Multidimensional Fuzzy Transforms with Inverse Distance Weighted Interpolation for Data Regression

Barbara Cardone and Ferdinando Di Martino *

Posted Date: 28 January 2025

doi: 10.20944/preprints202501.2127.v1

Keywords: F-transform; Multidimensional F-transform; regression mode; IDW; data interpolation

Article

Multidimensional Fuzzy Transforms with Inverse Distance Weighted interpolation for data regression

Barbara Cardone [1] and Ferdinando Di Martino [1,2,*]

[1] University of Naples Federico II, Department of Architecture, Via Toledo 402, 80134 Napoli, Italy; b. cardone@unina.it
[2] Center for Interdepartmental Research "Alberto Calza Bini", University of Naples Federico II, Via Toledo 402, 80134 Napoli, Italy
* Correspondence: Correspondence fdimarti@unina.it; Tel.: 0039-081-2538904, Fax: 0039-081-2538909

Abstract: The main limitation of the Multidimensional Fuzzy Transform algorithm applied in regression analysis is the fact that it cannot be used if the data are not dense enough with respect to the fuzzy partitions; in these cases, less fine fuzzy partitions must be used to the detriment of the accuracy of the results. In this work, a variation of the Multidimensional Fuzzy Transform regression algorithm is proposed in which the Inverse Distance Weighted interpolation method is applied as a data augmentation algorithm to satisfy the criterion of sufficient data density with respect to the fuzzy partitions. A preprocessing phase determines the optimal values of the parameters to be set in the algorithm's execution. Comparative tests with other well-known regression methods are performed on five regression datasets extracted from the UCI Machine Learning repository. The results show that the proposed method provides the best performance, in terms of regression error reductions.

Keywords: F-transform; Multidimensional F-transform; regression model; IDW; data interpolation

1. Introduction

Fuzzy Transform (for short F-transform) [1] is a technique that approximates a continuous function by a finite vector of components, determined from a set of points where the value of the function is known. F-transform was applied in its bi-dimensional form in image analysis for coding/decoding images [2,3,4,5].

The Multidimensional F-transform (for short, MF-transform) has been used as a ML technique in various data analysis applications. It was applied in [6,7] to detect dependencies among features in datasets. A comparison between the MF-transform and radial basis function neural networks is given in [8].

In [9,10] MF-transform is applied in forecasting analysis. In [11] MF-transform is used as a data classification method. In [12,13] MF-transform is applied in time series analysis; in [14] the MF-transform is encapsulated in a Long Short Term Memory architecture to reduce the size of datasets in fake news detection applications. An extensive description of the MF-transform-based techniques used in image and data analysis is given in [15,16,17].

The critical point of MF-transform is given by a constraint called *constraint of sufficient data density with respect to the fuzzy partitions*, , a constraint that requires that at least one datapoint belongs with non-null membership degree to every combination of fuzzy sets of the fuzzy partitions of the feature domains.

For example, assuming that the data points are composed of two features, and let $\{A_{11}, A_{12}, \ldots, A_{1n}\}$ and $\{A_{21}, A_{22}, \ldots, A_{2n}\}$ be two fuzzy partition of the domains of the first and the second feature, whose fuzzy sets are called basic functions, the constraint of

sufficient data density with respect to the fuzzy partitions requires that for every combination of basic functions A_{1h}, A_{2k} there exists at least one data point $p_j = (x_{1j}, x_{2j})$ such that $A_{1h}(x_{1j}) \cdot A_{2k}(x_{2j}) \neq 0$.

This limitation is present, usually, in ML methods. When an ML algorithm is applied to data that is significantly different from the training data, ML models can fail or produce inaccurate results, running into an overfitting problem. This happens because the algorithm adapts to the observed data, which are generally not sufficient to completely cover the domains of values of the variables. The model provides accurate performance only within the subdomains in which the training data fluctuates but cannot adapt to new data.

To address this problem in [10,11,12] is applied an iterative process for determining the finest combination of fuzzy partitions of the features domains that satisfies the sufficiently density constraint. This approach has the advantage of allowing the MF-transform to be used as a ML algorithm, however the finest combination of fuzzy partitions determined may not be sufficient to guarantee optimal accuracy of the results.

In [18] an out-of-sample variation of the F-transform is proposed to extend the discrete counterparts of the F-transform to the continuous case in order to adapt the use of the F-transform to new data. This method allows to construct a one-dimensional F-transform that models the continuous behavior of signals, but it is not applicable for data analysis.

The first order MF-transform is proposed in [19] as a classification algorithm to improve the accuracy of the MF-transform; in this work, the Principal Component Analysis method is applied to reduce the number of features, and the iterative process in [10] is applied to find the finest combination of fuzzy partitions satisfying the sufficiently density constraint. The authors show that this method improves the classification accuracy of MF-transform; However, it cannot address the overfitting problem and cannot fit new data outside the observed data domain.

