World Academy of Science, Engineering and Technology
International Journal of Mathematical and Computational Sciences
Vol:18, No:10, 2024

Enhancing Spatial Interpolation: A Multi-Layer Inverse Distance Weighting Model for Complex Regression and Classification Tasks in Spatial Data Analysis

Yakin Hajlaoui, Richard Labib, Jean-François Plante, Michel Gamache

Abstract—This study presents the Multi-Layer Inverse Distance Weighting Model (ML-IDW), inspired by the mathematical formulation of both multi-layer neural networks (ML-NNs) and Inverse Distance Weighting model (IDW). ML-IDW leverages ML-NNs' processing capabilities, characterized by compositions of learnable non-linear functions applied to input features, and incorporates IDW's ability to learn anisotropic spatial dependencies, presenting a promising solution for nonlinear spatial interpolation and learning from complex spatial data. We employ gradient descent and backpropagation to train ML-IDW. The performance of the proposed model is compared against conventional spatial interpolation models such as Kriging and standard IDW on regression and classification tasks using simulated spatial datasets of varying complexity. Our results highlight the efficacy of ML-IDW, particularly in handling complex spatial dataset, exhibiting lower mean square error in regression and higher F1 score in classification.

Keywords—Deep Learning, Multi-Layer Neural Networks, Gradient Descent, Spatial Interpolation, Inverse Distance Weighting.

I. Introduction

DEEP learning has emerged as a transformative tool for discerning intricate structures within complex datasets, proving its efficacy across a spectrum of tasks from regression to classification in diverse domains such as natural language processing and computer vision [1]. By leveraging the inherent compositional properties of learnable non-linear functions applied to inputs, multi-layer neural networks (ML-NNs) have demonstrated remarkable prowess in pattern recognition [2]. This capability is underpinned by parameter optimization through gradient descent, facilitated by the efficient computation of gradients using the back-propagation algorithm [3].

Nevertheless, feedforward ML-NNs, like many machine learning models, often assume that data observations are independently sampled from a given distribution, thereby overlooking the spatial dependency structures inherent in the data. Conversely, spatial interpolation models like Inverse Distance Weighting (IDW) excel at capturing such dependencies, but they frequently lack the flexibility and the learning capacity of ML-NNs.

Yakin Hajlaoui, Richard Labib, and Michel Gamache are with Department of Mathematics and Industrial Engineering, Polytechnique Montréal, Montréal, H3T 1J4, Quebec, Canada (e-mail: yakin-2.hajlaoui@polymtl.ca, richard.labib@polymtl.ca, michel.gamache@polymtl.ca).

Jean-François Plante is with Department of Decision Sciences, HEC Montréal, Montréal, H3T 2A7, Quebec, Canada (e-mail: jfplante@hec.ca).

Numerous studies have compared machine learning against spatial interpolation (SI) models [3]–[6], showcasing the advantages of ML models in accurately depicting nonlinear relationships [4]. However, discussions on the limitations of ML models in modeling spatial relationships have led to a preference for spatial interpolations in scenarios where data exhibit significant spatial correlations [7]. Additionally, by employing linear interpolation, SI models furnish smooth predictions that risk overlooking abrupt changes and non-linearities in the data [8]. In response to these constraints, many researchers have proposed the fusion of both approaches [9]–[14].

For instance, some studies have combined SI with artificial neural networks (ANNs), utilizing SI to augment data by interpolating in locations lacking observational features, subsequently leveraging ANN that uses the provided feature maps as dependent variables for prediction [12]. Other studies combined Regression Kriging with machine learning models like Support Vector Machine (SVM) where the ML model predicts the drift, while Kriging performs spatial interpolation on the residuals, treating them as corrections to the predicted values [9], [15]. This approach has also been extended to classification tasks, where Simplical Indicator Kriging is combined with SVM for classification purposes [15], [16].

Despite the effectiveness of many ML models, they still lack the representational capacity, computational efficiency, and flexibility of deep neural networks, in which model complexity is customizable relative to the datasets. Proposed architectures such as Residual Neural Networks have been introduced to safely use deeper models without risking the vanishing gradient problem [17]. To our knowledge, few studies are conducted to present a spatial interpolation model that has similar benefits. We aim to address this gap through our study by presenting such a model: the ML-IDW model. This model applies compositions of learnable non-linear functions to input data and uses IDW interpolation with learnable parameters. Trained using gradient descent and back-propagation, it is compatible to be stacked with other ANNs models or architectures. This would allow us to leverage the power of deep learning while simultaneously accounting for spatial dependency structures.

II. Theory and Methodology

A. Conventional IDW: A Brief Background

Let the data consist of targets t_i, $i = 1, ..., N$, which are spatially correlated and located in 2D spatial coordinates $\mathbf{s}_i = [x_i, y_i]^T$. We denote by $\mathcal{T}$ the set of indices in the test data and by $\overline{\mathcal{T}}$ the set of indices in the training data. Conventional Inverse Distance Weighting (IDW) performs a prediction at an unsampled location s_j, $j \in \mathcal{T}$, by computing an average weighted sum of targets t_i, $i \in \overline{\mathcal{T}}$:

$$\widehat{t}_j = \sum_{i \in \overline{\mathcal{T}}} w_{ji} t_i, \tag{1}$$

where $w_{ji} = d_a(\mathbf{s}_i, \mathbf{s}_j)^{-p} / \sum_k d_a(\mathbf{s}_k, \mathbf{s}_j)^{-p}$ are the IDW weights that depend on the power parameter p. The term $d_a(\mathbf{s}_i, \mathbf{s}_j) = \sqrt{(x_i - x_j)^2/\ell_x^2 + (y_i - y_j)^2/\ell_y^2}$ represents the Euclidean distance accounting for anisotropy in the spatial correlation, where ℓ_x is called the range, or length-scale, of the x principal direction, and similarly for ℓ_y the principal direction of y [18], [19]. In the conventional approach, the set of parameters to be learned, denoted $\theta = \{p, \ell_x, \ell_y\}$, are derived using the Leave One Out Cross Validation (LOOCV) algorithm [20], [21]. This algorithm involves conducting a grid search G of possible parameter candidates in order to choose the set of parameters that minimizes the LOOCV error: $E^{LOOCV} = (1/|\overline{\mathcal{T}}|) \sum_{i \in \overline{\mathcal{T}}} (\widehat{t}_i - t_i)^2$. This error consists of the average of the squared errors between every target t_i in the training data and its IDW estimate $\widehat{t}_i$, computed using IDW performed on all the training data, leaving the i^{th} observation out.

