection [27,28]. GA has been widely used for generating high-quality solutions to solve optimization problems. The procedure of GA has four parts: generating initial population, crossover operation, mutation operation, and selection (see Figure 3). A typical genetic algorithm requires a genetic representation of the solution domain and a fitness function to evaluate it.

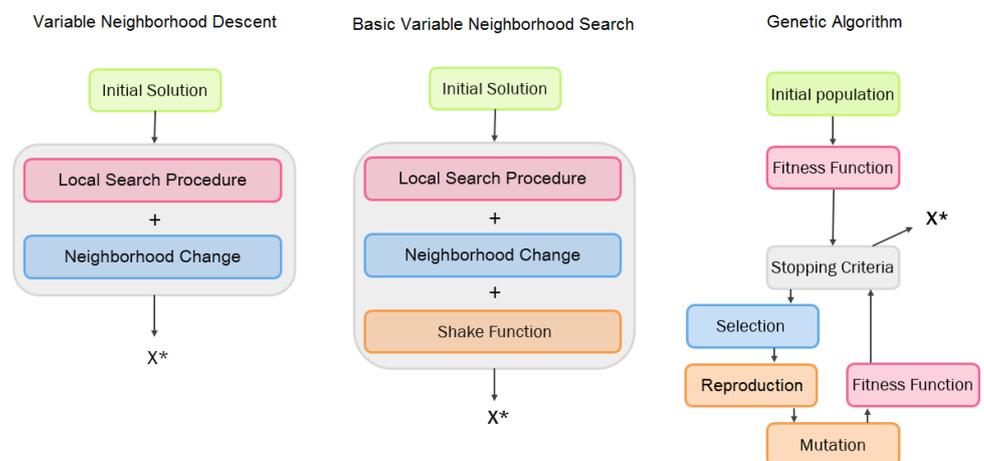


Figure 3. Schematic representation of the algorithms workflow. Variable Neighborhood Descent (VND) systematically changes the neighborhood in two phases: descent to find a local optimum and perturbation phase to get out of the corresponding valley. Basic Variable Neighborhood Search (BVNS) is a version of VND adjusted with the shake function. Genetic algorithm relies on biologically inspired operators such as mutation, crossover and selection. X^* defines resulting solution.

In the case with genetic algorithm the fitness function is considered to be an objective function. It is calculated in the same way as in VND and BVNS by maximizing R^2 score (all details about R^2 score and accuracy evaluation are discussed in Section 2.6. The solution to the optimization problem is a binary vector of length N , where N is the whole training data size. In this vector, ones correspond to the points included in the new training sample set and zeros correspond to the remaining points. The initial population of solutions is chosen randomly in the same way as the start solution in VND and BVNS algorithms. The size of

the initial population is regarded as an algorithm parameter. The fitness function decides which of the current population solutions should be included in the next population. In this case, the probability of choice depends on the place in the list of solutions sorted by the value of the fitness function.

Number N of the solutions to be selected in the following population is an algorithm parameter. Two or more parents are required for reproduction in genetic algorithms. At that, the offspring inherits features from both parents. In the algorithm, we used the strategy when both parents are randomly selected; thus, each individual in the population has an equal chance of being selected.

2.4. Baseline

Some preliminary actions are to be taken before the algorithms get compared; namely, it should be investigated whether they consider the problem in general terms or deal with it in greater detail. Obviously, the solutions obtained using optimization algorithms would outperform random sampling. Thus, we propose using the values of metrics for randomly selected solutions as a baseline.

2.5. Water Quality Prediction

To reconstruct spatial distribution of WQL, Gaussian Process Regression (GPR) was used [29]. Gaussian process is determined by its mean $\mu(\cdot)$ and covariance (kernel) $k(\cdot, \cdot)$ functions:

$$\begin{aligned} f(x) &\sim \mathcal{GP}(\mu(x), k(x, x')) \\ \mu(x) &= \mathbb{E} f(x) \\ k(x, x') &= \text{cov}(f(x), f(x')) \end{aligned} \quad (2)$$

where $x \in \mathbb{R}^d$ is a vector of input parameters. In this case, x is a vector of coordinates, so $d = 2$.

To solve the problem of reconstruction, the combination of the basic kernels was used [29,30]. Kernel hyper-parameter was obtained by using Bayesian Information Criteria (BIC) [31] according to [18], the selected kernels are represented by Equations (3) and (4) (see Table 1 for the obtained coefficients). The regression model for predicting the water quality index was implemented by using the *GPpy* library for python [32].

