Radiation Physics and Chemistry 221 (2024) 111767

Contents lists available at ScienceDirect

Radiation Physics and Chemistry

journal homepage: www.elsevier.com/locate/radphyschem

A deep neural network for predicting soil texture using airborne radiometric data

Andrea Maino [a,b,*], Matteo Alberi [a,b], Alessio Barbagli [c], Enrico Chiarelli [a,b], Tommaso Colonna [c], Michele Franceschi [a,b], Fabio Gallorini [a,c], Enrico Guastaldi [c], Nicola Lopane [a,b,c], Fabio Mantovani [a,b], Dario Petrone [a,c], Silvio Pierini [c], Kassandra Giulia Cristina Raptis [a,b], Virginia Strati [a,b], Gerti Xhixha [d]

[a] Department of Physics and Earth Sciences, University of Ferrara, 44122, Ferrara, Italy
[b] INFN Ferrara Section, 44122, Ferrara, Italy
[c] GeoExplorer Impresa Sociale s.r.l., 52100, Arezzo, Italy
[d] Department of Physics, Faculty of Natural Sciences, University of Tirana, Blv. Zogu I, 1001, Tirana, Albania

ARTICLE INFO

Handling Editor: Dr. Chris Chantler

Keywords:
Deep neural network
Hyperparameters optimization
Airborne gamma-ray spectroscopy
Soil texture mapping
Potassium
Thorium

ABSTRACT

The ternary nature of soil texture, defined by its proportions of clay, silt, and sand, makes it challenging to predict through linear regression models from other soil attributes and auxiliary variables. The most promising results in this field have been recently achieved by Machine Learning methods which are able to derive non-linear, non-site-specific models to predict soil texture. In this paper we present a method of constructing a pair of Deep Neural Network (DNN) algorithms that can predict clay and sand soil contents from Airborne Gamma Ray Spectrometry data of K and Th ground abundances.

We tested the algorithm's hyperparameters through various configurations to optimize the DNNs' performance, effectively avoiding underfitting and overfitting of the models. This led to the creation of a high-resolution 20 m × 20 m soil texture map from dense AGRS data, significantly refining the previous map's granularity. The application of the obtained DNN models to unseen sites can be supported by future training on additional textural classes.

1. Introduction

In recent years, Machine Learning (ML) techniques compete with classic methods of statistical analysis especially in the study of complex ecosystems like the soil and its features. The complexity of soil texture characterization as a mix of three components (i.e., sand, silt, and clay) is being investigated successfully with ML through image analysis (Azadnia et al., 2022; Zhao et al., 2022) and regression of soil attributes (Wu et al., 2018; Zhang et al., 2020).

Although the high cation exchange capacity of some clay minerals soils facilitates the retention of positively charged ions, such as those of K and Th radioelements, there are no a priori models based on a positive correlation between these radioelements and the presence of clay. In the linear regression models that have been empirically studied in the last decade (Elbaalawy et al., 2016; Mahmood et al., 2013; Petersen et al., 2012; Spadoni and Voltaggio, 2013; Van Der Klooster et al., 2011) a clear site-dependency emerges. This limitation can be overcome by applying ML techniques as showed in recent studies based on a small dataset size which is not well suited for ML tasks (Heggemann et al., 2017; Priori et al., 2014).

Facing this issue, this work utilizes Airborne Gamma-Ray Spectroscopy (AGRS), a technique to efficiently collect big amounts of radiometric data, to study correlations between clay and sand soil contents and K and Th abundances through Deep Neural Network (DNN) algorithms.

2. Materials and methods

2.1. Study area and data taking

The Mezzano Lowland (Emilia-Romagna, Italy) is a flat and reclaimed area of ~189 km^2 fully devoted to agricultural activities. The

* Corresponding author. Department of Physics and Earth Sciences, University of Ferrara, Ferrara, 44121, Italy.
E-mail address: maino@fe.infn.it (A. Maino).

https://doi.org/10.1016/j.radphyschem.2024.111767
Received 22 November 2023; Received in revised form 5 April 2024; Accepted 10 April 2024
Available online 16 April 2024

data-taking has been carried out in this area using the Radgyro (Baldoncini et al., 2017, 2018), an experimental aircraft equipped with a modular 16 L-NaI(Tl) scintillation detector.