This training approach has a time complexity scaling as $O(|\overline{\mathcal{T}}|^2 * |G|)$ and a space complexity of $O(|\overline{\mathcal{T}}| + |G|)$, where $|\overline{\mathcal{T}}|$ represents the size of the training dataset without the omitted observation, and $|G| = m^3$ denotes the size of the grid, with m being the number of parameter possibilities. This method faces challenges in scalability due to the potentially large size of the grid $|G|$. As m increases to explore more parameter possibilities, both time and space complexity grow significantly. While a higher m might lead to improved accuracy by considering a finer-grained search over parameter space, it also compromises scalability.

B. Gradient Based IDW

To address the challenges posed by conventional Inverse Distance Weighting (IDW) interpolation, we presented a Gradient-Based IDW (GB-IDW) approach in a previous study. In GB-IDW, we leveraged the matrix formulation of an IDW model to compute the leave-one-out cross-validation (LOOCV) error in a single forward pass: $E^{LOOCV}(\theta) = 1/|\overline{\mathcal{T}}| \left\|\widehat{\mathbf{t}} - \mathbf{t}\right\|^2 = 1/|\overline{\mathcal{T}}| \left\|\mathbf{W}(\theta)\mathbf{t} - \mathbf{t}\right\|^2$.

Here, $\mathbf{t}$ and $\widehat{\mathbf{t}}$ are vectors containing the true values and the IDW predicted values, respectively, for $i \in \overline{\mathcal{T}}$. $\mathbf{W}$ is a matrix with zeros on the diagonal and GB-IDW weights w_{ji} elsewhere. $\|.\|$ denotes the Euclidean norm.

We exploited the differentiability of this error with respect to the parameter set θ to apply gradient descent and backpropagation for minimizing $E^{LOOCV}(\theta)$. The time complexity of this training approach scales as $O(|\overline{\mathcal{T}}|^2 \times |\mathcal{I}|)$, where $|\mathcal{I}|$ denotes the number of training iterations. This is because the backpropagation algorithm scales linearly with the number of parameters ($O(3 \times |\mathcal{I}|)$) [22], making the time complexity dominated by $O(|\overline{\mathcal{T}}|^2 \times |\mathcal{I}|)$. This represents a significant reduction in time complexity compared to conventional methods, especially when $|\mathcal{I}| \ll |G|$.

Additionally, the space complexity is dominated by $O(|\overline{\mathcal{T}}|^2)$. While time and space complexity could be a drawback for large data sizes, leveraging the power of Graphics Processing Units (GPUs) in parallelism and memory efficiency can mitigate this challenge significantly.

By exploiting the optimized matrix operations inherent in GPUs, we can achieve substantial improvements in computational efficiency [23]–[25]. GPUs are designed to handle parallel operations efficiently, particularly in deep learning tasks, making them well-suited for accelerating the training process of GB-IDW. This can effectively address the scalability challenges encountered in large-scale interpolation tasks.

C. Single Layer IDW

GB-IDW, trained with gradient descent and backpropagation, is compatible with neural networks and can be stacked with an Artificial Neural Network (ANN) for further processing of the targets $\mathbf{t}$. We introduce a Single Layer IDW (SL-IDW) by adding a single processing layer to the vector $\mathbf{t}$ before using GB-IDW for interpolation.

Let X denote dependent covariates that may contains other dependent features along with the spatial locations $\mathbf{S} = [\mathbf{s}_i]$. We employ a simple feedforward Neural Network, denoted as NN, which takes X as input and produces two vectors $\mathbf{c} = [c_i]$ and $\mathbf{b} = [b_i]$ representing the slope and intercept for $\mathbf{t}$, respectively. A simple linear regression is performed on $\mathbf{t}$ to compute the hidden vector $\mathbf{h} = \mathbf{c} \odot \mathbf{t} + \mathbf{b}$ using these two parameters, where $\odot$ denotes elementwise product and $+$ denotes elementwise sum ($\mathbf{c} \odot \mathbf{t} + \mathbf{b} = [c_i t_i + b_i]$). We then apply an activation function $g^{(1)}$ to introduce nonlinearity and compute the hidden layer $\mathbf{a}^{(1)} = g^{(1)}(\mathbf{h})$.

Spatial interpolation is then applied to $\mathbf{a}^{(1)}$ by computing $\widehat{\mathbf{t}} = \mathbf{W}(\theta)\mathbf{a}^{(1)}$. The LOOCV error $E^{LOOCV}(\theta, \varphi) = 1/|\overline{\mathcal{T}}| \left\|\mathbf{W}(\theta)\mathbf{a}^{(\mathbf{1})} - \mathbf{t}\right\|^2$ remains differentiable with respect to all parameters of the model, comprising $\theta = \{p, \ell_x, \ell_y\}$ as GB-IDW parameters, and φ, as the parameters of the feedforward neural network NN.