Radial Basis Function (RBF)

$$k_{RBF}(x, x') = \exp\left(-\frac{(x - x')^2}{2\ell_{RBF}^2}\right) \quad (3)$$

Periodic kernels (PE)

$$k_{PE}(x, x') = \exp\left(-\frac{2 \sin^2(\pi\omega(x - x'))}{\ell_{PE}^2}\right) \quad (4)$$

Table 1. Optimal kernel parameters for Gaussian Process Regression obtained by using Bayesian Information Criteria [31].

Parameter	Value
RBF variance	0.0367
ℓ_{RBF}	4.86
PE variance	0.0204
ω_{PE}	5.67
ℓ_{PE}	0.1

Usually, all data are divided into two parts; one is used for training and the other for testing, in order to measure the quality of model predictions. Such an approach required us to consider different subsets of the training sample set. The training and testing data were evenly spread across the area (Figure 4a): the size of test data was 119 sample (10% of the whole dataset) (Figure 4c) and that of training data was 1063 samples (90% of the total set) (Figure 4b).

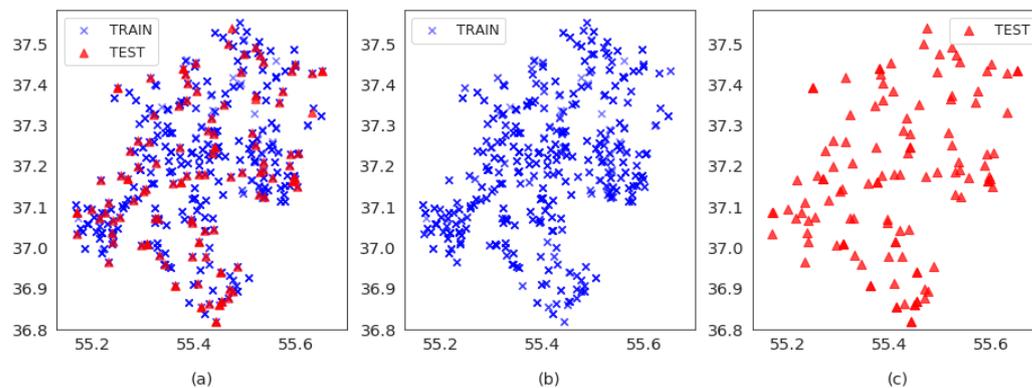


Figure 4. (a) Training and test sample sets. (b) Training part is 90% of the whole sample set, (c) test part—10%. Samples evenly cover the whole area of interest.

2.6. Optimization Problem Formulation and Accuracy Evaluation

To estimate the performance of the above-described algorithms, two statistical indices were used. The former is the coefficient of determination denoted R^2 , which shows how well the observed outcomes would be replicated by the model, based on the proportion of total variation of the outcomes explained by the model [33].

In terms of optimization theory, the considered problem can be written as follows.

Suppose we are given a set of points $X_{train} \in \mathbb{R}^{n \times d}$ and $X_{test} \in \mathbb{R}^{k \times d}$ with corresponding target vectors $y_{train} \in \mathbb{R}^n$ and $y_{test} \in \mathbb{R}^k$ and regression model $f(\cdot)$. We need to find a subset $X' \subset X_{train}$ which will maximize the following objective function on the test set X_{test} :

$$R^2score_{f(X')}(y_{pred}, y_{test}) \rightarrow \max_{X' \subset X_{train}} \tag{5}$$

$$R^2(y_{pred}, y_{test}) = 1 - \frac{\sum_{i=1}^k (y_{pred_i} - y_{test_i})^2}{\sum_{i=1}^k (y_{test_i} - \overline{y_{test}})^2}, \tag{6}$$

where y_{pred} is prediction of $f(X')$ on X_{test} and R^2score is a metric for measuring model prediction quality. The best possible value of R^2 score is 1.0. The goal is to find subset of training sample set which maximizes R^2 score on the test set, so summation only for points in test set

The second statistical metric is the structural similarity index measure, denoted SSIM. This index is usually used for comparing two images. Since in our study we display the predicted water quality index on a map, we can treat the predictions of the model trained on different data (subsets of different sizes from training set) as pictures to compare them.