The survey has been divided into three flights carried out at a mean velocity of 102 km/h and at a mean height of 104 m, totalling 4 h and 45 min of flight time and 482 km linear distance covered (Maino et al., 2022). The 1469 data of K and Th abundances derived from an equal number of 10 s integrated AGRS spectra were spatialized using ordinary kriging with 500 m × 500 m pixels. To each of the 723 obtained pixels, soil texture (percentage of sand, silt, and clay) was assigned as ground truth data derived from the regional soil map of the Emilia Romagna Region (RER) (Tarocco, 2015).

The combined AGRS and RER data (723 data total) have been divided into a training dataset (578 data), a validation dataset (72 data) and a test dataset (73 data), following an 80:10:10 split (Le et al., 2023; Wang et al., 2022) via random sampling, making sure that all data subsets were evenly distributed in the entire study area.

2.2. DNN's basics

ML algorithms are designed to train on data features (here K and Th measured abundances) over a certain number of epochs (Table 1). During each epoch an algorithm updates its weights and biases (parameters) to make better and better estimations of the data labels (here clay and sand contents from the RER soil texture map). At the end of the entire training phase the algorithm has learned a model to predict data labels starting from data features without any a priori restrictions.

The calculations of a DNN are performed in its processing units, called nodes, that are organized in a series of layers. In a DNN, there are three types of layers: an input layer,[1] a series of hidden layers[2] and an output layer.[3] In a layer l each node i receives inputs x_j from each node j of the preceding layer $l-1$. The node i calculates a value y_i given by:

$$y_i = \sum_j w_{ij} x_j + b_i.$$

where w_{ij} is a weight that depends on both the input x_j and the node i, while b_i is a node-specific bias.

The value y_i gets then fed to an activation function (Table 1) which transforms it to assist the model in learning the intricate patterns present in the input data.

Table 1
Hyperparameters of the DNN analysed in this study and their respective definitions.

Hyperparameter	Definition
Width	Number of nodes in a given layer
Depth	Number of hidden layers
Batch size	Number of input data processed before the model's parameters are updated
Number of Epochs	Complete passages of data features through the DNN structure
Activation function	Function that takes as inputs the values calculated by a layer's nodes and produces an output that is passed on to the next layer
Loss function	Function that quantifies the difference between predicted and ground-truth labels
Optimizer	Algorithm that adjusts the model's parameters during training to minimize the loss function
Learning rate	Fraction of the optimizer-calculated parameter updates to apply to each parameter after a batch of input data has been analysed

In this work, we adopt the Rectified Linear Unit (ReLU) (Nair and Hinton, 2010) as the activation function for all hidden layers because of its wide use in regression problems (Sharma et al., 2017). This enables us to introduce non-linearity in the model, as the ReLU function sets all negative y_i values to 0 while leaving unmodified all positive ones.

The output value z_i (which represent an input of the following $l+1$ layer) can therefore be written as:

$$z_i = f(y_i) = max(0, y_i),$$

where f represents the ReLU function.

We developed a pair of independent DNNs predicting clay and sand soil contents respectively, both starting from K and Th abundance data as inputs.

In Table 1 we report the list of hyperparameters for our DNNs, which are all the non-trainable settings decided before training begins (Yang and Shami, 2020). Excluding the number of epochs and the activation function, all the hyperparameters values have been chosen after an optimization process aimed at finding the best configuration specifically for our task. The number of epochs has been automatically chosen by applying the Early Stopping regularization method. This method permitted to set a minimum value of 1 (to be achieved in the span of 5 epochs) for the improvement of the loss function between clay and sand contents predictions and ground-truth data. When, at any point during training, this condition wasn't met, the Early Stopping method interrupted the training process and restored the parameters that gave the lowest loss value recorded.

Both DNNs have been tested with various hyperparameters values to find the optimal configuration in terms of model prediction accuracy, computational time and model complexity.

Since in our study the data labels distribution doesn't present outliers, the prediction accuracy is inferred by evaluating the value of the loss function at convergence, where the loss function is defined as the Mean Squared Error (MSE) (Mohri et al., 2018). Computational time depends on the convergence speed, which is determined by the number of epochs required to reach convergence, and on the time required for each epoch of training. Finally, the complexity of a model can be inferred via the sum of the number of trainable parameters learned by each layer, which can be calculated as:

$$t_l = (n_{l-1} + 1) \cdot n_l,$$

where t_l represents the number of trainable parameters of layer l, while n_l and n_{l-1} represent the number of nodes of layers l and $l-1$ respectively.