In this work, a variation of the MF-transform method, called IDWF-transform, is proposed, in which a data augmentation algorithm based on the Inverse Distance Weighted interpolation method (for short, IDW) [20] is applied to ensure sufficient data density of data points concerning the combination of fuzzy partitions.

IDW is a K-nearest neighbor multivariate interpolation method applied to the scattered set of points. The assigned values to unknown points are calculated with a weighted average of the values available at the K nearest known points, where the weight is given by the inverse of the distances between the unknown point and the known point, raised to a power value p and the Euclidean metric is applied for calculating the distances. For p = 0, the average becomes a weighted average and the distance from the unknown point does not affect the estimate of the interpolated value, as p increases, the closer known points have a greater influence than the farther ones.

Compared to traditional regression methods using MF-transform, IDWF-Transform can be executed even if the constraint of sufficient density is not respected; in fact, it adds interpolated data in the regions of the feature space where the absence of data points causes the violation of the constraint.

Furthermore, it provides better regression accuracy than that provided by MF-transform, since it allows the use of fine fuzzy partitions, so as to reduce the regression error.

The paper is structured as follows. In section 2, preliminary concepts are briefly discussed and the MF-transform regression method and the IDW interpolation algorithm are described. The proposed method is discussed in depth in section 3. In section 4, comparative results of tests performed on well-known regression datasets are shown and discussed. Concluding remarks are given in section 5.

2. Preliminaries

In this section the MF-transform data regression method and the IDW interpolator are briefly described.

2.1. F-transform concepts

Let X = [a,b] be a close interval of *R*; in (Perfilieva & Haldeeva, 2001, Perfilieva, 2006) was introduced the following definition of fuzzy partition of X.

Let $x_0, x_1, x_2, \ldots, x_n$ be a set of n+1 fixed points, called *nodes*, in [a,b] such that n ³ 3 and $a = x_1 < x_2 < \ldots < x_n = b$. We say that fuzzy sets $A_1,\ldots,A_n : [a,b] \to [0,1]$ form a *generalized fuzzy partition* of [a,b], if for each k = 1,2,…, n, the following constraints hold:

1. $A_k(x) = 0 \ \forall x \notin (x_{k-1}, x_{k+1})$ (locality)
2. $A_k(x) > 0 \ \forall x \in (x_{k-1}, x_{k+1})$ and $A_k(x_k) = 1$ (positivity)
3. Ak is continuous in $\left[x_k - h_k^{'}, x_k + h_k^{"}\right]$ (continuity)
4. Ak is strictly decreasing in (xk-1, xk) and strictly increasing in (xk, xk+1)
5. $\sum_{k=1}^{n} A_k(x) = 1 \ \forall x \in [a,b]$ (Ruspini condition).

The membership functions $\{A_1,\ldots,A_n\}$are called basic functions. If the nodes $x_1,\ldots, x_n$ are equidistant, the fuzzy partition $\{A_1,\ldots,A_n\}$ is called h-uniform fuzzy partition of [a,b], where h = (b-a)/(n+1) is the distance between two consecutive nodes.

For a h-uniform fuzzy partition the following additional properties hold:

6. $A_k(x_k - x) = A_k(x_k + x) \ \forall x \in [0,h]$
7. $A_k(x) = A_{k-1}(x-h)$ and $A_{k-1}(x) = A_k(x+h) \ \forall x \in [x_k, x_{k+1}]$

An h-uniform fuzzy partition can be generated (see [22]) by a even function A_0: [-1,1] → [0,1], which is continuous, positive in (-1,1) and null on boundaries {-1,1}. The function A_0 is called *generating function* of the h-uniform fuzzy partition. The following expression represents an arbitrary basic function from an h-uniform generalized fuzzy partition

$$A_k(t) = \begin{cases} A_0\left(\frac{x - x_k}{h}\right) & x \in [x_k - h, x_k + h] \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

As example of generating function, we consider the triangular function:

$$A_0(t) = \begin{cases} 0 & t < 1 \\ t+1 & -1 \le t \le 0 \\ 1-t & 0 \le t \le 1 \\ 0 & t > b \end{cases} \quad (2)$$

The basic functions of the generated h-uniform fuzzy partition are given by:

$$A_k(x) = \begin{cases} 0 & x < x_k - h \\ \frac{x - x_k}{h} + 1 & x_k - h \le x \le x_k \\ 1 - \frac{x - x_k}{h} & x_k \le x \le x_k + h \\ 0 & x > x_k + h \end{cases} \quad k = 1,\ldots,n \quad (3)$$

Let $\{A_1, A_2, \ldots, A_n\}$ be a fuzzy partition of [a,b] and *f(x)* be a continuous function on [a,b]. Thus, we can consider the following real numbers for i = 1, …, n:

$$F_k = \frac{\int_a^b f(x)A_k(x)dx}{\int_a^b A_k(x)dx} \qquad k = 1,\ldots,n \quad (4)$$

The n-tuple $[F_1, F_2, \ldots, F_n]$ is called the *fuzzy transform* of f with respect to $\{A_1, A_2, \ldots, A_n\}$. The F_k are called *components* of the F-transform.