The structural similarity index is calculated for various windows of an image [34]. For two windows x and y of typical size $N \times N$:

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}, \tag{7}$$

where μ_x is mean x , μ_y —mean y , σ_x^2 —variance x , σ_y^2 —variance y , σ_{xy} —covariance, $c_1 = (k_1 L)^2$ and $c_2 = (k_2 L)^2$, L —dynamic range of pixels $2^{(\text{bits per pixel})} - 1$, $k_1 = 0.01$ and $k_2 = 0.03$. For identical images the SSIM value is equal to 1.0.

SSIM score is significantly more computationally expensive than R^2 score; thus, we did not use it as the objective function for the considered optimization problems. Still, we calculated the SSIM index for the best solutions that were found.

3. Results and Discussion

Figure 5 shows the result of comparing all algorithms. We plotted R^2 score (a, b, c) and SSIM score (d, e, f) against the training sample size. Red lines on the graph correspond to the algorithms, while blue ones show the baseline, which contained 10 random solutions. The black line stands for the score on full training data.

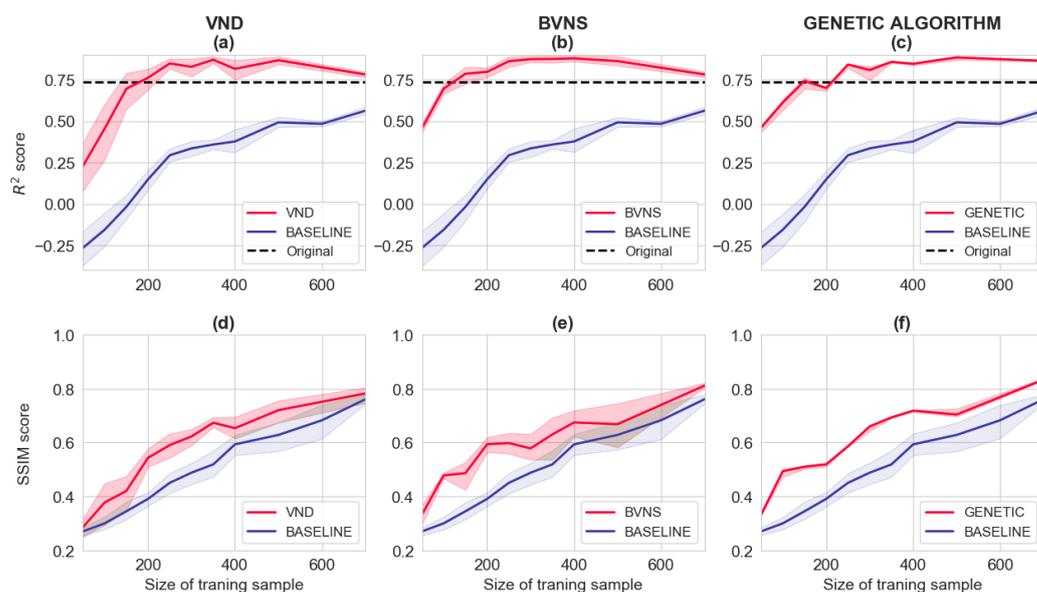


Figure 5. The performance of algorithms compared by R^2 score (a–c) and $SSIM$ score (d–f) against the size of the training sample. Red lines correspond to the algorithms, and blue lines stand for the baseline. The black line shows the score on full training data. All the considered algorithms outperforms the baseline both by R^2 and $SSIM$, BVNS shows the best results.

We obtained 10 solutions by each algorithm using the same value of the parameter and plotted the confidence intervals. The baseline contains 10 random solutions. All the considered algorithms outperformed the baseline both by R^2 and $SSIM$; BVNS showed the best results

Table 2 shows the best and average values of statistical characteristics for the considered algorithms obtained using the parameter search. The line “best perform n samples” in Table 2 is the size of the training sample set. It is to be noted that that the problem under discussion and the implementation of BVNS algorithm appeared to get better solutions than others algorithms and had a smaller variance of values of the objective function. Moreover, the quality of prediction based on selected samples was higher than the quality of the prediction based on a full number of samples.

Figure 6 presenting spatial visualization of WQI prediction for water quality index predictions made using Gaussian Process Regression. In our case the sizes of training samples sets were 400, 500 and 700 sampling points. Noteworthy, predictions on 500 samples showed better results than predictions on 700 samples. The best performance was obtained by VND on 500 samples. We assume that the model finds specific patterns in the training data that described dependencies on the site better than the remaining data. This hypothesis requires further research in terms of multi-objective optimization.

Table 2. Best and average values of statistical characteristics for the considered algorithms obtained by the parameter search. BVNS shows the best accuracy for R^2 and $SSIM$ and has the smallest variance for R^2 . All the algorithms show their best performance on 500 training samples. (see Appendix A for the final set of best parameter values.)