2.3. Optimization process

The prediction accuracy and the computational time of a DNN are dependent on the hyperparameters' values adopted, that need to be optimized to avoid possible underfit or overfit situations.

Underfit happens when the error between the model's predicted labels and the ground-truth labels is too high and/or doesn't reach convergence. Common causes of underfit are a low learning rate value, a small epoch number or a too simple DNN structure (small width and depth). In contrast, overfit happens when the model is too heavily tuned on training data (e.g., too complex models, too many training epochs), failing to generalize to unseen data (validation dataset).

Model error, as well as possible underfitting or overfitting situations, are inferred via the loss function which gets minimized by the optimizer updating the model's parameters.

Convergence to the loss minimum is helped by standardizing the different input data features to a similar range. This step ensures that all data features (in our case K and Th abundances) have a similar impact on the determination of the updated model parameters during training. Feature normalization is here performed in the input layer (first layer) of

[1] The first layer of the DNN. This layer receives raw data features as inputs and sends outputs to the next layer.

[2] Neither first nor last layers of the DNN. These layers receive inputs from a previous layer and send outputs to the next layer.

[3] The last layer of the DNN. This layer receives inputs from the last hidden layer and produces predictions as final outputs.

both developed DNNs via the following equation:

$$\hat{x}_j^i = \frac{x_j^i - \mu_j}{\sigma_j},$$

where $\hat{x}_j^i$ is the normalized feature j (here either K or Th abundance) of the data x^i, x_j^i is the input feature j of the data x^i, and μ_j and σ_j are the mean value and the standard deviation of feature j across input data.

The training dataset has been utilized for conducting a series of investigations aimed at assessing various hyperparameters values effects on the training process and enhancing the DNNs' performances via their optimization.

The DNN structure has been tuned adjusting the width between the values 4 and 16 (kept constant throughout all hidden layers) and the depth between the values 2, 4, and 8 (Fig. 1a). In addition, a variety of batch size values have also been tested for understanding their impact on both convergence speed and total computational time (Fig. 1b). Furthermore, different algorithms are adopted as optimizers for the DNN, i.e., *Adadelta*, *Adagrad*, *Adam*, *Adamw*, *Adamax*, *Nadam*, *Adafactor*, *Ftrl*, *RMSprop* and *Stochastic Gradient Descent* (*SGD*) (Fig. 1c). Finally, a range of values from 10^{-1} to 10^{-4} have been studied for the effect of the learning rate on the training process (Fig. 1d).

When not directly tested, the hyperparameters in all performed tests are those from the final configuration.

3. Results

The performed tests highlighted some key points to consider when optimizing a DNN's hyperparameters.

When testing for different width-depth configurations (Fig. 1a) we learned how complex models improve convergence speed, but at the cost of being more prone to overfit. It is therefore important to find the right balance between convergence speed and model complexity.

The simplest DNN configuration (depth = 2, width = 4, 37 learnable parameters) doesn't reach convergence within 40 epochs, while the most complex configuration (depth = 8, width = 16, 1969 learnable parameters) converges more rapidly at the expense of a substantial increase in the number of parameters. The medium-sized configuration with depth = 4 and width = 16 has been chosen for the final DNN structure as a balance between convergence speed (27 epochs) and computational complexity.

The test performed on the batch size (Fig. 1b) uncovered instead a non-linearly scaling inverse relation between total computational time and both convergence speed and batch size, highlighting the need to optimize both time and generalizability of the model in a synchronous way. The highest convergence speed is shown by the model utilizing the lowest batch size value, as this had the most opportunities to update internal parameters values during each epoch. This behavior is not reflected in the total computational time, with the model utilizing a batch size of 4 requiring more than double the time needed by the model with a batch size of 64 to complete training. Although large batch size values require less total computational time to complete the training process, we opted for the middle-ground value of 16 to both reduce computational time and prevent loss of generalizability, as large values drop the generalizing ability of the network (Keskar et al., 2016).

Out of the algorithms tested as optimizers for our DNNs (Fig. 1c), not all showed convergence (i.e., *Adadelta*, *Adagrad*, *Ftrl* and *Adafactor*). All the tested algorithms have more (sometimes optimizer-specific) hyperparameters than simply the learning rate, so those would also need to be optimized especially for the non-converging algorithms. Among the algorithms that produced converging models we chose *Adam* as the final DNN optimizer, with the intention of avoiding instability at convergence.

