

Resampling in neural networks with application to spatial analysis

Bruno Póvoa Rodrigues, Vinicius Francisco Rofatto, Marcelo Tomio Matsuoka & Talita Teles Assunção

To cite this article: Bruno Póvoa Rodrigues, Vinicius Francisco Rofatto, Marcelo Tomio Matsuoka & Talita Teles Assunção (2022) Resampling in neural networks with application to spatial analysis, *Geo-spatial Information Science*, 25:3, 413-424, DOI: [10.1080/10095020.2022.2040923](https://doi.org/10.1080/10095020.2022.2040923)

To link to this article: <https://doi.org/10.1080/10095020.2022.2040923>



© 2022 Wuhan University. Published by Informa UK Limited, trading as Taylor & Francis Group.



Published online: 03 Mar 2022.



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



Article views: 1134



View related articles [↗](#)



View Crossmark data [↗](#)



Citing articles: 1 View citing articles [↗](#)

Resampling in neural networks with application to spatial analysis

Bruno Póvoa Rodrigues ^a, Vinicius Francisco Rofatto ^{a,b}, Marcelo Tomio Matsuoka ^{a,b}
and Talita Teles Assunção ^b

^aGraduate Program in Agriculture and Geospatial Information, Federal University of Uberlândia, Monte Carmelo, Brazil; ^bInstitute of Geography, Federal University of Uberlândia, Monte Carmelo, Brazil

ABSTRACT

In developing Artificial Neural Networks (ANNs), the available dataset is split into three categories: training, validation and testing. However, an important problem arises: How to trust the prediction provided by a particular ANN? Due to the randomness related to the network itself (architecture, initialization and learning procedure), there is usually no best choice. Considering this issue, we provide a framework, which captures the randomness related to the network itself. The idea is to perform several training and test trials based on the Jackknife resampling method. Jackknife consists of iteratively deleting a single observation each time from the sample and recomputing the ANN on the rest of the sample data. Consequently, interval prediction is available instead of point prediction. The proposed method was applied and tested using pH, Ca and P data obtained by analyzing 118 georeferenced soil points. The results, based on the dataset size simulation, showed that 60% reduction in available dataset offers compatible accuracy in relation to full dataset, and therefore a higher cost of sampling in the field would not be necessary. The re-sampling method spatially characterizes the points of greater or lesser accuracy and uncertainty. The re-sampling method increased the success rate by using interval prediction instead of using the mean as the most probable value. Although we restrict it to the regression neural network model, the resampling method proposed can also be extended to other modern statistical tools, such as Kriging, Least Squares Collocation (LSC), Convolutional Neural Network (CNN), and so on.

ARTICLE HISTORY

Received 11 August 2021
Accepted 7 February 2022

KEYWORDS

Artificial Neural Network (ANN); data splitting; resampling; delete-1 Jackknife; spatial analysis

1. Introduction

One of the most important tasks in developing a neural network is related to partitioning the dataset. Often, most researchers randomly and uniformly split a known dataset into three categories: training, validation and testing. The training set is often used to estimate the unknown parameters of the net (e.g. weights and biases in a regression feed-forward neural network). The validation set helps one in making a decision on how to train the network and how to stop it in order to prevent oneself from overfitting. In machine learning languages, test sets are unseen data, which play an important role in the evaluation of the generalized performance of the network. This method of splitting data randomly is known as Hold-out.

This method in Artificial Neural Network (ANN) has been used in geo-spatial applications, especially when the data set is large, as can be seen here Aboufazel, Jahani, and Farahpour (2021), Jahani et al. (2021a), Jahani and Saffariha (2021b), and Aslam et al. (2021). There are, however, significant limitations of using this method, as pointed out by Ziggah et al. (2019) as follow: (i) the results produced is based on the uncontrolled chosen split sets; (ii) improper split of the data set could have an adverse effect on the model

performance; (iii) depends on having large dataset making it unsuitable to be applied in data-insufficient situations (i.e. sparse dataset).

To overcome these problems, the K-fold cross-validation has been recommended as an alternative method (Burman 1989; Reitermanová 2010). In this case, the data is separated into K disjoint subsets (K-fold) of approximately equal size, so that each subset will be in a test set just once. Although the K-fold present advantages, it is still not clear how to choose the K disjoint subsets and in some cases the subsets are not of equal sizes, which does not guarantee a balanced version of cross-validation. Furthermore, the two methods above provide only a single-point prediction. Therefore, splitting data methods still seems to be challenging. Thus, an important issue arises here: *how to split the data properly so that the prediction is as reliable as possible?*

Within the context of neural network modeling, the best choice is one in which the test set is as small as possible. In other words, if the training set is close to the full sample size, the more accurate the neural network model, as can be seen in the recent work of Balmer, Weibel, and Huang (2021). For this, one can use the Delete-1 Jackknife resampling method (Quenouille 1949). The Delete-1 Jackknife is a special case of the K-fold method by choosing K to be equal to

the size of the available dataset (sample), say n . In this approach, each observation is used to validate the performance of the neural network trained with the remaining $(n - 1)$ observations. In the context of neural networks, the Delete-1 Jackknife method is a balanced version of cross-validation.

However, the uncertainties related to the network itself (architecture, initialization and learning procedure) are still questionable. Therefore, there is usually no best choice (Pan 1998). In light of this issue, here we extend the Delete-1 Jackknife resampling method to capture the randomness related to the network itself. The basic idea consists of repeating the Delete-1 Jackknife process a large number of times. Hence, each test subset, which in this case is composed of only one single observation, predicts a desired number of times. Consequently, we obtain an interval prediction for each test subset, and not only a single prediction as in classical methods. In this way, we are able to describe the uncertainty of each predicted point, which was not possible using traditional Hold-Out and K-fold methods.

2. Extension of delete-1 Jackknife resampling method in neural networks

In general, the number of possible combinations for independent cross-validation, denoted by C_n^d , and the number of repetitions of each point prediction (R_n^d) are given respectively by

$$C_n^d = \binom{n}{d} = \frac{n!}{(n-d)!d!} \quad (1)$$

$$R_n^d = C_n^d \times \left(\frac{d}{n}\right) \quad (2)$$

where n is the size of the available sample (or the size of the dataset at hand), is the number of sample points, which are removed from the available sample for validation. This means that we validate the subset of size d and train the remaining $(n - d)$ observations at a time. The subset size d is selected from all of the observations without replacement. For instance, if $n = 10$ and $d = 2$, we would have $C_{10}^2 = 45$ independent cross-validation subsets, with each output predicted $R_{10}^2 = 9$ times. This method is known as Jackknife-d (Efron 1980, 1992; Wang and Yu 2020).

The Delete-1 Jackknife is a special case by taking $d = 1$. In this case, each observation is used to validate the performance of the neural network trained with the remaining $(n - 1)$ observations (Figure 1). In the context of neural networks, the Delete-1 Jackknife method is a balanced version of cross-validation. In machine learning languages, the Delete-1 Jackknife method is commonly referred to as the leave-one-out procedure (Efron 1982).

For instance, if $n = 10$, the application of the Delete-1 Jackknife method would provide 10 cross-validation, with each observation predicted only once, i.e. $C_{10}^1 = 10$ and $R_{10}^1 = 1$.