Characteristics	VND	BVNS	GA	All Samples
Best perform n samples	500	500	500	1063
Best perform R^2	0.905	0.905	0.889	0.737
Best perform $SSIM$	0.681	0.748	0.698	
Average variance R^2	0.0122	0.0011	0.0012	
Average variance $SSIM$	0.0032	0.0032	0.0002	

The results show that all 3 algorithms managed to cope with the problem quite successfully. These algorithms helped to improve the quality of the prediction model from 0.73 to 0.9–0.93 of R^2 score (Table 2). Hence, the model can be claimed to be effective in finding in the training sample set some random patterns that have not described the data in general [35]. Considering the structural similarity index, all algorithms also appeared to cope with a specified baseline. Moreover, the values of $SSIM$ over 0.6 are considered as quite good for such an environmental problem.

All the abovementioned results were obtained by certain splitting data into train and test. The issue to be focused upon in the course of the study was whether the splitting data affect the solution of the optimization problem and what results we may face if all significant points would fall into the test sample. Normally, to estimate accuracy of the predictive model accuracy, the cross-validation procedure is used Table A1. We used k-fold cross-validation (with $k = 5$, size of training part 80%, size of test part 20%). To estimate the values of the objective function on these splits, it is necessary to find a solution of the optimization problem

Based on the results of the greedy strategy, we would hypothesize that among the points, a certain subset of points mainly contributes to the quality of model predictions (e.g., this was the reason for a sharp jump of the R^2 score in the value of size if the training set is 242 points, see Figure 5). Having obtained this set, we observed no dramatic improvements in the objective function. When a lot of points were included in the training sample (over 800), the model found the patterns in the training data that do not describe the full data. The proposed hypothesis requires further investigation in terms of multi-objective optimization.

The dataset cross-validation process under discussion was found to affect insignificantly the search for the optimal subset of points if the points are uniformly distributed across the study area. Running time appeared to be the main factor that hampers the analysis of the implemented algorithms. Each calculation of the objective function required the regression model to be trained and tested. Since the algorithms are randomized to study the parameter space (as well as to search for the most suitable ones), the algorithms have to be run several times to obtain a sample of solutions for each parameter. Though the sample of 10 solutions does not seem large enough and sufficient, even such a sample does demonstrate the performance of the proposed algorithms.

The resultant approach can be further used to create effective systems for monitoring water quality in other geographical areas. To do so, firstly, an excessive number of measurements in a new area should be carried out; then, using the proposed optimization algorithms, the main points have to be determined at which water monitoring is required. At that, no modifying of the implemented algorithms are needed; users can only give the algorithms new data and get the results.

Moreover, the considered algorithms can be used in other monitoring problems, at which geographical coordinates of the points to be monitored and values of the necessary characteristics should be provided. Thus, to implement these algorithms, the prediction

model has to be changed. All implemented algorithms accept the prediction model as an input parameter.