When testing the effects of different learning rates on the training process, very high values like 10^{-1} and 10^{-2} produce high instability levels at convergence (Fig. 1d), with the loss curve referring to the

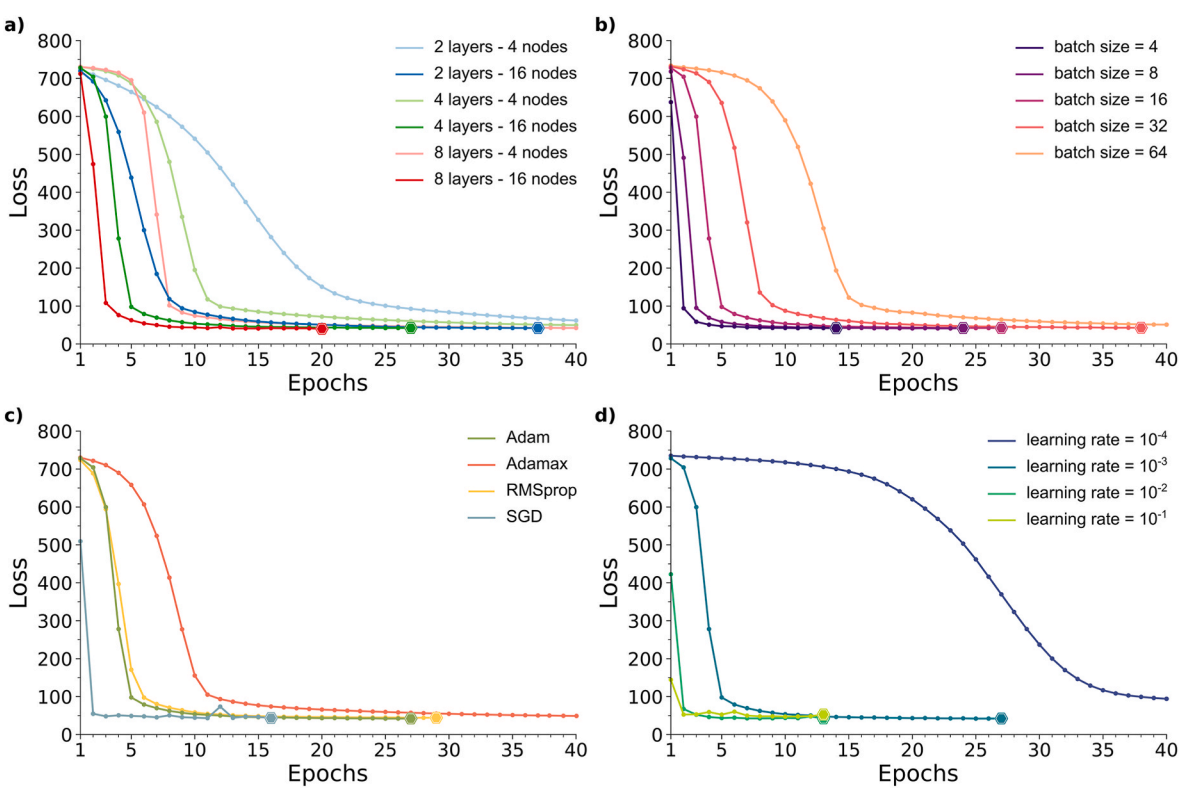

Fig. 1. Loss curves between clay predictions and ground-truth data for variations in a) width and depth of the DNN's structure, b) batch size, c) optimizer used, and d) learning rate. For each curve, hexagons mark the end of the training process when this happens before epoch 40. In panel c) only the converging loss curves are shown.

learning rate value of 10^{-2} showing a bump at epoch 12. On the contrary, the model with the lowest learning rate value (10^{-4}) can't reach convergence in the 40 epochs limit. The most suited learning rate value is therefore 10^{-3}, which produces the quickest stable convergence. Changes in learning rate values have a big impact on training, so this hyperparameter needs to be carefully optimized.

The results of the previous tests have been evaluated on both the training and the validation datasets, where the loss curves for both the clay-predicting model and the sand-predicting one show no sign of underfit or overfit since they reach convergence (Fig. 2).