In summary, the interpolation equation of the Single Layer IDW is:

$$\widehat{\mathbf{t}} = \mathbf{W}(\theta)\mathbf{a}^{(1)},$$

$$\mathbf{a}^{(1)} = g^{(1)}(\mathbf{h}),$$

$$\mathbf{h} = \mathbf{c} \odot \mathbf{t} + \mathbf{b}, \quad \mathbf{c}, \mathbf{b} = NN(X).$$

D. Mutli-Layer IDW for Regression

Multi-Layer IDW (ML-IDW) for regression extends Single Layer IDW (SL-IDW) by incorporating multiple processing layers for $\mathbf{t}$. Each layer k takes $\mathbf{a}^{(k-1)}$ from the previous layer as input and computes the hidden vector $\mathbf{h}^{(k)} = \mathbf{c}^{(k)} \odot \mathbf{a}^{(k-1)} + \mathbf{b}^{(k)}$, where $\mathbf{c}^{(k)}, \mathbf{b}^{(k)} = NN^{(k)}(X)$ represent the intercept and slope provided by the feedforward neural network for layer k, $NN^{(k)}$. An activation function $g^{(k)}$ is then applied to compute $\mathbf{a}^{(k)} = g^{(k)}(\mathbf{h}^{(k)})$.

Inspired by Residual Neural Networks (ResNet), which learn a nonlinear residual to facilitate identity map learning and address vanishing gradients [17], we present a skipping connection layer such that $\mathbf{a}^{(k)} = g^{(k)}(\mathbf{h}^{(k)}) + \mathbf{h}^{(k-2)}$. Spatial interpolation is performed on the output of the last layer L: $\widehat{\mathbf{t}} = \mathbf{W}(\theta)\mathbf{a}^{(L)}$.

In summary, the interpolation equation of ML-IDW is:

$$\begin{aligned}
\widehat{\mathbf{t}} &= \mathbf{W}(\theta)\mathbf{a}^{(L)}, \\
\mathbf{a}^{(k)} &= g^{(k)}(\mathbf{h}^{(k)}) + \mathbf{h}^{(k-2)}, \quad k = 2, ..., L, \\
\mathbf{a}^{(1)} &= g^{(1)}(\mathbf{h}^{(1)}), \\
\mathbf{a}^{(0)} &= \mathbf{t}, \\
\mathbf{h}^{(k)} &= \mathbf{c}^{(k)} \odot \mathbf{a}^{(k-1)} + \mathbf{b}^{(k)}, \\
\mathbf{c}^{(k)}, \mathbf{b}^{(k)} &= NN^{(k)}(X), \quad k = 1, ..., L.
\end{aligned}$$

It is worth noting that with this setup, the model will also be able to learn through spatial interpolation on the values $\mathbf{t}$, any unknown mapping $\mathbf{u}$ of $\mathbf{t}$: $\mathbf{u} = f(\mathbf{t})$. A special case could be explored for classification in which we transform $\mathbf{t}$ into categorical variables $\mathbf{u}$ using the function $u_i = f(t_i) = 1$ if $t_i \geq \alpha$, $u_i = f(t_i) = 0$ otherwise, where α is a threshold. Such a setup will make our model competitive with nonlinear spatial interpolation methods such as Simplicial Indicator Kriging.

E. Mutli-Layer IDW for Classification

In ML-IDW for classification, we aim to predict categorical spatial targets $u_i \in \{1, ..., C\}$, where C is the number of possible categories obtained through categorizing the targets t_i. We extend ML-IDW for regression to perform classification tasks by introducing a feedforward neural network FN and applying a softmax function to each value $\widehat{t}_i$ to obtain the vector of probabilities $\mathbf{p}_i = [p_i^{(1)}, ..., p_i^{(C)}]$ containing the probability of each category. The classification rule in this case is assigning the class characterized by the highest probability: $\hat{u}_i = \text{argmax}(\mathbf{p}_i)$. The cross-entropy loss function is utilized as the minimization criterion: $L(\mathbf{U}, \mathbf{P}) = 1/|\overline{\mathcal{T}}| \sum_{i=1}^{|\overline{\mathcal{T}}|} \sum_{l=1}^{C} u_i^{(l)} \ln(p_i^{(l)})$. Here, $\mathbf{U} = [u_i^{(l)}]$ contains the ground truth labels (1 if sample i belongs to class l, 0 otherwise), and $\mathbf{P} = [p_i^{(l)}]$ contains the predicted probabilities that sample i belongs to class l. In summary the interpolation equation of ML-IDW in case of classification is:

$$\begin{aligned}
\mathbf{P} &= \text{softmax}(FN(\mathbf{W}(\theta)\mathbf{a}^{(L)})), \\
\mathbf{a}^{(k)} &= g^{(k)}(\mathbf{h}^{(k)}) + \mathbf{h}^{(k-2)}, \quad k = 2, ..., L, \\
\mathbf{a}^{(1)} &= g^{(1)}(\mathbf{h}^{(1)}), \\
\mathbf{a}^{(0)} &= \mathbf{t}, \\
\mathbf{h}^{(k)} &= \mathbf{c}^{(k)} \odot \mathbf{a}^{(k-1)} + \mathbf{b}^{(k)}, \quad \mathbf{c}^{(k)}, \\
\mathbf{b}^{(k)} &= NN^{(k)}(X), \quad k = 1, ..., L.
\end{aligned}$$

III. Experiments and Results

A. Data Description

To assess our models, we employed the Turning Band method [26] for geostatistical simulation to generate 2D spatial data across varying sizes, ranging from 100 to 10,000 data points. We utilized two types of variogram models for simulation to provide spatially correlated datapoints: a simple model composed of a Gaussian variogram and a nested model combining Gaussian and exponential variograms. Across all datasets, the partial sill was set at 20 with a nugget of 4. Anisotropy was introduced by establishing the range along the y-axis as one third of the range along the x-axis ($\ell_y = \ell_x/3$). We introduced complexity variations by considering different range values ℓ_x: 20, 50, and 80. A smaller range implies a more localized correlation, introducing local abrupt changes, thus increasing the complexity of the data. Table I summarizes the simulated datasets.