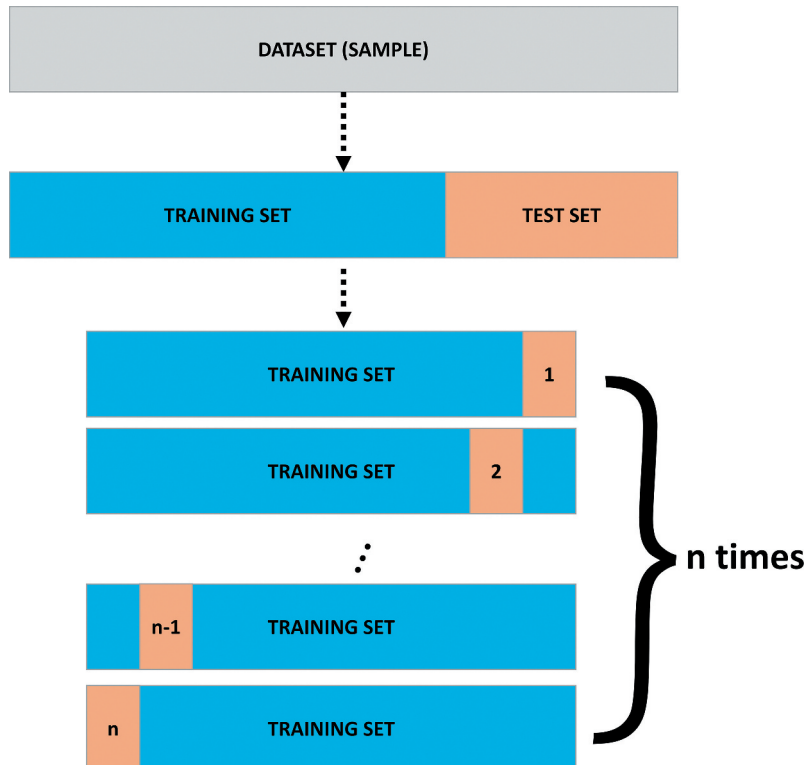


Figure 1. Delete-1 Jackknife resampling method.

Here, on the other hand, we extend the Delete-1 Jackknife method by drawing many trials in order to capture the randomness related to the network itself (architecture, initialization and learning procedure). We call this method Delete-1 Jackknife Trials (or simply, Jack-1 T). Therefore, instead of having only one single particular neural network predictor, we will have hundreds or even thousands of predictors. This means that instead of making a single-point prediction, we can describe its empirical distribution. In other words, this gives us the opportunity to do interval prediction instead of just one-point prediction.

The accuracy of the Jackknife re-sampling method depends on the choice of the number of groups that will be deleted from the neural network training process (i.e. d). In general, the more observations in the training set, the better the network in terms of learning, which justifies the idea of the Jack-1 T. Similar idea can be found in (Miller 1974; Pan 1998).

3. Outcomes from Jack-1 T

The most probable estimate (expected value) of an output quantity is not based on only one single prediction, but rather on an interval basis. For this, we sort the N predicted values of a point into strictly increasing order. The sorted predicted values provide an empirical distribution function for each output point (denoted by G). Then, for a stipulated coverage probability (denoted by α) we are able to compute the desired percentiles p as follows:

$$p = G_{[\lfloor (1-\alpha) \times N \rfloor]} \quad (3)$$

where $\lfloor \cdot \rfloor$ denotes rounding down to the next integer, which indicates the position of the selected elements in the ascending order of G . This position corresponds to a prediction for a stipulated overall probability α . This can be done for any α . Indeed, a confidence interval for the predictions may also be constructed. This procedure is very similar to that found for the computation of critical values of outlier statistical tests (Rofatto et al. 2020).

The re-sampling methods presented here also allow us to measure the prediction performance for the entire sample set. This is due to the fact that all sampling points are replicated in the test set. The Root Mean Squared Error (RMSE) on the test set can be used to measure the accuracy of the prediction of each point “ i ” as follows

$$RMSE_{(i)} = \sqrt{\frac{1}{N} \sum_{k=1}^N (T_k - \hat{O}_k)^2}, \quad i = 1, \dots, n \quad (4)$$

where \hat{O}_k are the N predicted values for each sample point i and T_k are the N true output values. The RMSE has been used as standard statistical parameters to

measure model performance in several applications. The RMSE indicates how far the output responses of the neural network are from the true outputs (targets) for each i . Note that RMSE is computed for each of the n available sample points. Although this expression represents the RMSEs for a single output, it can be applied to the case where one more output is available. Consequently, accuracy and uncertainty maps can also be provided.

To make it clearer, the procedure associated with Jack-1 T is summarized step-by-step as follows:

- (1) The dataset available of size n is divided into two parts: training and test sets.
- (2) Train the model with $(n - 1)$ observations and evaluate it using the remaining observations. For instance, given x as input and y as output variables in a neural network, we would have $\{(x_1, y_1), (x_2, y_2) \dots, (x_{k-1}, y_{k-1}), (x_{k+1}, y_{k+1}), \dots, (x_n, y_n)\}$ as training set, and the remaining observation (x_k, y_k) would be used to evaluate the model.
- (3) Repeat the procedure in (2) n times, excluding at each moment a different observation until all the observations have been deleted (Figure 1).
- (4) The procedures in (2) and (3) are repeated a certain number of times (number of trials t). For example, if $n = 10$ and the number of trials (denoted by t) is $t = 500$, one will get $10 \times 500 = 5000$ neural network predictors, being 500 predictions for each single test point sample. In this case, the number of neural network predictors will be equal to $n \times t$.
- (5) Summarize the results using the repeated predictions by constructing the empirical distribution function for each variable in play, as can be seen in Equation (3). The RMSEs can also be obtained for each testing point (Equation 4).

In practical applications, the user will be interested in applying the neural networks generated by the resampling process to new input variables where the outputs are unknown. In this case, the RMSE will not be available. However, other statistics can be computed for each predicted point, such as mean, standard deviation (uncertainty), quartiles and coefficient of variation. Consequently, an uncertainty map can be provided.

4. Material and methods

The proposed method was tested using soil chemistry spatial data. We also investigate whether the proposed method is able to provide good predictions under conditions of low data density.

4.1. Available datasets

The data set used consists of soil chemical attributes, as follows: potential of hydrogen (pH); exchangeable calcium (Ca); and phosphorus concentration (P). They were obtained by analyzing soils from 118 georeferenced points following standard laboratory methods (Figure 2). The coordinates are given in the UTM system. These chemical attributes play an important role in the soil fertility, soil life and soil management practices. Furthermore, the chemical attributes have different variability as can be seen in Table 1, which makes it an interesting problem for the application of the proposed method.