Since the validation dataset differs in mean RER clay content (25.8 %) and mean RER sand content (37.1 %), we normalize the models' MSEs over these mean values utilizing the Coefficient of Variation of the Root Mean Square Error (CV(RMSE)) between ground-truth and predicted labels defined as:

$$CV(RMSE) = \frac{1}{\overline{Y}}\sqrt{\frac{\sum_{i=1}^{N}(Y_i - \widehat{Y}_i)^2}{N}},$$

where $\overline{Y}$ is the mean value of the ground-truth labels, Y_i and $\widehat{Y}_i$ are the i-th ground-truth and predicted label values respectively, and N is the total number of labels. When evaluated on clay and sand content predictions, we obtain CV(RMSE) values of ~0.25 and ~0.28 respectively. While the sand-predicting model converges to a higher MSE value than the clay-predicting one (Fig. 2), the normalized variations of their prediction errors are actually comparable.

The generalizability of the model was explored through an analysis correlating clay and sand values across 73 spatially randomly distributed data points that were not previously utilized during the training/validation phases. The determination coefficients and slope coefficients of correlation for clay (sand) are found to be 0.53 (0.52) and 0.93 (0.91) respectively, as illustrated in Fig. 3, demonstrating a good agreement between the model's estimations and the RER data.

Since the AGRS dataset, thanks to its size (1469), can produce K and Th abundance maps that outmatch the resolution of the RER soil texture map (500 m × 500 m), the clay-predicting and sand-predicting models have been used to obtain a high resolution (20 m × 20 m) soil texture map (Fig. 4a), which shows the predictions classified following the United States Department of Agriculture (USDA) definitions (Fig. 4b).

The obtained map shows a prevalence of Loam and Clay loam textural classes, followed by Sandy loam, Sandy clay loam and Clay.

4. Conclusions

In this work, we constructed a pair of DNNs trained on 578 K and Th abundance data measured via AGRS to predict clay and sand soil contents in the Mezzano Lowland (Emilia-Romagna, Italy).

Through a series of tests we optimized the final hyperparameters configuration, which includes 4 hidden layers composed by 16 nodes each, the mean squared error as the loss function minimized by the optimizer *Adam*, a learning rate of 10^{-3}, a batch size of 16, ReLU as the activation function and an epoch number determined by Early Stopping (27 for the clay-predicting DNN and 37 for the sand-predicting one). This configuration prevented both underfit and overfit of the DNN models, which show comparable CV(RMSE) values for clay and sand predicted labels (~0.25 and ~0.28 respectively).

By applying the obtained models to the highly dense AGRS data we obtained a 20 m × 20 m resolution soil texture map of the surveyed area, improving the pre-existing 500 m × 500 m resolution RER soil texture map.

Having been trained on AGRS data from soils belonging to 7 textural classes, both obtained models are prone to be non-site-specific, especially in the case of a supplementary future training with AGRS data from soils belonging to the remaining 5 soil textural classes.

Finally, since the prediction errors are presumably related to radiometric and soil texture data uncertainties, a refinement of the methodology could implement this kind of data as inputs for further improving the prediction of soil texture.

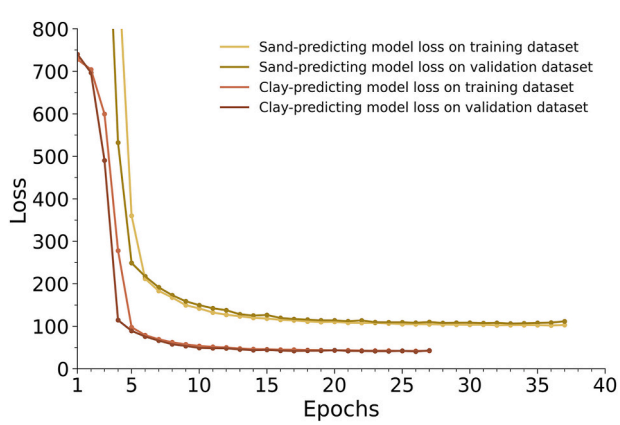

Fig. 2. Clay-predicting model loss curves on training (light brown) and validation (brown) datasets, together with sand-predicting model loss curves on training (light gold) and validation (gold) datasets.