In many cases we only know that the function *f* assumes determined values in a set of m points $p_1,\ldots,p_m \in [a,b]$.

We assume that the set P of these nodes is *sufficiently dense with respect to the fixed fuzzy partition*, i.e. for each k = 1,…,n there exists an index $j \in \{1,\ldots,m\}$ such that $A_k(p_j) > 0$. Then we can define the n-tuple $[F_1, F_2,\ldots,F_n]$ as the *discrete F-transform* of *f* with respect to $\{A_1, A_2, \ldots, A_n\}$, where each F_k is given by

$$F_k = \frac{\sum_{j=1}^{m} f(p_j)A_k(p_j)}{\sum_{j=1}^{m} A_k(p_j)} \qquad k = 1,\ldots,n \quad (5)$$

Is called *discrete inverse F-transform* of *f* with respect to $\{A_1, A_2, \ldots, A_n\}$ the following function defined in the same points $p_1,\ldots, p_m$ of [a,b]:

$$f_{F,n}(x) = \sum_{k=1}^{n} F_k A_k(x) \quad x[a, b] \tag{6}$$

We have the following approximation theorem (Perfilieva, 2006 - Theorem 5):

Theorem 1. Let $f(x)$ be a function assigned on a set P of points $p_1,\ldots, p_m$ of [a,b]. Then for every $\varepsilon > 0$, there exist an integer $n(\varepsilon)$ and a related fuzzy partition $\{A_1, A_2, \ldots, A_{n(\varepsilon)}\}$ of [a,b] such that P is sufficiently dense with respect to $\{A_1, A_2, \ldots, A_{n(\varepsilon)}\}$ and for every $p_j \in [a, b]$, j = 1,…,m,

$$\left| f(x) - f_{F,n(\varepsilon)}(x) \right| < \varepsilon \tag{7}$$

Compliance with the constraint of sufficient density with respect to the partition is essential to ensure the existence of the discrete F-transform of *f*. In fact, if exists a fuzzy set A_k of the fuzzy partition for which $\forall j \in \{1,\ldots,m\}$ $A_k(p_j) = 0$, then the (1.34) cannot be applied to calculate the F-transform component F_k. The meaning of this is that the fuzzy partition of the domain [a, b] is too fine with respect to the dataset of the measures of the function *f*.

2.2. Multidimensional F-transform

Let $f : X \subseteq R^n \rightarrow Y \subseteq R$ be a continuous s-dimensional function defined in a closed interval $X = [a_1,b_1]\times[a_2,b_2]\times\ldots\times[a_s,b_s] \subseteq R^s$ and known in a discrete set of N points $P=\{(p_{11}, p_{12}, \ldots, p_{1s}), (p_{21}, p_{22}, \ldots, p_{2s}),\ldots,(p_{N1}, p_{N2}, \ldots,p_{Ns})\}$.

For each k= 1,…,s let $x_{k1}, x_{k2}, \ldots, x_{knk}$ with $n_k \geq 2$ be a set of n_k nodes of $[a_k,b_k]$, where $x_{k1} = a_k < x_{k2} <\ldots< x_{knk} = b_k$. We suppose that the set of n_k nodes are equidistant, and the distance between two consecutive nodes is $h_k = (b_k-a_k)/(n_k+1)$. Then, the h-uniform fuzzy partition of $[a_k,b_k]$ $\{A_{k1},\ldots,A_{knk}\}$ forms a set of basic functions of $[a_k,b_k]$.

We say that the set $P=\{(p_{11}, p_{12}, \ldots, p_{1s}), (p_{21}, p_{22}, \ldots, p_{2s}),\ldots,(p_{N1}, p_{N2}, \ldots,p_{Ns})\}$ is sufficiently dense w.r.t. the set of fuzzy partition $\{A_{11}A_{12}\ldots A_{1n_1}\},\ldots, \{A_{k1}A_{k2}\ldots A_{kn_k}\},\ldots, \{A_{s1}A_{s2}\ldots A_{sn_n}\}$ if for each combination $A_{1h_1}A_{2h_2}\ldots A_{sh_s}$ exists at least a point $p_j = (p_{j1}, p_{j2}, \ldots, p_{js}) \in P$, such that $A_{1h_1}(p_{j1}) \cdot A_{2h_2}(p_{j2}) \cdot\ldots\cdot A_{sh_s}(p_{js}) > 0$. In this case we can define the direct multidimensional F-transform of *f* with the $(h_1,h_2,\ldots,h_s)$*th* component $F_{h_1h_2\ldots h_s}$ given by

$$F_{h_1h_2\ldots h_s} = \frac{\sum_{j=1}^{N} f(p_{j1}, p_{j2}, \ldots p_{js}) \cdot A_{1h_1}(p_{j1}) \cdot A_{2h_2}(p_{j2}) \cdot\ldots\cdot A_{sh_s}(p_{js})}{\sum_{j=1}^{N} A_{1h_1}(p_{j1}) \cdot A_{2h_2}(p_{j2}) \cdot\ldots\cdot A_{sh_s}(p_{js})} \tag{8}$$