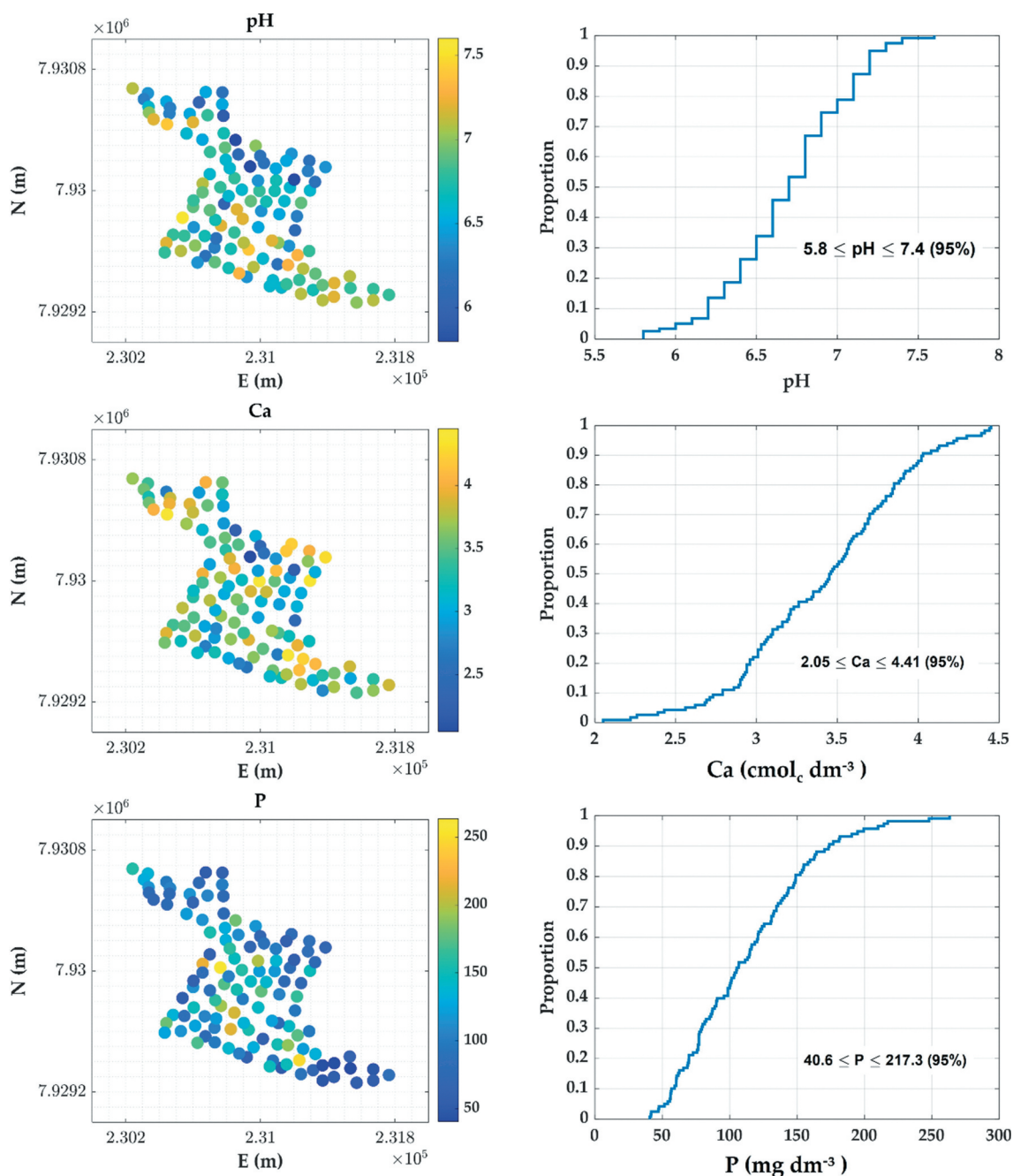


Figure 2. Data sets of the soil chemistry attribute.

4.2. Spatial neural network development

A feed-forward neural network with back-propagation learning procedure was developed by using the UTM coordinates as input data set and the soil chemical attributes (pH, Ca and P) as the targets. The command-line functionality of the MATLAB's Neural Network toolbox (R2019b) was used for training and validation of the neural network. After several trials, the optimum neural network architecture found was defined by three hidden layers, with the first layer composed of three neurons, the second of 14, and the last of a single neuron, i.e. [3 14 1] (Figure 3). Since the input data is only composed of UTM

Table 1. Original dataset characterization.

Data Set	Minimum	Maximum	Mean	Standard Deviation	Coefficient of Variation (%)
pH	5.8	7.6	6.7	0.4	5.6
Ca	2.1	4.5	3.4	0.5	14.9
P	40.6	190.7	104.8	38.2	36.5

PH = potential of hydrogen; Ca = exchangeable calcium ($\text{cmol}_c \text{ dm}^{-3}$); P = phosphorous contents (mg dm^{-3}); SD = standard-deviation; CV = coefficient of variation.

coordinates (E, N), the neural network is called a spatial neural network. This architecture was also found in Cagliari, Veronez, and Alves (2011).

Among the best practices for training a neural network is to normalize the dataset to speed up the learning and to get a faster convergence (Huang et al. 2020). Hence, both input and target values were normalized as follows:

$$y_n = 2 \times \left(\frac{y - y_{min}}{y_{max} - y_{min}} \right) - 1 \quad (5)$$

being: y_n the inputs or targets mapped to interval $[-1, 1]$, y the original inputs or target data, y_{min} and y_{max} the minimum and maximum values obtained from the original inputs or targets, respectively. In this case, the normalized function gives the input/target values between $[-1, 1]$.

The log-sigmoid transfer function (activation function) was applied in the intermediary layers (hidden layers) in order to compute a layer's output from its net input. The logistic function (denoted by φ) adopts values between 0 and 1, as follows

$$\varphi(x_i) = \frac{1}{1 + e^{-x_i}} \quad (6)$$

where x_i is a value coming from the linear combination of the inputs, weights and bias for a given neuron "i," as follows

$$x_i = w_{1i} \times I_1 + w_{2i} \times I_2 + \dots + w_{ni} \times I_n + \theta_i \quad (7)$$

with w being the weights parameters of the network, I the input data, and θ the bias parameter for 1, ..., n input sample data.

During the learning stage, it is customary and convenient to minimize the Mean Squared Error (MSE) between the targets (t_i) and estimated output (o_i), as follows (Haykin 1999):

$$\chi^2 = \sum_{i=1}^m (t_i - o_i)^2 \rightarrow \min \quad (8)$$

The weights and biases of the neural network were estimated by minimizing the function in (4) in order to produce the desired outcome, i.e. the chemical attributes. Here, Levenberg–Marquardt algorithm (LM), also known as the Damped Least-Squares (DLS) method, was employed to estimate the unknown parameters of the neural network, i.e. weights and biases (Levenberg 1944; Madsen, Nielsen, and Tinglef 2004; Marquardt 1963; Gavin 2020). The Nguyen-Widrow algorithm (LW) was used to initialize the weights and biases (Nguyen and Widrow 1990).

4.3. Experiments

At first, we used the 100% of the available data (118 points) for the application of the Jack-1 T re-sampling method (Figure 2). In this stage, the cross-validation is based only on the test set, which consists of the internal validation of the resampling procedure. The number of trials was adopted as $t = 100$, which provided 11,800 neural network predictors for the case where we have used 100% of the available dataset. The size of the validation set is fixed by randomly taking 10 points for each trial. The aim of the validation set is to improve the generalization of the network learning.

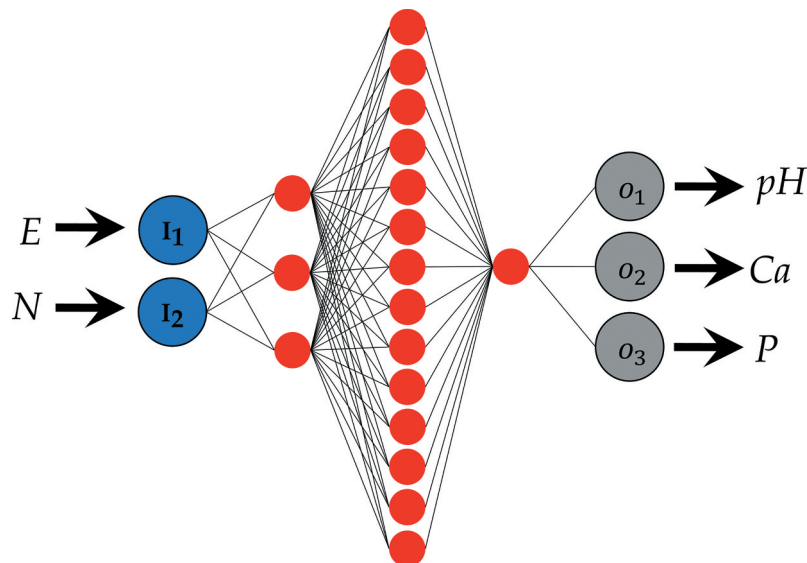


Figure 3. Topology of the feed-forward neural network for prediction of soil chemical attributes.