These datasets serve both regression and classification purposes. For classification tasks, categorical targets ($\mathbf{u}$) were derived from continuous targets ($\mathbf{t}$) by setting $u_i = 1$ if $t_i \geq \alpha$, $u_i = 0$ otherwise. Here, the threshold α is set as the median of the observations $\mathbf{t}$ to ensure data balance.

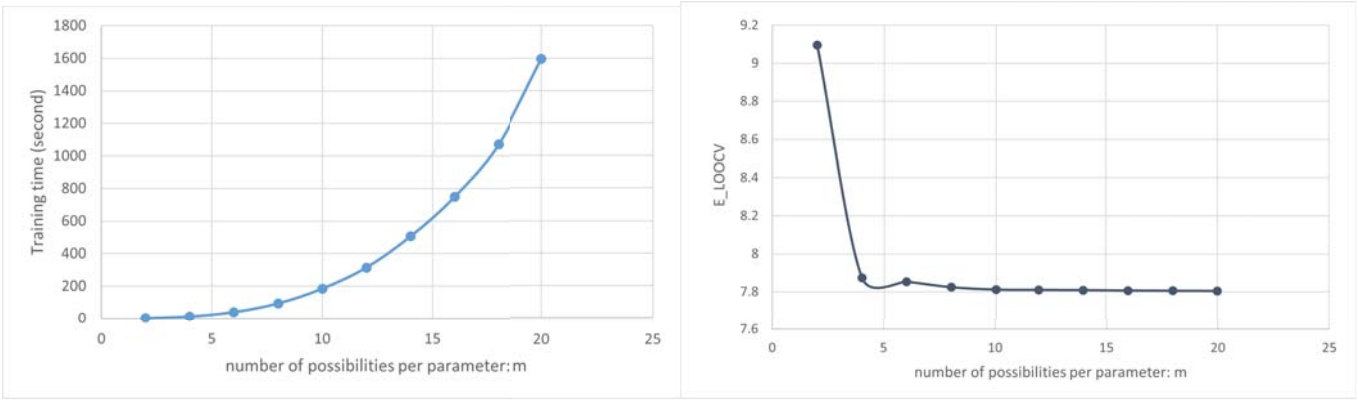
B. Training Efficiency

In this section, we explore the training efficiency of GB-IDW and ML-IDW using gradient descent and backpropagation, comparing them to the conventional approach of training an IDW model via a search in a discretized grid. As previously noted, time complexity can be a significant challenge for conventional IDW, it varies polynomially with the number of possibilities per parameter m, and exponentially with the number of parameters. Fig. 1 illustrates different values of m versus the training time and the training error E^{LOOCV}, utilizing dataset 6 comprising 1000 data points. As anticipated, employing a finer grid leads to lower training error; however, the training time increases significantly.

Table II presents the E^{LOOCV} values for IDW, GB-IDW, and ML-IDW, along with the number of gradient descent iterations, the grid size for conventional IDW, and the total training time. GB-IDW achieves a comparable training error in a shorter duration, while ML-IDW with 5 layers achieves a lower training error in a very short time. For larger datasets, such as those with 10,000 data points, the time complexity will significantly increase, making it exceedingly challenging to train conventional IDW within a reasonable timeframe while considering all three parameters $\{p, \ell_x, \ell_y\}$.

TABLE I
GEOSTATISTICAL CHARACTERISTICS OF THE SIMULATED DATASETS

Datasets	Variogram model	Size	Nugget	Partial Sill	Range	Anisotropy
dataset 1	Gaussian	100	4	20	20	$\ell_y = \ell_x/3$
dataset 2	Gaussian	100	4	20	50	$\ell_y = \ell_x/3$
dataset 3	Gaussian	100	4	20	80	$\ell_y = \ell_x/3$
dataset 4	Gaussian	1000	4	20	20	$\ell_y = \ell_x/3$
dataset 5	Gaussian	1000	4	20	50	$\ell_y = \ell_x/3$
dataset 6	Gaussian	1000	4	20	80	$\ell_y = \ell_x/3$
dataset 7	Gaussian	10000	4	20	20	$\ell_y = \ell_x/3$
dataset 8	Gaussian	10000	4	20	50	$\ell_y = \ell_x/3$
dataset 9	Gaussian	10000	4	20	80	$\ell_y = \ell_x/3$
dataset 10	Gaussian + Exponential	100	4	20	20	$\ell_y = \ell_x/3$
dataset 11	Gaussian + Exponential	100	4	20	50	$\ell_y = \ell_x/3$
dataset 12	Gaussian + Exponential	100	4	20	80	$\ell_y = \ell_x/3$
dataset 13	Gaussian + Exponential	1000	4	20	20	$\ell_y = \ell_x/3$
dataset 14	Gaussian + Exponential	1000	4	20	50	$\ell_y = \ell_x/3$
dataset 15	Gaussian + Exponential	1000	4	20	80	$\ell_y = \ell_x/3$
dataset 16	Gaussian + Exponential	10000	4	20	20	$\ell_y = \ell_x/3$
dataset 17	Gaussian + Exponential	10000	4	20	50	$\ell_y = \ell_x/3$
dataset 18	Gaussian + Exponential	10000	4	20	80	$\ell_y = \ell_x/3$



(a) (b)

Fig. 1 (a) Training time of conventional IDW vs. number of possibilities per parameter; (b) E^{LOOCV} error vs. number of possibilities per parameter for conventional IDW

TABLE II
MODELS' EFFICIENCY

Model	Number of iterations $\lvert\mathcal{I}\rvert$ or grid size $\lvert G\rvert$	E^{LOOCV}	Training time (seconds)
Conventional IDW	$\lvert G\rvert = 20^3 = 8000$	7.804	1594.709
GB-IDW	$\lvert\mathcal{I}\rvert$ =1000	7.804	**12.49**
ML-IDW (5 layers)	$\lvert\mathcal{I}\rvert$ =1000	**6.680**	136.234

C. Results on Regression Tasks

To investigate model performance on regression tasks, we partitioned each dataset so that seventy percent of the data served as training data, while the remaining thirty percent served as test data. We began with datasets containing 10,000 data points and monitored the prediction performance of Ordinary Kriging, Grad-IDW, and ML-IDW with 5 layers with Relu activation function (referred to as Deep-IDW) while varying complexity (specifically, varying the range).