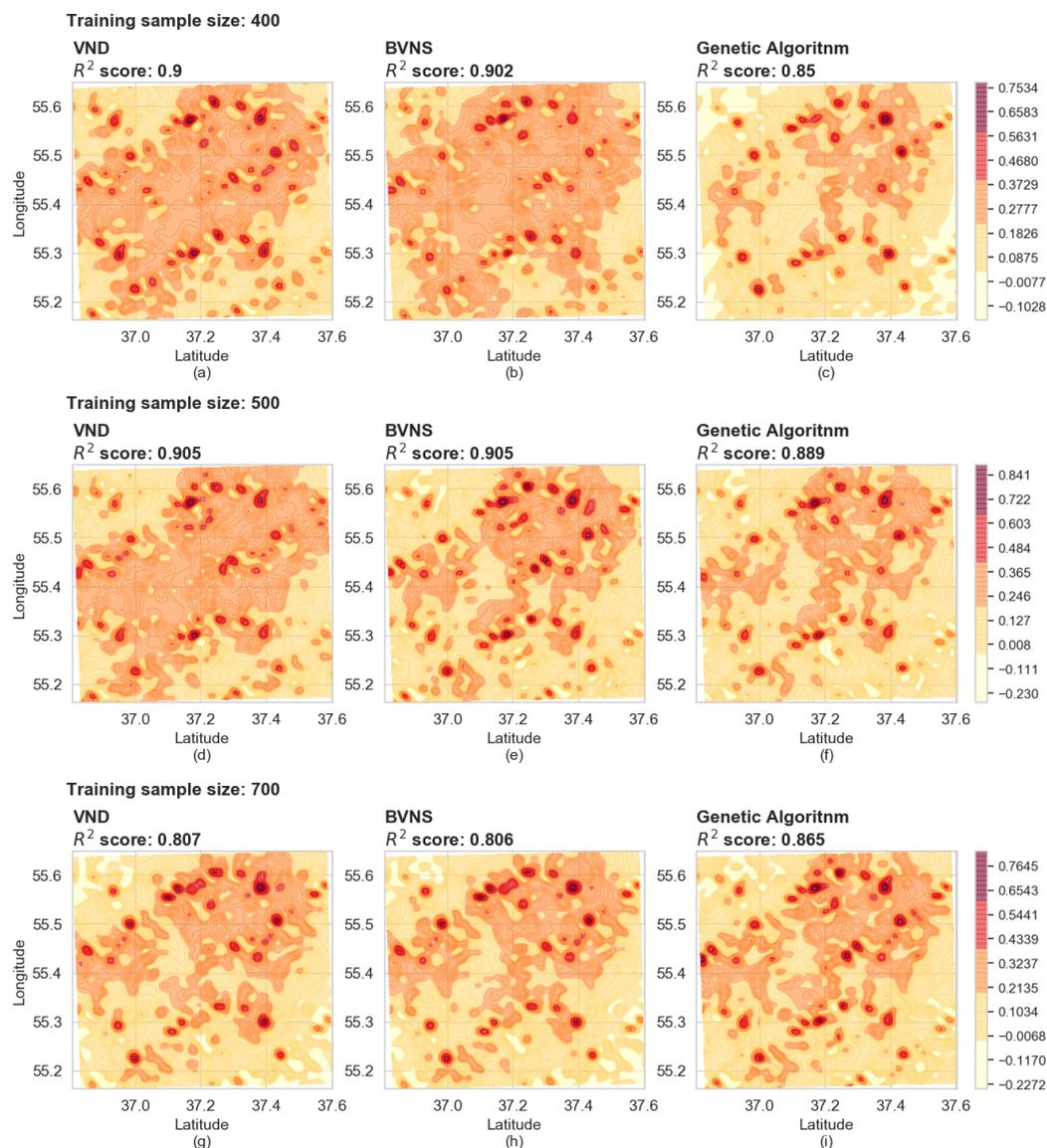


Figure 6. Spatial visualization of water quality index predictions made using Gaussian Process Regression. Training samples sets has sizes 400 (a–c), 500 (d–f) and 700 (g–i) sampling points. It can be noticed that predictions on 500 samples show better results than epy predictions on 700 samples. The best performance is obtained by VND on 500 samples.

4. Conclusions

The considered algorithms (variable neighborhood descent, basic variable neighborhood search, genetic algorithm) make it possible to reduce the size of the training sample set. For instance, the number of points for monitoring water quality in the New Moscow area can be reduced by 500, which two times decreases the total cost of the analysis. The quality of prediction has initially reached 93%, which was later improved by 18% after the size of the training sample had been reduced. This may possibly be due to the model managed to detect some specific random patterns not describing the data in general. All the algorithms showed good results; the BVNS algorithm performed significantly better than others as regarded to the value of the objective function and its variance. Thus, it seems reasonable enough to use a combination of the algorithms to get better solutions. In

addition, the implemented algorithms under investigation have been proved to be suitable for the optimization problem under consideration in similar monitoring cases.

The presented study provides a good starting point for discussion and further research. Future studies could investigate precisely the reasons why the fewer number of sampling points gives a better WQI prediction. Furthermore, an interesting topic for future work is considering the problem of water quality assessment as a multi-objective optimization problem. For example, R^2 score could be maximized simultaneously with minimization of sampling points number. In this case, a subset of sampling points could be considered not as a parameter of algorithm but as an input variable.