When the network starts to over-fit the training set, the performance on the validation set typically begins to decrease. When the validation error increases for a specified number of epochs, the training is stopped, and the weights and biases at the minimum of the validation error are returned. This method for avoiding over/under-fitting is called early stopping (Reitermanová 2010). The learning process is configured to stop when the network performance on the validation set fails to improve or remains the same for six epochs (iterations). Since the validation set has an effect on how to train the network and how to stop it, it was considered as part of the training process. Therefore, the training was taken as being the set without the test set.

Still, at this stage, we investigate to what extent Jack-1 T is able to provide good predictions under conditions where the samples are smaller than the original dataset. For this purpose, the original dataset ($n = 118$ points) was intentionally and randomly reduced to $\sim 40\%$ (47 points). The number of trials was taken as $t = 100$, which provided 4700 neural network predictors for the case where we have used 40% of the available dataset. Thus, 100 predictions were provided for each sample test point, which allowed an internal validation with respect to the original dataset available.

The remaining (71 points) which have not participated in the Jack-1 T procedure, were used to evaluate the performance of the Jack-1 T in terms of external validation. They were totally left out of network's training and testing set and therefore were taken as being out-of-sample sets of the resampling procedure. This guarantees an unbiased evaluation of the neural network-based predictions. Thus, 4700 neural predictors generated from the resampling of the 40% of the data (47 points) were applied to each of the 71 points, which were left out of the resampling procedure (i.e. 60% remaining that were part of the out-of-sample set). Consequently, each sample point was predicted 4700 times, which allowed for interval prediction instead of the classical point prediction one.

The analysis was performed from two perspectives: (i) quantitatively by comparing the descriptive statistics between the original attributes and those predicted using the Jack-1 T-based neural network; (ii) qualitatively by comparing the class to which each individual prediction belongs with respect to its original class from classification tables of soil chemical attributes.

5. Results and discussion

5.1. Quantitative evaluation

On average, there were no differences, according to the levels of significance, between the predictions from different sample sizes and the original dataset in terms of internal validation (Figure 4). The maximum and minimum values (shown as solid black lines in Figure 4) for the predictions are the maximum of the 97.5th percentile values ($\alpha = 0.975$ computed from Equation 7) and the minimum of the 2.5th percentile values ($\alpha = 0.025$ computed from Equation 7), respectively. These percentiles correspond to the lower ($\alpha = 0.025$) and upper ($\alpha = 0.975$) boundaries of the 95% empirical confidence interval, respectively. This means that the predictions that fall outside this confidence interval are considered outliers and are not considered in the solutions. The use of lower and upper bounds was suggested by Maldaner, Molin, and Spekken (2020). Regardless of the data size, we observed that the predictions for pH and Ca showed a slightly greater variability than the original data (ref). On the other hand, the maximum values of the predictions for P were underestimated. This means that it would be necessary to develop a specific neural network architecture for the variable P. However, here we are not concerned with the issue of improving the architecture by adding new parameters (more hidden layers and/or more nodes) or by tuning the hyperparameters on the training set. Instead, we try to adjust the prediction interval with respect to the interval of the original dataset. In fact, the variability is better described by adjusting the upper bound to the maximum values instead of the

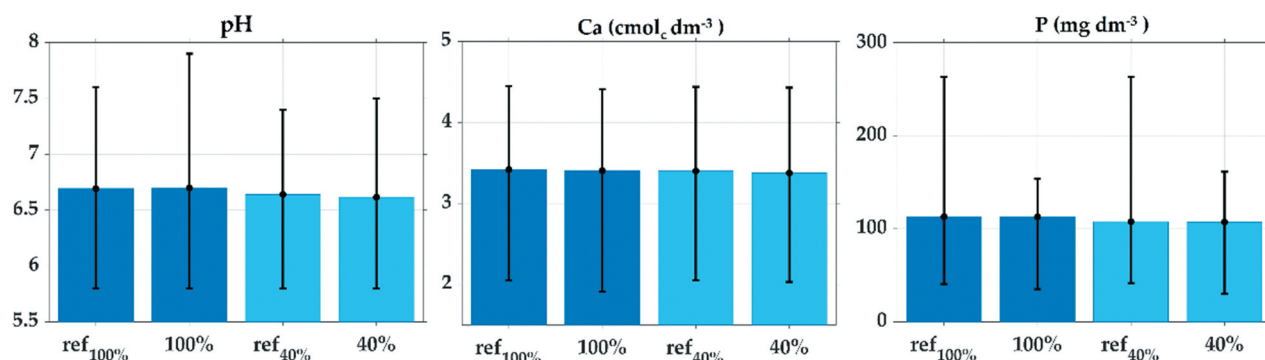


Figure 4. Bar chart of original dataset (ref) and neural network-based predictions for 118 (100%) and 47 (40%) sampling points of pH, Ca and P.

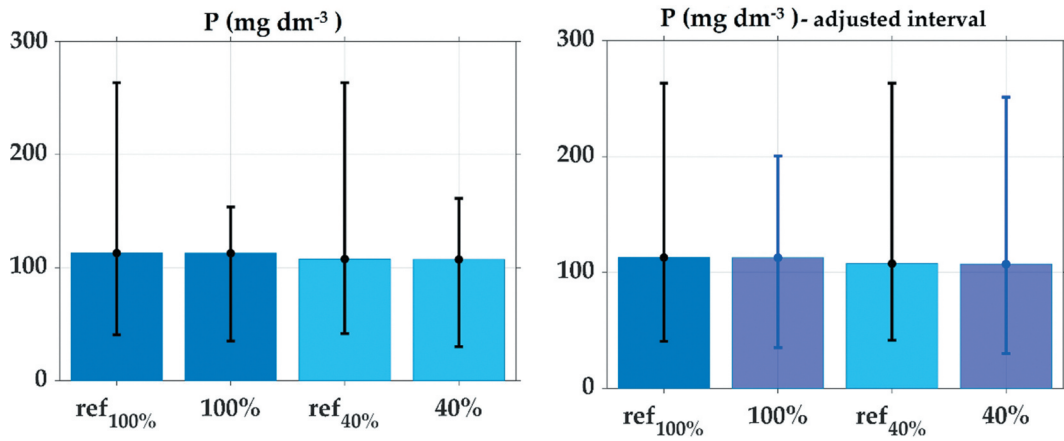


Figure 5. Bar chart of upper bound predictions based on $\alpha = 0.975$ (left) and upper bound given by the maximum-predicted values (right) for P.