Fig. 2 depicts, from the upper left figure to bottom right figure, the simulated data (dataset 7 characterized by high complexity with a range of 20), the training data, the test data, predictions from Ordinary Kriging, predictions from Grad-IDW, and predictions from Deep-IDW.

On datasets characterized by high complexity, Deep-IDW exhibits higher resolution, with predicted values closer to the ground truth values of the test data. Grad-IDW demonstrates lesser resolution. However, Ordinary Kriging appears very smooth and is unable to capture abrupt changes in the data.

Figs. 3a and 3b illustrate the Mean Square Error (MSE) for the three models concerning complexity respectively for datasets with simple and nested variogram models. It is evident that Deep-IDW excels in higher complexity scenarios (range=20), exhibiting lower MSE_{test}. However, the results are remarkably comparable in medium complexity situations (range=50), with Grad-IDW demonstrating the lowest test error in Fig 3 (a). On low complexity datasets (range = 80), Ordinary Kriging performs slightly better.

The advantage of ML-IDW in this context lies in its customised nature: we can adjust the model complexity by adding or removing layers according to data complexity. It is

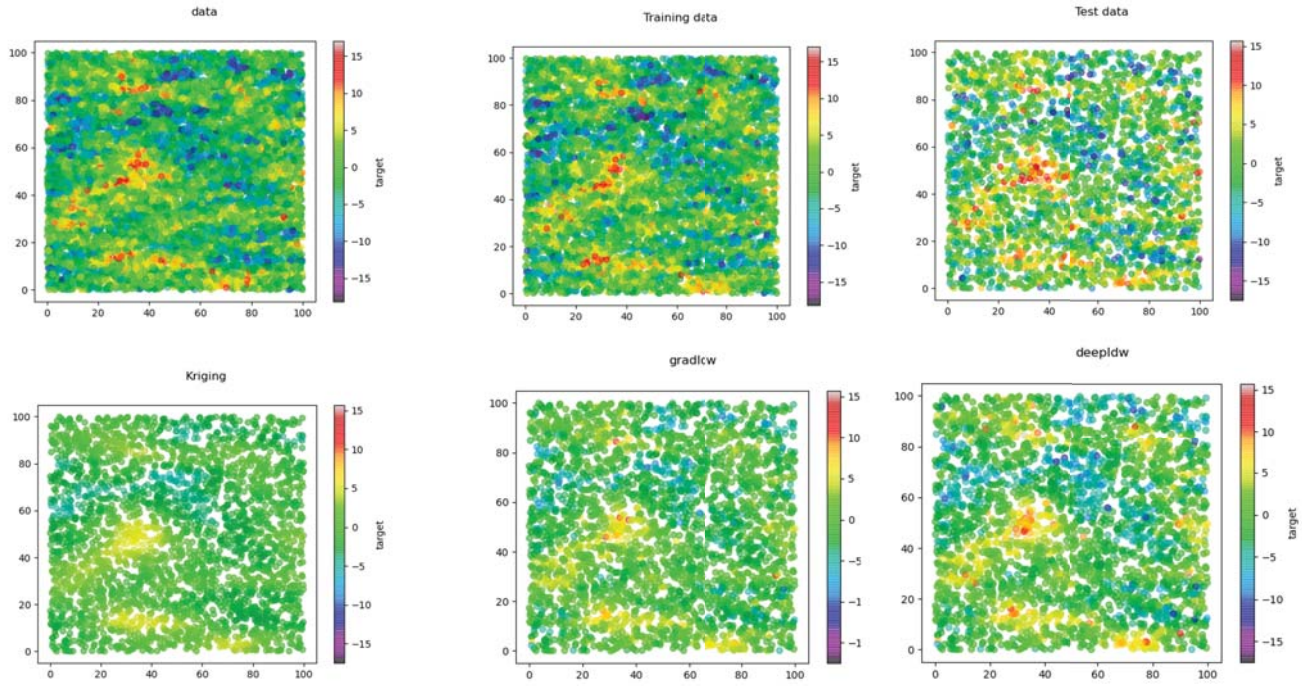
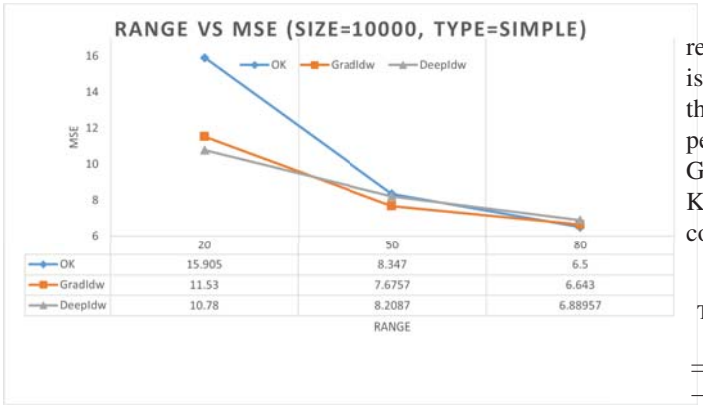
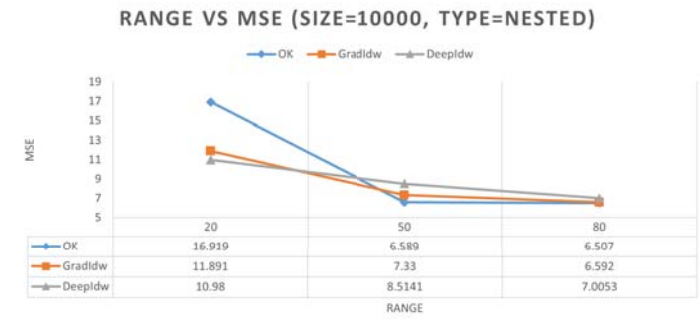


Fig. 2 Sets and predictions

worth noting that Grad-IDW is a variant of ML-IDW with no layers, making it a simpler alternative in certain scenarios.