Author Contributions: Conceptualization, E.Y., D.S. and M.P.; Data curation, E.Y.; Formal analysis, E.Y., A.P. and D.S.; Funding acquisition, M.P.; Investigation, E.Y., A.P. and D.S.; Methodology, E.Y. and D.S.; Project administration, M.P. and I.O.; Resources, E.Y. and E.C.; Software, E.Y. and D.S.; Supervision, M.P. and I.O.; Validation, E.Y.; Visualization, A.P.; Writing—original draft, E.Y., A.P. and P.T.; Writing—review & editing, A.P., D.S., P.T. and E.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Russian Science Foundation (project No. 20-74-10102).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data that support the findings of this study are openly available at <https://doi.org/10.6084/m9.figshare.10283225.v4>.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

BVNS	Basic Variable Neighborhood Search
GA	Genetic Algorithm
GPR	Gaussian Process Regression
PCA	Principal Component Analysis
SSIM	Structural Similarity Index Measure
VNS	Variable Neighborhood Search
VND	Variable Neighborhood Descent
WQI	Water Quality Index

Appendix A

Table A1. Application of the genetic algorithm. Cross-validation.

Number of Points	Metric	Calculation Experiments for Different Train/Test Split						
		1	2	3	4	5	Mean	Std
100	R^2	0.643	0.460	0.594	0.566	0.587	0.570	0.060
	SSIM	0.296	0.293	0.382	0.414	0.273	0.332	0.056
	Time (h)	1.224	1.240	1.027	1.406	1.139	1.207	0.125
200	R^2	0.695	0.611	0.675	0.586	0.713	0.656	0.049
	SSIM	0.408	0.332	0.432	0.500	0.408	0.416	0.054
	Time (h)	2.491	2.547	2.546	2.415	2.339	2.468	0.081

Table A1. Cont.

Number of Points	Calculation Experiments for Different Train/Test Split							
	Metric	1	2	3	4	5	Mean	Std
300	R^2	0.719	0.709	0.741	0.638	0.717	0.705	0.035
	SSIM	0.474	0.642	0.562	0.540	0.473	0.538	0.063
	Time (h)	6.384	6.004	6.486	6.932	7.591	6.679	0.543
400	R^2	0.724	0.702	0.751	0.670	0.754	0.720	0.032
	SSIM	0.558	0.641	0.692	0.658	0.449	0.600	0.087
	Time (h)	14.70	14.23	16.29	15.75	14.82	15.16	0.752
959	R^2	0.638	0.469	0.574	0.544	0.586	0.562	0.055
	SSIM (t)	0.938	0.920	0.908	0.932	0.941	0.928	0.012

Appendix A.1. Variable Neighborhood Descent

Algorithm A1 VND

INPUT a set of neighborhoods $N_k, k = 1, \dots, k_{max}$, initial solution $x := x_0$

- 1: **while** $k \leq k_{max}$ **do**
- 2: Find the best neighbor x' of x in $N_k(x)$
- 3: **if** $f(x') < f(x)$ **then**
- 4: $x := x', k := 1$
- 5: **else**
- 6: $k := k + 1$

OUTPUT Best solution found

Hyperparameters used in the article

Observed part of neighborhood: 10%

Stopping criteria: lack of improvements in all neighborhoods or OF ≥ 0.9

Number of clusters: input parameter

Size of training sample set: input parameter

Appendix A.2. Basic Variable Neighborhood Search

Algorithm A2 Basic VNS

INPUT a set of neighborhoods $N_k, k = 1, \dots, k_{max}$, initial solution $x := x_0$

- 1: **repeat until** stopping criteria
- 2: $k := 1$
- 3: **while** $k \leq k_{max}$ **do**
- 4: $x' := \text{Shake}()$
- 5: Find the best neighbor x'' of x' in $N_k(x)$ (local search)
- 6: **if** $f(x'') < f(x)$ **then**
- 7: $x := x'', k := 1$
- 8: **else**
- 9: $k := k + 1$

OUTPUT Best solution found

Hyperparameters used in the article

Observed part of neighborhood: 10%
Number of shaking: 5
Stopping criteria: lack of improvements in all neighborhoods or $OF \geq 0.9$
Number of clusters: input parameter
Size of training sample set: input parameter

Appendix A.3. Genetic Algorithm

Algorithm A3 Genetic Algorithm

INPUT Initial population

- 1: **repeat until** stopping criteria
- 2: Reproduction
- 3: Mutation
- 4: Calculate the value of the Fitness Function for all individuals (solutions)
- 5: New population formation (Selection)

OUTPUT Resulting population

Hyperparameters used in the article

Population size: 30
Number of populations: 1000
Number of mutations: 10%
Number of survivors individuals: 40%
Number of reproduced individuals: 60%
Stopping criteria: number of populations or $OF \geq 0.9$
Number of clusters: input parameter
Size of training sample set: input parameter