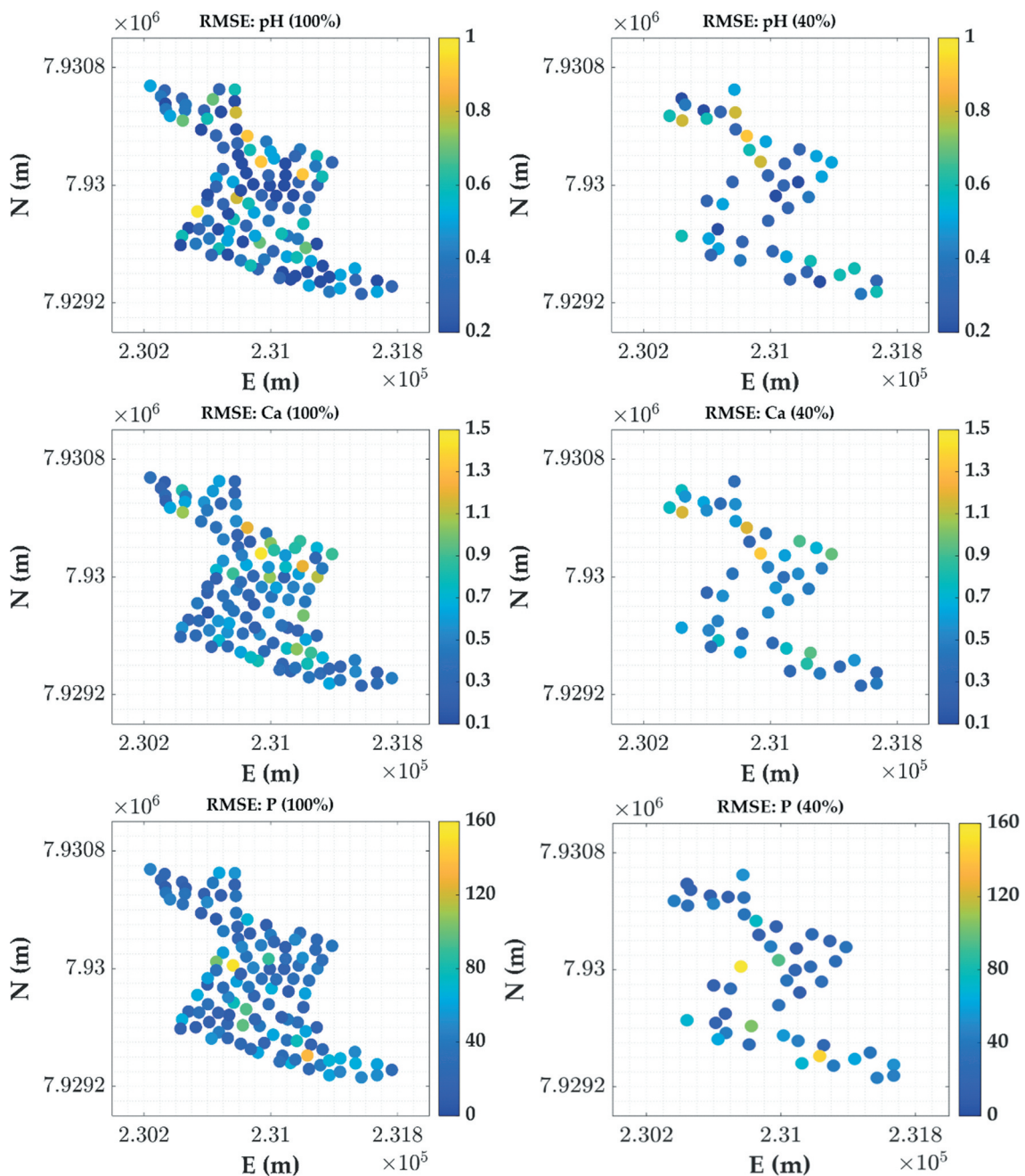


Figure 6. Bar chart of upper-bound predictions based on $\alpha = 0.975$ (left) and upper bound given by the maximum-predicted values (right) for P.

97.5th percentile values (Figure 5). In terms of internal validation (Figure 6), 95% of RMSE's were below ~ 0.7 (pH), $\sim 1 \text{ cmol}_c \text{ dm}^{-3}$ (Ca) and $\sim 100 \text{ mg dm}^{-3}$ (P).

In terms of external validation (Figure 7), 90% of RMSE's were below ~ 0.7 (pH), $\sim 0.9 \text{ cmol}_c \text{ dm}^{-3}$ (Ca) and $\sim 70 \text{ mg dm}^{-3}$ (P). These results are consistent with those from the internal validation. This means that the neural network model is able to generalize the results. Consequently, the use of 47 sampling points to generate a prediction model based on neural networks is sufficient, and a higher cost of sampling in the field would not be necessary.

5.1. Qualitative evaluation based on the out-of-sample set

The predictions for pH, Ca and P (71 points: 60% out-of-sample set) and their original values (true values) are classified according to Table 2. The success rate of the predictions was computed as being the ratio of the number of predictions correctly identified to their class and the total number of sampling points. The most probable value of each prediction is given by the mean value, which is used to classify the attributes according to Table 2. We recall that 4700 neural predictors generated from the resampling of the 40% of

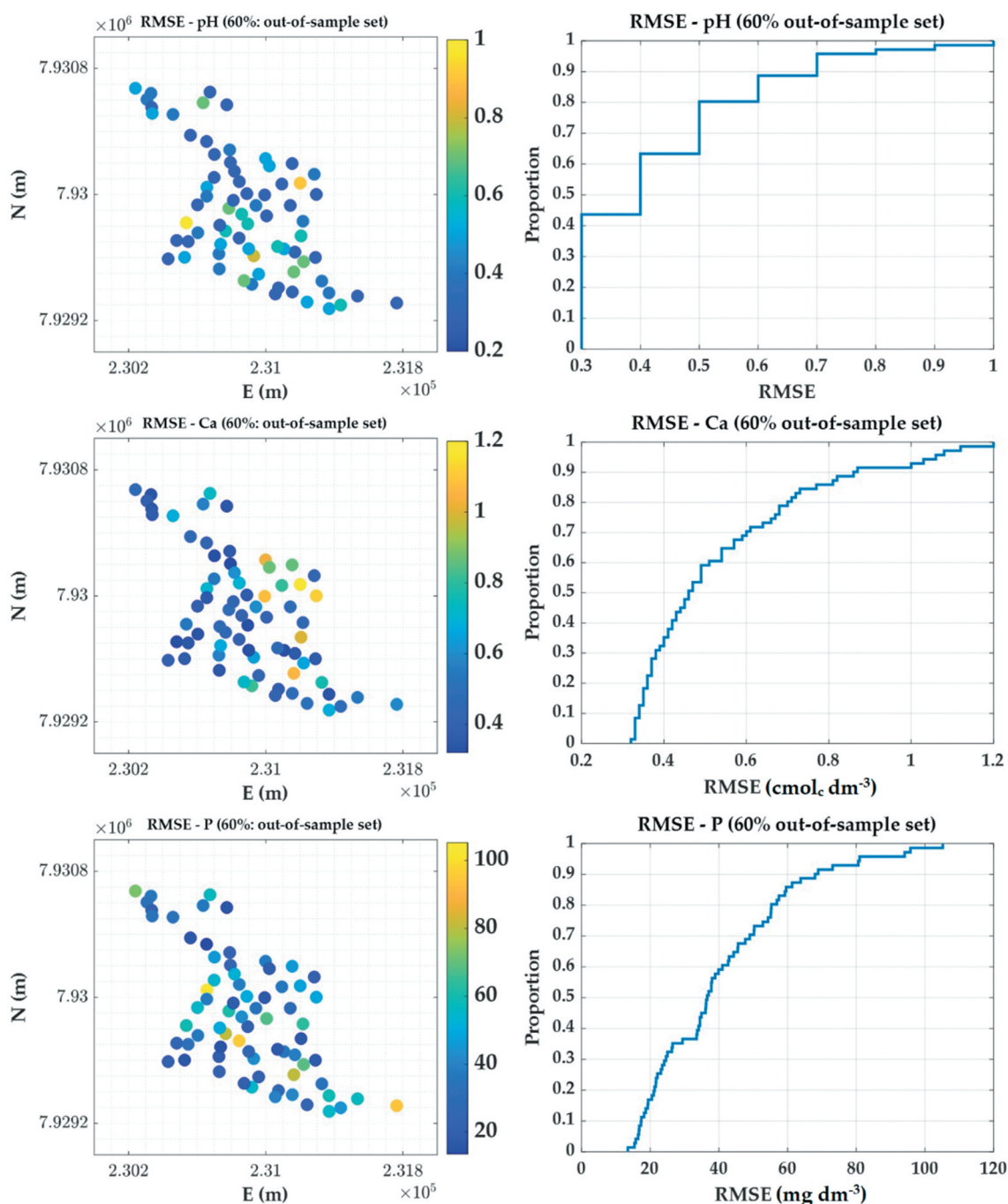


Figure 7. RMSE spatial distribution (left) and overall proportion of the RMSE (right) of the pH, Ca and P computed from the application of 4700 neural predictors generated from the resampling of the 40% of the data (47 points) into the 71 out-of-sample points (60%).

the data (47 points) were applied to each of the 71 points (60%) which were left out of the resampling procedure (out-of-sample set).