(a)



(b)

Fig. 3 Mean Square Error for OK, Grad-IDW, and DeepIDW on 10,000-point dataset: (a) Simple variogram; (b) Nested variogram

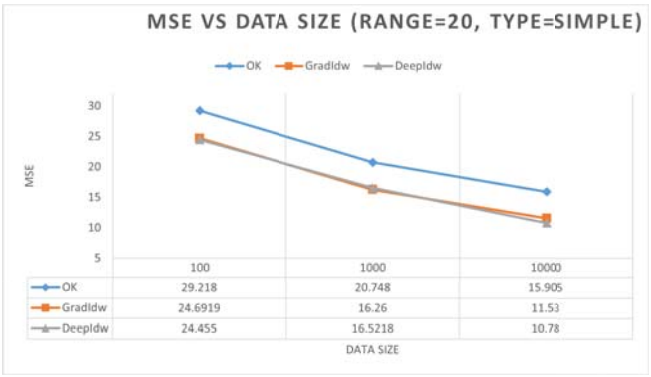
Figs. 4a and 4b display the Mean Square Error (MSE) test results for the three models across different data sizes, specifically focusing on high complexity (range=20) and low complexity (range=80) scenarios, respectively.

Table III shows the MSE for all dataset. Observing the results while fixing complexity and varying data size, it is evident that the performance of all models declines as the data size decreases. Notably, Deep-IDW exhibits better performance than OK, and comaparable performance with Grad-IDW in high complexity scenarios, while Ordinary Kriging exhibits slightly superior performance in low complexity situations.

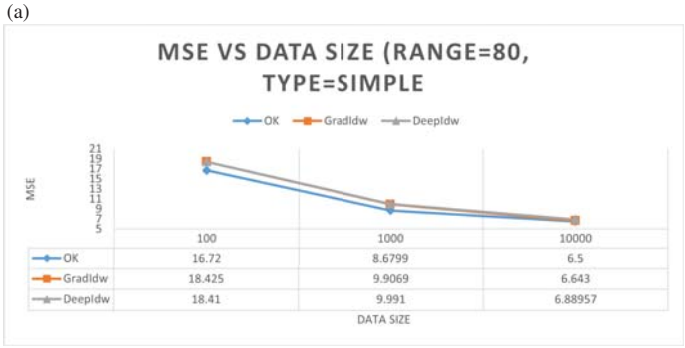
TABLE III
THE MEAN SQUARE ERROR TEST (MSE) FOR EACH MODEL AND EACH DATASET

Datasets	OK	Grad-IDW	Deep-IDW (5 layers ML-IDW)
dataset 1	29.218	24.6919	**24.455**
dataset 2	**23.64497**	26.115776	26.46
dataset 3	**16.72**	18.425	18.41
dataset 4	20.748	**16.26**	16.5218
dataset 5	**10.339**	12.695	12.443
dataset 6	**8.6799**	9.9069	9.991
dataset 7	15.905	11.53	**10.78**
dataset 8	8.347	**7.6757**	8.2087
dataset 9	6.5	**6.643**	6.88957
dataset 10	39.19	**36.139**	36.54
dataset 11	**22.829**	25.23613	25.397
dataset 12	**15.705**	16.7	16.63
dataset 13	23.047	**17.45**	18.89
dataset 14	**9.6436**	13.549	14.19
dataset 15	**8.133**	12.393	12.7872
dataset 16	16.919	11.891	**10.98**
dataset 17	6.589	**7.33**	8.5141
dataset 18	**6.507**	6.592	7.0053

Note: Lower MSE indicates closer predictions to the ground truth. The **bold** values are the lowest for each dataset.



(a)



(b)

Fig. 4 Mean Square Error for OK, Grad-IDW, and DeepIDW vs. data size on a simple variogram model: (a) High complexity; (b) Low complexity

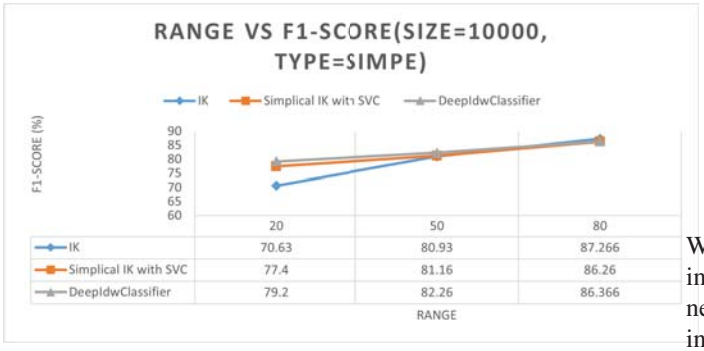


Fig. 5 F1 score for IK, Simplical IK with SVC, and DeepIDW vs. data complexity on a 10,000-point simple variogram dataset

D. Results on Classification

In the classification task, we compare a ML-IDW classifier with 5 layers, referred to as Deep-IDW-Classifier, against Indicator Kriging (IK) and Simplical Indicator Kriging (SIK) combined with Support Vector Machine for Classification (SVC). We employ the F1 score as the performance criterion (higher values indicating better performance) to evaluate the prediction capabilities on the test data. Similar to the regression task, we divide the data into training and test sets.