Funding

This work was partially supported by i) the ITALian RADioactivity (ITALRAD) project of the National Institute of Nuclear Physics (INFN), ii) the "TOMato for baby food: Monitoring heavY metal in production chain" project - CUP C66B20001120008, iii) FAR 2020–2021 of the University of Ferrara, iv) the Bando per il finanziamento della ricerca scientifica "Fondo per l'Incentivazione alla Ricerca" (FIR) - anno 2021 of the University of Ferrara, v) the "Early warning system per la PrEvenzione della diffusione della flavescenza doRata BAsato sul monitoraggio multiparametriCo airborne delle COlture vinicole' (PERBACCO) project - CUP E47F23000030002, vi) the ICSC – Centro Nazionale di Ricerca in High Performance Computing, Big Data and Quantum Computing, funded by European Union – NextGenerationEU - CUP F77G22000120006, and vii) the "GeoneUtrinos: mESSengers of the Earth's interior' (GUESS) project - CUP F53D23001280006.

CRediT authorship contribution statement

Andrea Maino: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Software, Validation, Visualization, Writing – original draft. **Matteo Alberi:** Data curation, Formal analysis, Investigation, Software, Writing – review & editing. **Alessio Barbagli:** Investigation, Visualization, Writing – review & editing. **Enrico Chiarelli:** Data curation, Formal analysis, Investigation, Software, Writing – review & editing. **Tommaso Colonna:** Funding acquisition, Project administration, Writing – review & editing. **Michele Franceschi:** Investigation, Visualization, Writing – review & editing. **Fabio Gallorini:** Investigation, Visualization, Writing – review & editing. **Enrico Guastaldi:** Funding acquisition, Project administration, Writing – review & editing. **Nicola Lopane:** Investigation, Visualization, Writing – review & editing. **Fabio Mantovani:** Conceptualization, Funding acquisition, Investigation, Methodology, Project administration, Supervision, Validation, Visualization, Writing – original draft. **Dario Petrone:** Investigation, Visualization, Writing – review & editing. **Silvio Pierini:** Investigation, Visualization, Writing – review & editing. **Kassandra Giulia Cristina Raptis:** Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Visualization, Writing – original draft. **Virginia Strati:** Conceptualization, Investigation, Methodology, Project administration, Supervision, Validation, Writing – original draft. **Gerti Xhixha:** Conceptualization, Writing – review & editing.

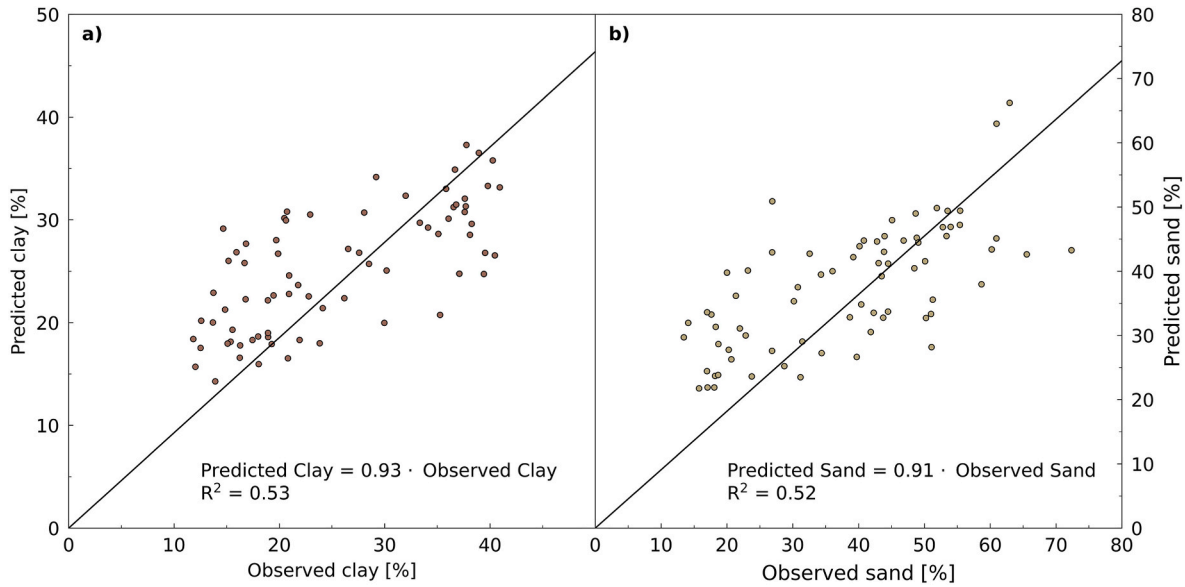

Fig. 3. Correlation plots between a) test dataset predicted and observed clay values and b) test dataset predicted and observed sand values. For both plots, the equation of the regression line (solid line) is reported together with the coefficient of determination (R^2).