The results are displayed in Figure 8. The success rates were 72% (pH), 87% (Ca) and 89% (P), respectively. All means of the predictions were classified within the same class. This means that the resampling-based regression feed-forward neural network is shown not to be sensitive to abrupt changes, which can be interpreted as outliers. The original dataset also had a predominance of the same class although there are some anomalous points.

Since we have at our disposal the uncertainty of each sampling point, it is also possible to analyze it in terms of the confidence interval instead of just considering the mean of each prediction. The spatial distribution of the both standard-deviation (σ) and its distribution for the predictions are displayed in

Table 2. Classification of the pH, Ca ($\text{cmol}_c \text{dm}^{-3}$) and P (mg dm^{-3}).

Attribute	Class	Values
pH	Very Low	< 4.5
	Low	4.5–5.4
	Suitable	5.5–6.0
	High	6.1–7.0
Ca	Very High	> 7.0
	Very Low	≤ 0.40
	Low	0.41–1.20
	Medium	1.21–2.40
P	High	2.41–4.00
	Very High	> 4.00
	Very Low	≤ 5
	Low	6–12
	Medium	13–30
	High	31–60
	Very High	> 60

Classification of pH and Ca from Alvarez et al. (1999), and P from Raji et al. (1996).

(Figure 9). In this case, the success rates are based on the confidence interval given by mean Probable Error

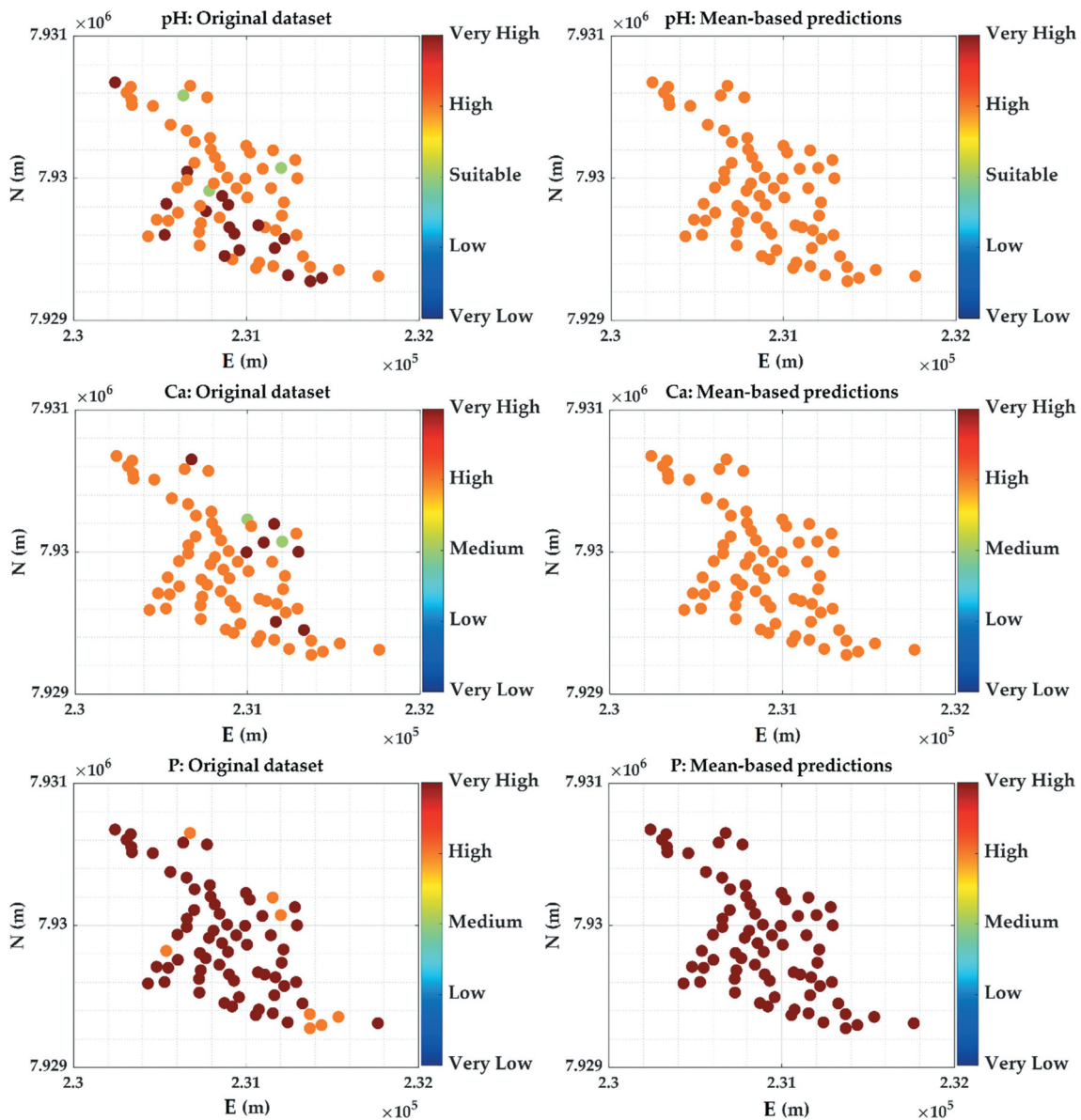


Figure 8. Classification of the pH, Ca and P based on 71 sampling points (60% out-of-sample set) for original dataset (left) and the mean-based Jack-1 T predictions (right).