References

1. Howden, N.; Mather, J. *History of Hydrogeology*; CRC Press: Boca Raton, FL, USA, 2012.
2. Tavakol, M.; Arjmandi, R.; Shayeghi, M.; Monavari, S.M.; Karbassi, A. Application of multivariate statistical methods to optimize water quality monitoring network with emphasis on the pollution caused by fish farms. *Iran. J. Public Health* **2017**, *46*, 83.
3. Alilou, H.; Nia, A.M.; Keshtkar, H.; Han, D.; Bray, M. A cost-effective and efficient framework to determine water quality monitoring network locations. *Sci. Total Environ.* **2018**, *624*, 283–293. [[CrossRef](#)]
4. Zhu, X.; Yue, Y.; Wong, P.W.; Zhang, Y.; Ding, H. Designing an Optimized Water Quality Monitoring Network with Reserved Monitoring Locations. *Water* **2019**, *11*, 713. [[CrossRef](#)]
5. Lee, S.; Kim, J.; Hwang, J.; Lee, E.; Lee, K.J.; Oh, J.; Park, J.; Heo, T.Y. Clustering of Time Series Water Quality Data Using Dynamic Time Warping: A Case Study from the Bukhan River Water Quality Monitoring Network. *Water* **2020**, *12*, 2411. [[CrossRef](#)]
6. Dixon, W.; Smyth, G.K.; Chiswell, B. Optimized selection of river sampling sites. *Water Res.* **1999**, *33*, 971–978. [[CrossRef](#)]
7. Nguyen, T.H.; Helm, B.; Hettiarachchi, H.; Caucci, S.; Krebs, P. Quantifying the Information Content of a Water Quality Monitoring Network Using Principal Component Analysis: A Case Study of the Freiburger Mulde River Basin, Germany. *Water* **2020**, *12*, 420. [[CrossRef](#)]
8. Jiang, J.; Tang, S.; Han, D.; Fu, G.; Solomatine, D.; Zheng, Y. A comprehensive review on the design and optimization of surface water quality monitoring networks. *Environ. Model. Softw.* **2020**, *132*, 104792. [[CrossRef](#)]
9. Biggs, J.; Von Fumetti, S.; Kelly-Quinn, M. The importance of small waterbodies for biodiversity and ecosystem services: implications for policy makers. *Hydrobiologia* **2017**, *793*, 3–39. [[CrossRef](#)]
10. Mooselu, M.G.; Liltved, H.; Nikoo, M.R.; Hindar, A.; Meland, S. Assessing optimal water quality monitoring network in road construction using integrated information-theoretic techniques. *J. Hydrol.* **2020**, *589*, 125366. [[CrossRef](#)]
11. Puchinger, J.; Raidl, G. Combining Metaheuristics and Exact Algorithms in Combinatorial Optimization: A Survey and Classification. In Proceedings of the International Work-Conference on the Interplay between Natural and Artificial Computation, La Palma, Spain, 15–18 June 2005; Volume 3562, pp. 41–53. [[CrossRef](#)]
12. Boschetti, M.A.; Maniezzo, V.; Roffilli, M.; Bolufé Röhler, A. Matheuristics: Optimization, Simulation and Control. In *Hybrid Metaheuristics*; Blesa, M.J., Blum, C., Di Gaspero, L., Roli, A., Sampels, M., Schaerf, A., Eds.; Springer: Berlin/Heidelberg, Germany, 2009; pp. 171–177.
13. Cormen, T.H.; Leiserson, C.E.; Rivest, R.L.; Stein, C. *Introduction to Algorithms*, 2nd ed.; MIT Press: Cambridge, MA, USA, 2001.