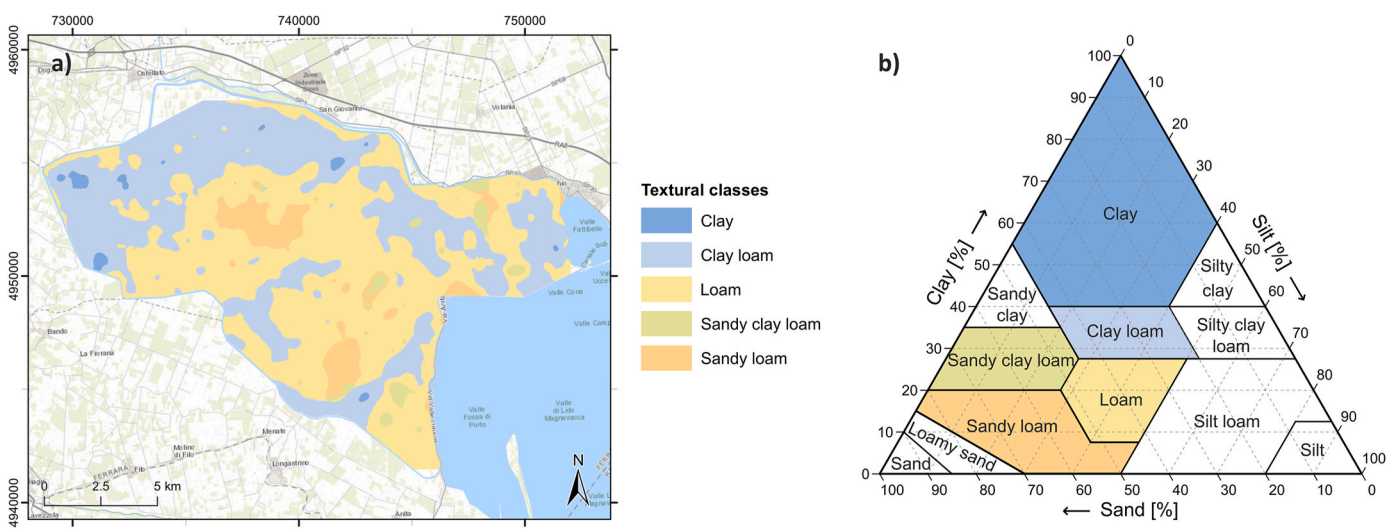

Fig. 4. a) soil textural map of the Mezzano Lowland obtained from the clay and sand content estimates from the respective models. Spatial resolution: 20 m × 20 m. Cartographic reference system: WGS 84, UTM Zone 32 N. b) The USDA soil textural triangle highlighting only the classes found in the map. The color used for each class in the triangle corresponds to the same class on the map.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The dataset, models, Python script and Anaconda environment files can be downloaded from: https://www.fe.infn.it/radioactivity/AI/.

Acknowledgments

The authors would like to thank Adriano Facchini, Giovanni Fiorentini, Maurizio Marcialis, Nicola Martini, Silvia Piccioli and Andrea Serafini for the valuable discussions, and Maurino Antongiovanni, Paolo Baldassarri, Stefano Calderoni, Claudio Pagotto, and Olivio Vassalli for the logistical assistance provided.

References

Azadnia, R., Jahanbakhshi, A., Rashidi, S., Bazyar, P.J.M., 2022. Developing an automated monitoring system for fast and accurate prediction of soil texture using an image-based deep learning network and machine vision system, 190, 110669.

Baldoncini, M., Alberi, M., Bottardi, C., Minty, B., Raptis, K.G.C., Strati, V., Mantovani, F., 2018. Airborne gamma-ray spectroscopy for modeling cosmic radiation and effective dose in the lower atmosphere. IEEE Trans. Geosci. Rem. Sens. 56, 823–834.