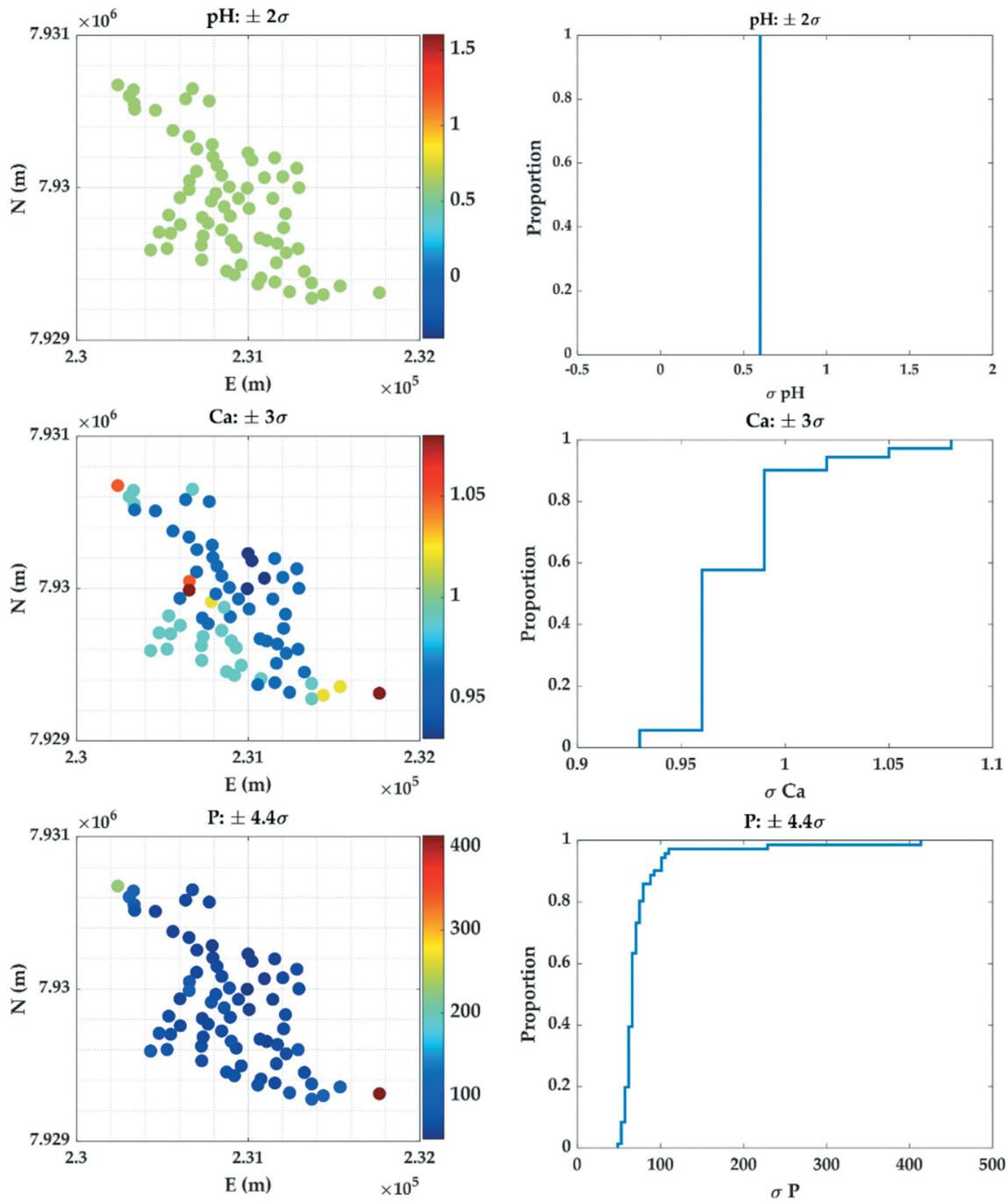


Figure 9. Uncertainties of the pH, Ca and P predictions based on $\pm 2\sigma$, $\pm 3\sigma$ and $\pm 4.4\sigma$ probability, respectively, for the 71 sampling points (60% out-of-sample set).

(\pm PE). Here, PE was defined as being $\pm 2\sigma$ for pH, $\pm 3\sigma$ for Ca and $\pm 4.4\sigma$ for P, where σ is the standard-deviation of the predictions. These PE choices were based on the variability behavior described by the results of the 4700 neural predictors generated from the resampling of the 40% of the data (47 points), as can be seen in the previous section (Figures 5 and 6). The success rate occurs when the original field value is within the predicted confidence interval. For this case, the success rates were 92%, 93% and 92% for pH, Ca, and P, respectively. The confidence interval stayed exactly between [6, 7.2] for all pH predictions, which corresponds to the classes (Table 2) “Suitable” (lower

bound of -2σ) and “High” (upper bound of $+2\sigma$), respectively. On the other hand, all upper bounds of Ca predictions ($+3\sigma$) were classified as “Very High,” but the lower bounds (-3σ) were classified as “Medium” for 30 points (42%) and the others 41 points (58%) as “High.” The upper bounds ($+4.4\sigma$) for P were classified as “Very High,” whereas the lower bounds (-4.4σ) were classified as “High” for 58 points (82%), “Medium” for six points (8%) and “Very Low” for seven points (10%).

The proposed method makes it possible to describe the uncertainties in the predictions of each sampling point. This can help neural network developers in

decision-making, such as detecting regions that need more samples to reduce errors and improve the model. For instance, Qi et al. (2020) could use the Jack-1 T idea in their convolutional neural network for regression as both the study area and available dataset of their work are relatively small. However, the limitation of the Jack-1 T method is due to the sample size. If the sample size is too large, the method may be unfeasible in terms of computational time and hardware resources. For instance, if we had a dataset of $n = 10,000$, and if we took the number of trials $t = 100$, the total number of neural networks would be one million so each prediction would be repeated 100 times. These large sample set scenarios are often encountered in many remote-sensing imagery classification by deep learning algorithms, as can be seen here Liao et al. (2020) and Hong et al. (2021). In this case, it is very common to use Hold-out or K-fold methods. Thus, the proposed method may be extended to the case of Hold-out and K-fold by repeating these processes by resampling a certain number of times.

6. Conclusion

Here, we describe a data-based re-sampling method called Jack-1 T in the case of cross-validation back-propagation associated with neural networks. The proposed method can be used to construct inferential procedures for modern statistical spatial data analysis. The method replaces the theoretical derivations required in applying traditional methods by repeatedly re-sampling the original data and making inferences from the re-samples. The advantage of these re-sampling techniques is that they allow us to determine the empirical probability distribution for the predictor rather than simply providing a point prediction. Consequently, it allows us to measure the prediction performance for each individual sampling point.

The results, based on the dataset size simulation, show that a 60% reduction in available data offers compatible accuracy in relation to the full dataset, and therefore reduces the cost of sampling in the field. Therefore, the proposed method supports in decision-making to define the sample size in the field. The re-sampling method spatially characterizes the points of greater or lesser accuracy and uncertainty. The re-sampling method increases the success rate by using interval prediction instead of using the mean. Although we restrict it to the neural network model, the re-sampling method proposed can also be extended to other modern statistics tools such as Kriging, Least Squares Collocation (LSC), Convolutional Neural Network (CNN), and so on. Although we restrict the Jack-1 T resampling method to a neural network for

a regression problem, it can be applied to classification problems and for the case where deep learning algorithms is in play. Furthermore, the proposed method can be extended to other resampling methods such as Hold-out and K-fold.

Disclosure statement

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

Notes on contributors

Bruno Póvoa Rodrigues received his MSc degree in Agriculture and Geospatial Information from the Federal University of Uberlandia, Monte Carmelo city, Minas Gerais state, Brazil, in 2021. He has developed his research in the field of artificial neural networks applied in agriculture.

Vinicius Francisco Rofatto received his PhD degree in Remote Sensing from the Federal University of Rio Grande do Sul, Brazil, in 2020. He is currently holding lecturer and researcher positions at the Federal University of Uberlandia, Monte Carmelo city, Minas Gerais state, Brazil. In 2018, he was appointed the Grand Master of Cartographers Order by the Brazilian Cartography Society. In 2021, he received the Honorable Mention of the CAPES Thesis 2021 Award in the field of Geosciences. His research interests include Estimation Theory, Quality Control, Computational Methods for solving linear and nonlinear geodetic problems, Monte Carlo Methods, Global Navigation Satellite Systems and Deep Learning in Computer Vision for geospatial applications.

Marcelo Tomio Matsuoka received his MSc and PhD degrees in Cartographic Sciences from the São Paulo State University, Brazil, in 2003 and 2007, respectively. He has held lecturer and researcher positions (Undergraduate Surveying Engineering and Graduate Program in Agriculture and Geospatial Information) at Federal University of Uberlandia, Monte Carmelo city, Minas Gerais state, Brazil. In 2021, he received the Honorable Mention of the CAPES Thesis 2021 Award in the field of Geosciences. His research interests include Global Navigation Satellite Systems, Quality Control of Geodetic Networks and Monte Carlo and Intelligent Computational Methods Applications in Geodesy.

Talita Teles Assunção received her MSc degree in Agriculture and Geospatial Information from the Federal University of Uberlandia, Monte Carmelo city, Minas Gerais state, Brazil, in 2021. He has developed her research in the field of environmental planning and management.

ORCID

Bruno Póvoa Rodrigues  <http://orcid.org/0000-0002-5275-9049>

Vinicius Francisco Rofatto  <http://orcid.org/0000-0003-1453-7530>

Marcelo Tomio Matsuoka  <http://orcid.org/0000-0002-2630-522X>

Talita Teles Assunção  <http://orcid.org/0000-0001-9607-4060>

Data availability statement

For those interested in the dataset of this work, please send their request to the corresponding author by e-mail or access <https://data.mendeley.com/datasets/z3399p5h9f/>.