Fig. 6 illustrates the training and test data, along with the predictions of the three models for dataset 7. It is evident that Deep-IDW can capture nonlinearities and abrupt changes in the data. Simplical Indicator Kriging combined with SVC also performs well, while Indicator Kriging struggles to mimic the behavior of the test set.

Table IV summarizes the results for all datasets. We observe that Deep-IDW-Classifier excels in most cases and its results where at least comparable with the best classifier.

Fig. 5 presents the F1 score for each model with respect to complexity. In this case, Deep-IDW-Classifier outperforms Simplical Indicator Kriging with SVC and Indicator Kriging in high and medium complexity scenarios, whereas Indicator Kriging performs better in low complexity scenarios.

TABLE IV
THE F1 SCORE IN PERCENTAGE FOR EACH MODEL AND EACH DATASET

Datasets	IK	SIK with SVC	Deep-IDW-Classifier
dataset 1	50	56.66	**60**
dataset 2	**76.66**	53.33	**76.66**
dataset 3	**63.33**	50	**63.33**
dataset 4	69.66	67	**70.33**
dataset 5	**78.66**	73.66	75.33
dataset 6	**81**	78.33	79
dataset 7	70.63	77.4	**79.2**
dataset 8	80.93	81.16	**82.26**
dataset 9	**87.266**	86.26	86.366
dataset 10	**80**	56.66	70
dataset 11	66.66	53.33	43.33
dataset 12	**70**	60	40
dataset 13	50.33	66	**69.66**
dataset 14	**76.66**	76	76.33
dataset 15	**83**	81.66	82.33
dataset 16	74.5	79.6	**80.2**
dataset 17	85.56	85.2	**85.66**
dataset 18	87.766	89.23	**89.9**

Note: Higher F1 score indicates better classification. The **bold** values are the highest for each dataset.

IV. CONCLUSION

In this study, we proposed the Multi-Layer Inverse Distance Weighting Model (ML-IDW), a spatial interpolation model inspired by both the formulation of multi-layer neural networks—as compositions of learnable functions—and the inverse distance weighting (IDW) approach, which utilizes an average weighted sum of inverse distance functions. The goal was to enhance spatial interpolation capabilities, enabling the model to learn from complex data without compromising the inherent spatial dependency structure.

Our results demonstrated that ML-IDW outperforms conventional geostatistical approaches, such as Ordinary Kriging, Indicator Kriging, and Simplical Indicator Kriging, in both regression and classification tasks when data complexity is high. In scenarios with lower complexity, the results were comparable. Notably, ML-IDW is a customizable model, allowing for the addition or removal of layers according to the data's complexity.

Furthermore, ML-IDW exhibited higher training efficiency compared to traditional IDW training methods. By utilizing gradient descent and backpropagation for training, the model is compatible with other neural network architectures.

For future work, we plan to stack ML-IDW neural network projection models to learn spatial dependencies on latent representations.

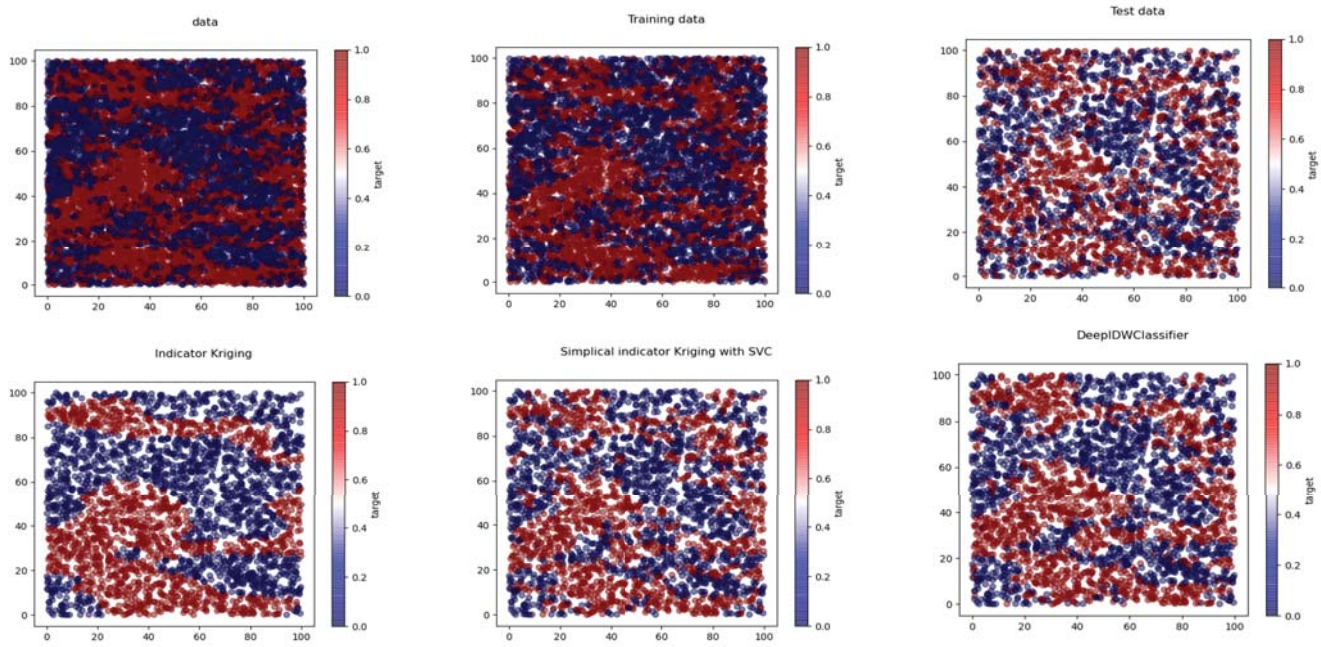


Fig. 6 Sets for classification and predictions

REFERENCES

[1] J. Heaton, "Ian goodfellow, yoshua bengio, and aaron courville: Deep learning: The mit press, 2016, 800 pp, isbn: 0262035618," *Genetic programming and evolvable machines*, vol. 19, no. 1, pp. 305–307, 2018.