14. Puri, D.; Borel, K.; Vance, C.; Karthikeyan, R. Optimization of a water quality monitoring network using a spatially referenced water quality model and a genetic algorithm. *Water* **2017**, *9*, 704. [CrossRef]
15. Park, S.Y.; Choi, J.H.; Wang, S.; Park, S.S. Design of a water quality monitoring network in a large river system using the genetic algorithm. *Ecol. Model.* **2006**, *199*, 289–297. [CrossRef]
16. Weldeslassie, T.; Naz, H.; Singh, B.; Oves, M. Chemical Contaminants for Soil, Air and Aquatic Ecosystem. In *Modern Age Environmental Problems and Their Remediation*; Oves, M., Zain Khan, M., Ismail, I., Eds.; Springer International Publishing: Cham, Switzerland, 2018; pp. 1–22. [CrossRef]
17. Pukalchik, M.; Shadrin, D.; Nikitin, A.; Jana, R.; Tregubova, P.; Matveev, S. Freshwater Chemical Properties for New Moscow Region. 2019. Available online: https://figshare.com/articles/dataset/freshwater_chemical_properties_for_New_Moscow_region/10283225/4 (accessed on 21 March 2021).
18. Shadrin, D.; Nikitin, A.; Tregubova, P.; Terekhova, V.; Jana, R.; Matveev, S.; Pukalchik, M. An Automated Approach to Groundwater Quality Monitoring—Geospatial Mapping Based on Combined Application of Gaussian Process Regression and Bayesian Information Criterion. *Water* **2021**, *13*, 400. [CrossRef]
19. Snyder, J. *Map Projections: A Working Manual*; Professional Paper; United States Geological Survey, U.S. Government Printing Office: Washington, DC, USA, 1994.
20. Bieniek, T. utm: Bidirectional UTM-WGS84 Converter for Python. 2012. Available online: <https://github.com/Turbo87/utm> (accessed on 17 February 2021).
21. Boyacioglu, H. Development of a water quality index based on a European classification scheme. *Water SA* **2007**, *33*, doi:10.4314/wsa.v33i1.47882. [CrossRef]
22. Tripathi, M.; Singal, S.K. Use of Principal Component Analysis for parameter selection for development of a novel Water Quality Index: A case study of river Ganga India. *Ecol. Indic.* **2019**, *96*, 430–436. [CrossRef]
23. Duarte, A.; Sánchez-Oro, J.; Mladenović, N.; Todosijević, R. Variable Neighborhood Descent. In *Handbook of Heuristics*; Martí, R., Pardalos, P.M., Resende, M.G.C., Eds.; Springer International Publishing: Cham, Switzerland, 2018; pp. 341–367. [CrossRef]
24. Hansen, P.; Mladenovic, N. Variable Neighborhood Search Methods. In *Encyclopedia of Optimization*; Floudas, C., Pardalos, P., Eds.; Springer: Berlin/Heidelberg, Germany, 2008. [CrossRef]
25. Harzi, M.; Krichen, S. Variable neighborhood descent for solving the vehicle routing problem with time windows. *Electron. Notes Discret. Math.* **2017**, *58*, 175–182. [CrossRef]
26. Hansen, P.; Mladenović, N. Variable neighborhood search. In *Search Methodologies*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 313–337.
27. Holland, J.H. Genetic algorithms and adaptation. In *Adaptive Control of Ill-Defined Systems*; Springer: Berlin/Heidelberg, Germany, 1984; pp. 317–333.
28. Li, X.; Parrott, L. An improved Genetic Algorithm for spatial optimization of multi-objective and multi-site land use allocation. *Comput. Environ. Urban Syst.* **2016**, *59*, 184–194. [CrossRef]
29. Rasmussen, C.E.; Williams, C.K.I. *Gaussian Processes for Machine Learning*; The MIT Press: Cambridge, MA, USA, 2006.
30. Wilson, A.; Adams, R. Gaussian Process Kernels for Pattern Discovery and Extrapolation. In *Machine Learning Research, Proceedings of the 30th International Conference on Machine Learning, Atlanta, GA, USA, 16–21 June 2013*; Dasgupta, S., McAllester, D., Eds.; PMLR: Atlanta, GA, USA, 2013; Volume 28, pp. 1067–1075.
31. Duvenaud, D.; Lloyd, J.R.; Grosse, R.; Tenenbaum, J.B.; Ghahramani, Z. Structure discovery in nonparametric regression through compositional kernel search. *arXiv* **2013**, arXiv:1302.4922.
32. GPy: A Gaussian Process Framework in Python. 2012. Available online: <https://github.com/SheffieldML/GPy> (accessed on 17 February 2021).
33. Carpenter, R. Principles and procedures of statistics, with special reference to the biological sciences. *Eugen. Rev.* **1960**, *52*, 172.
34. Wang, Z.; Simoncelli, E.P.; Bovik, A.C. Multiscale structural similarity for image quality assessment. In *Proceedings of the Thirty-Seventh Asilomar Conference on Signals, Systems Computers, Pacific Grove, CA, USA, 9–12 November 2003*; Volume 2, pp. 1398–1402.
35. Cawley, G.C. Over-Fitting in Model Selection and Its Avoidance. In *Advances in Intelligent Data Analysis XI*; Hollmén, J., Klawonn, F., Tucker, A., Eds.; Springer: Berlin/Heidelberg, Germany, 2012; p. 1.