Baldoncini, M., Albéri, M., Bottardi, C., Minty, B., Raptis, K.G.C., Strati, V., Mantovani, F., 2017. Exploring atmospheric radon with airborne gamma-ray spectroscopy. Atmos. Environ. 170, 259–268.

Elbaalawy, A.M., Benbi, D.K., Benipal, D.S., 2016. Potassium forms in relation to clay mineralogy and other soil properties in different agro-ecological sub-regions of northern India. Agric. Res. J. 53.

Heggemann, T., Welp, G., Amelung, W., Angst, G., Franz, S.O., Koszinski, S., Schmidt, K., Pätzold, S.J.S., Research, T., 2017. Proximal Gamma-Ray Spectrometry for Site-independent in Situ Prediction of Soil Texture on Ten Heterogeneous Fields in Germany Using Support Vector Machines, vol. 168, pp. 99–109.

Keskar, N.S., Mudigere, D., Nocedal, J., Smelyanskiy, M., Tang, P.T.P., 2016. On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima.
Le, D.K., Yoon, J.Y., Design, 2023. A hybrid CFD–Deep learning methodology for improving the accuracy of pressure drop prediction in cyclone separators. Chem. Eng. Res. 190, 296–311.
Mahmood, H.S., Hoogmoed, W.B., van Henten, E.J., 2013. Proximal gamma-ray spectroscopy to predict soil properties using windows and full-spectrum analysis methods. Sensors (Basel) 13, 16263–16280.
Maino, A., Alberi, M., Anceschi, E., Chiarelli, E., Cicala, L., Colonna, T., De Cesare, M., Guastaldi, E., Lopane, N., Mantovani, F., 2022. Airborne radiometric surveys and machine learning algorithms for revealing soil texture. Rem. Sens. 14, 3814.
Mohri, M., Rostamizadeh, A., Talwalkar, A., 2018. Foundations of Machine Learning. MIT press.
Nair, V., Hinton, G.E., 2010. Rectified linear units improve restricted Boltzmann machines. In: Proceedings of the 27 Th International Conference on Machine Learning.
Petersen, H.W.T., Attia al Hagrey, S., Rabbel, W., 2012. Characterization of some Middle European soil textures by gamma-spectrometry. J. Plant Nutr. Soil Sci. 175, 651–660.
Priori, S., Bianconi, N., Costantini, E.A.J.G., 2014. Can γ-radiometrics predict soil textural data and stoniness in different parent materials? A comparison of two machine-. Learn.Methods 226, 354–364.
Sharma, S., Sharma, S., Athaiya, A.J.T.D.S., 2017. Activation Functions in Neural Networks, pp. 310–316.
Spadoni, M., Voltaggio, M., 2013. Contribution of gamma ground spectrometry to the textural characterization and mapping of floodplain sediments. J. Geochem. Explor. 125, 20–33.
Tarocco, P.S.F., Ungaro, F., 2015. Note illustrative della Carta della tessitura dei suoli della pianura emiliano-romagnola strato 0-30 cm, scala, vol. 1, 50000.
Van Der Klooster, E., Van Egmond, F.M., Sonneveld, M.P.W., 2011. Mapping soil clay contents in Dutch marine districts using gamma-ray spectrometry. Eur. J. Soil Sci. 62, 743–753.
Wang, H., Shang, S., Wang, D., He, X., Feng, K., Zhu, H., 2022. Plant disease detection and classification method based on the optimized lightweight YOLOv5 model. Agriculture 12, 931.
Wu, W., Li, A.-D., He, X.-H., Ma, R., Liu, H.-B., Lv, J.-K.J.C., Agriculture, E.i., 2018. A comparison of support vector machines, artificial neural network and classification tree for identifying soil texture classes in southwest China, 144, 86–93.
Yang, L., Shami, A.J.N., 2020. On hyperparameter optimization of machine learning algorithms. Theor. Pract. 415, 295–316.
Zhang, M., Shi, W., Xu, Z.J.H., Sciences, E.S., 2020. Systematic comparison of five machine-learning models in classification and interpolation of soil particle size fractions using different transformed data, 24, 2505–2526.
Zhao, Z., Feng, W., Xiao, J., Liu, X., Pan, S., Liang, Z.J.A., 2022. Rapid and accurate prediction of soil texture using an image-based deep learning autoencoder convolutional neural network random forest (DLAC-CNN-RF) algorithm, 12, 3063.