[2] C. M. Bishop and N. M. Nasrabadi, *Pattern recognition and machine learning*. Springer, 2006, vol. 4, no. 4.

[3] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[4] J. Kim, Y. Lee, M.-H. Lee, and S.-Y. Hong, "A comparative study of machine learning and spatial interpolation methods for predicting house prices," *Sustainability*, vol. 14, no. 15, p. 9056, 2022.

[5] I. Chahrour and J. Wells, "Comparing machine learning and interpolation methods for loop-level calculations," *SciPost Physics*, vol. 12, no. 6, p. 187, 2022.

[6] D. Radočaj, M. Jurišić, R. Župan, and O. Antonić, "Spatial prediction of heavy metal soil contents in continental croatia comparing machine learning and spatial interpolation methods," *Geodetski list*, vol. 74, no. 4, pp. 357–372, 2020.

[7] G. T. Nwaila, S. E. Zhang, J. E. Bourdeau, H. E. Frimmel, and Y. Ghorbani, "Spatial interpolation using machine learning: from patterns and regularities to block models," *Natural Resources Research*, vol. 33, no. 1, pp. 129–161, 2024.

[8] J. K. Yamamoto, "Correcting the smoothing effect of ordinary kriging estimates," *Mathematical geology*, vol. 37, pp. 69–94, 2005.

[9] X. Li, Y. Ao, S. Guo, and L. Zhu, "Combining regression kriging with machine learning mapping for spatial variable estimation," *IEEE Geoscience and Remote Sensing Letters*, vol. 17, no. 1, pp. 27–31, 2019.

[10] Z. Huang, H. Wang, and R. Zhang, "An improved kriging interpolation technique based on svm and its recovery experiment in oceanic missing data," 2012.

[11] R. Karami and P. Afzal, "Estimation of elemental distributions by combining artificial neural network and inverse distance weighted (idw) based on lithogeochemical data in kahang porphry deposit, central iran," *Universal Journal of Geoscience*, vol. 3, no. 2, pp. 59–65, 2015.

[12] T. Tunçay, P. Alaboz, O. Dengiz, and O. Başkan, "Application of regression kriging and machine learning methods to estimate soil moisture constants in a semi-arid terrestrial area," *Computers and Electronics in Agriculture*, vol. 212, p. 108118, 2023.

[13] D. Cho, J. Im, and S. Jung, "A new statistical downscaling approach for short-term forecasting of summer air temperatures through a fusion of deep learning and spatial interpolation," *Quarterly Journal of the Royal Meteorological Society*, vol. 150, no. 760, pp. 1222–1242, 2024.

[14] J. Tan, X. Xie, J. Zuo, X. Xing, B. Liu, Q. Xia, and Y. Zhang, "Coupling random forest and inverse distance weighting to generate climate surfaces of precipitation and temperature with multiple-covariates," *Journal of Hydrology*, vol. 598, p. 126270, 2021.

[15] B. S. Murphy, "Pykrige: development of a kriging toolkit for python," in *AGU fall meeting abstracts*, vol. 2014, 2014, pp. H51K–0753.

[16] R. Tolosana-Delgado, V. Pawlowsky-Glahn, and J. Egozcue, "Simplicial indicator kriging," *Journal of China University of Geosciences*, vol. 19, no. 1, pp. 65–71, 2008.

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[18] D. Shepard, "A two-dimensional interpolation function for irregularly-spaced data," in *Proceedings of the 1968 23rd ACM national conference*, 1968, pp. 517–524.

[19] N. S.-N. Lam, "Spatial interpolation methods: a review," *The American Cartographer*, vol. 10, no. 2, pp. 129–150, 1983.

[20] O. Babak and C. V. Deutsch, "Statistical approach to inverse distance interpolation," *Stochastic Environmental Research and Risk Assessment*, vol. 23, pp. 543–553, 2009.

[21] Z.-N. Liu, X.-Y. Yu, L.-F. Jia, Y.-S. Wang, Y.-C. Song, and H.-D. Meng, "The influence of distance weight on the inverse distance weighted method for ore-grade estimation," *Scientific Reports*, vol. 11, no. 1, p. 2689, 2021.

[22] D. E. Rumelhart, R. Durbin, R. Golden, and Y. Chauvin, "Backpropagation: The basic theory," in *Backpropagation*. Psychology Press, 2013, pp. 1–34.

[23] Q. Guan, P. C. Kyriakidis, and M. F. Goodchild, "A parallel computing approach to fast geostatistical areal interpolation," *International Journal of Geographical Information Science*, vol. 25, no. 8, pp. 1241–1267, 2011.

[24] P. J. van Oosterom, H. Ploeger, A. Mansourian, S. Scheider, R. Lemmens, and B. Van Loenen, "Proceedings-the 26th agile international conference on geographic information science spatial data for design: Preface," in *26th AGILE Conference on Geographic Information Science, AGILE 2023: Spatial data for design*. Copernicus, 2023.

[25] F. Huang, S. Bu, J. Tao, and X. Tan, "Opencl implementation of a parallel universal kriging algorithm for massive spatial data interpolation on heterogeneous systems," *ISPRS international journal of geo-information*, vol. 5, no. 6, p. 96, 2016.

[26] J.-P. C. Pierre Delfiner, *Geostatistics: Modeling Spatial Uncertainty*. WILEY, Mar. 2012. [Online].

Available: https://www.ebook.de/de/product/15356462/pierre_delfiner_jean_paul_chiles_geostatistics_modeling_spatial_uncertainty.html