References

- Aboufazeli, S., A. Jahani, and M. Farahpour. 2021. "A Method for Aesthetic Quality Modelling of the Form of Plants and Water in the Urban Parks Landscapes: An Artificial Neural Network Approach." *MethodsX* 8: 1–8. doi:10.1016/j.mex.2021.101489.
- Alvarez, V.V.H., R.F. Novais, N.F. Barros, R.B. Cantarutti, and A.S. Lopes. 1999. "Interpretation of Soil Analysis Results." In *Recommendations for the Use of Correctives and Fertilizers in Minas Gerais 5th Approximation*, edited by A. C. Ribeiro, P. T. G. Guimarães, and V. V. H. Alvarez. 1 ed., 25–32. (in Portuguese). Brazil, Minas Gerais, Viçosa: Soil Fertility Commission of the State of Minas Gerais.
- Aslam, N.S., M.R. Ibrahim, T. Cheng, H. Chen, and Y. Zhang. 2021. "ActivityNET: Neural Networks to Predict Public Transport Trip Purposes from Individual Smart Card Data and POIs." *Geo-spatial Information Science* 24 (4): 711–721. doi:10.1080/10095020.2021.1985943.
- Balmer, M., R. Weibel, and H. Huang. 2021. "Value of Incorporating Geospatial Information into the Prediction of On-Street Parking Occupancy: A Case Study." *Geo-spatial Information Science* 24 (3): 438–457. doi:10.1080/10095020.2021.1937337.
- Burman, P. 1989. "A Comparative Study of Ordinary Cross-Validation, V-Fold Cross-Validation and the Repeated Learning-Testing Methods." *Biometrika* 76 (3): 503–514. doi:10.2307/2336116.
- Cagliari, J., M.R. Veronez, and M.E. Alves. 2011. "Remaining Phosphorus Estimated by Pedotransfer Function." *Revista Brasileira Ciências Solo* 35 (1): 203–212. doi:10.1590/S0100-06832011000100019.
- Efron, B. 1980. "The Jackknife, the Bootstrap and Other Resampling Plans." *Technical Report, Stanford University* 63: 1–135.
- Efron, B. 1982. "The Jackknife, the Bootstrap and Other Resampling Plans." *Society for Industrial and Applied Mathematics, Stanford University*. doi:10.1137/1.9781611970319.
- Efron, B. 1992. "Jackknife-After-Bootstrap Standard Errors and Influence Functions." *Journal of the Royal Statistical Society: Series B (Methodological)* 54 (1): 83–111.
- Gavin, H.P. 2020. "The Levenberg-Marquardt Algorithm for Nonlinear Least Squares Curve-Fitting Problems." *Duke University: Department of Civil and Environmental Engineering*, September: 1–19.
- Haykin, S. 1999. *Neural Networks: A Comprehensive Foundation*. 2 ed. Hoboken: Prentice Hall.
- Hong, D., L. Gao, J. Yao, B. Zhang, A. Plaza, and J. Chanussot. 2021. "Graph Convolutional Networks for Hyperspectral Image Classification." *IEEE Transactions on Geoscience and Remote Sensing* 59 (7): 5966–5978. doi:10.1109/TGRS.2020.3015157.
- Huang, L., J. Qin, Y. Zhou, F. Zhu, L. Liu, and L. Shao. 2020. "Normalization Techniques in Training DNNs: Methodology, Analysis and Application." *arXivLabs* 1: 1–20.
- Jahani, A., S. Allahverdi, M. Saffariha, A. Alitavoli, and S. Ghiyasi. 2021a. "Environmental Modeling of Landscape Aesthetic Value in Natural Urban Parks Using Artificial Neural Network Technique." *Modeling Earth Systems and Environment*: 1–10. January. doi:10.1007/s40808-020-01068-2.
- Jahani, A., and M. Saffariha. 2021b. "Modeling of Trees Failure under Windstorm in Harvested Hyrcanian Forests Using Machine Learning Techniques." *Scientific Reports* 11 (1124): 1–13. doi:10.1038/s41598-020-80426-7.
- Levenberg, K. 1944. "A Method for the Solution of Certain Non-Linear Problems in Least Squares." *Quarterly of Applied Mathematics* 2 (2): 164–168. doi:10.1090/qam/10666.
- Liao, X., K. Li, X. Zhu, and K.J.R. Liu. 2020. "Robust Detection of Image Operator Chain with Two-Stream Convolutional Neural Network." *IEEE Journal of Selected Topics in Signal Processing* 14 (5): 955–968. doi:10.1109/JSTSP.2020.3002391.
- Madsen, K., H.B. Nielsen, and O. Tinglef. 2004. "Methods for Nonlinear Least Squares Problems." In *Informatics and Mathematical Modelling*. 2 ed. 1–58. Lyngby: Informatics and Mathematical Modelling (IMM), Technical University of Denmark (DTU).
- Maldaner, L.F., J.P. Molin, and M. Spekken. 2020. "Methodology to Filter Out Outliers in High Spatial Density Data to Improve Maps Reliability." *Scientia Agricola* 79 (1): 1–7. doi:10.1590/1678-992X-2020-0178.
- Marquardt, D. 1963. "An Algorithm for Least Squares Estimation of Non-Linear Parameters." *Journal of the Society for Industrial and Applied Mathematics* 11: 431–441. doi:10.1137/0111030.
- Miller, R.G. 1974. "The Jackknife: A Review." *Biometrika* 61 (1): 1–15. doi:10.2307/2334280.
- Nguyen, D., and B. Widrow. 1990. "Improving the Learning Speed of 2-Layer Neural Networks by Choosing Initial Values of the Adaptive Weights." The International Joint Conference on Neural Networks, San Diego, June 17–21.
- Pan, L. 1998. "Resampling in Neural Networks with Application to Financial Time Series." PhD diss., University of Guelph.
- Qi, Y., S.C. Drolma, X. Zhang, J. Liang, H. Jiang, J. Xu, and T. Ni. 2020. "An Investigation of the Visual Features of Urban Street Vitality Using a Convolutional Neural Network." *Geo-spatial Information Science* 23 (4): 341–351. doi:10.1080/10095020.2020.1847002.
- Quenouille, M.H. 1949. "Approximate Tests of Correlation in Time-Series." *Journal of the Royal Statistical Society: Series B (Methodological)* 11 (1): 68–84.
- Raij, B.V., H. Cantarella, J.A. Quaggio, and A.M.C. Furlani. 1996. *Fertilization And Liming Recommendations For The State of São Paulo, (Technical Bulletin, 100)*. 2 ed. (in Portuguese).
- Reitermanová, Z. 2010. "Data Splitting." The WDS'10 Proceedings of Contributed Papers, Part I: Mathematics and Computer Sciences, Prague, June 1–4.
- Rofatto, V.F., M.T. Matsuoka, I. Klein, M.R. Veronez, and L. G. Silveira. 2020. "A Monte Carlo-Based Outlier Diagnosis Method for Sensitivity Analysis." *Remote Sensing* 12 (5): 1–41. doi:10.3390/rs12050860.
- Wang, L., and F. Yu. 2020. "Jackknife Resampling Parameter Estimation Method for Weighted Total Least Squares." *Communications in Statistics - Theory and Methods* 49 (23): 5810–5828. doi:10.1080/03610926.2019.1622725.
- Ziggah, Y.Y., Y.J. Hu, A.R. Tierra, and P.B. Laari. 2019. "Coordinate Transformation between Global and Local Datums Based on Artificial Neural Network with K-Fold Cross-Validation: A Case Study, Ghana." *Earth Sciences Research Journal* 23 (1): 67–77. doi:10.15446/esrj.v23n1.63860.