The multidimensional inverse F-transform, calculated in the point p_j, is given by:

$$f^{F}_{n_1n_2\ldots n_s}(p_{j1}, p_{j2}, \ldots, p_{js}) = \sum_{h_1=1}^{n_1}\sum_{h_2=1}^{n_2}\ldots\sum_{h_s=1}^{n_s} F_{h_1h_2\ldots h_s} \cdot A_{1h_1}(p_{j1}) \cdot\ldots\cdot A_{sh_s}(p_{js}) \tag{9}$$

It approximates the function *f* in the point p_j.

The multidimensional F-transform (for short, MF-transform) can be applied in regression analysis and classification. It is used in [7] to detect dependencies among features in datasets, using numeric encoding to transform categorical data. In [9,10,12] MF-transform is applied in forecasting analysis. In [11] MF-transform is applied as a data classification method.

The critical point of MF-transform is the constraint of sufficient data density with respect to the fuzzy partitions. In fact, if exists a combination of basic function $\{A_{1h_1}, A_{2h_2,}\ldots, A_{sh_s}\}$ such as for every point $(p_{j1}, p_{j2}, \ldots,p_{js})$, with j = 1,…,N, $A_{1h_1}(p_{j1}) \cdot A_{2h_2}(p_{j2}) \cdot\ldots\cdot A_{sh_s}(p_{js}) = 0$, then the direct MF-transform (8) cannot be used.

This limitation is present in machine learning methods. When an ML algorithm is applied to data that is significantly different from the training data, ML models can fail or produce inaccurate results. In these cases, we speak of data overfitting problems.

2.3. IDW interpolation method

IDW [20,21,22] is a K-nearest neighbor interpolation method in which the value in an unknown point is given by a weighted average of the values of K nearest known points. It is one of the most popular methods used for geospatial data interpolation and is usually applied to highly variable data. IDW is a computationally fast interpolation method; compared to polynomial or spline-based interpolation methods, it is more efficient when the data has strong variations over short distances [23].

The basic principle of IDW is that data points that are progressively further away from the unknown point influence the calculated value much less than those that are closer to the node and this influence is measured by considering the Euclidean distance between the data point and the unknown point.

Formally, if x is the position of a unknown point in the n-dimensional space of the feature domains, then the interpolated value of a function *f* in the point x is given by:

$$f(\mathrm{x}) = \frac{\sum_{\mathrm{j}=1}^{\mathrm{K}} f(\mathrm{x_j})\mathrm{w}(\mathrm{x_j})}{\sum_{\mathrm{j}=1}^{\mathrm{K}} \mathrm{w}(\mathrm{x_j})} \tag{10}$$

where $x_1, x_2, \ldots, x_K$ are the K sample points closest to x and the weight w(xj) is given by the formula:

$$w(\mathrm{x_j}) = \frac{1}{\mathrm{d}(\mathrm{x}, \mathrm{x_j})^p} \qquad \mathrm{j}=1,2,\ldots,\mathrm{K} \tag{11}$$

In the equation (11) $\mathrm{d}(\mathrm{x}, \mathrm{x_j})$ is the Euclidean distance between the j*th* sample point and the unknown point x, and p is a positive power parameter that controls the smoothness of interpolation. For p = 0 the weighted average becomes a simple average. The higher the value of p the higher the contribution of the closest points compared to the most distant ones.

The (10) can be obtained minimizing the following function expressing the deviation between the expected values and sample values [21,22]:

$$\mathrm{d}(\mathrm{x}) = \frac{1}{2} \sum_{\mathrm{j}=1}^{\mathrm{K}} \frac{1}{\mathrm{d}(\mathrm{x}, \mathrm{x_j})^p} \left(f(\mathrm{x}) - f(\mathrm{x_j})\right)^2 \tag{12}$$

The two parameters to be set in (10) are K and p. Cross validation techniques can be used to set them [24].

3. The IDWMF-transform method

To apply MF-Transform for data regression and classification a data augmentation method based on the IDW algorithm is applied in order to address the sufficient data point density problem.

To focus on the problem, in Fig.1 an example of insufficient data density with respect to the fuzzy partitions is shown for data points with two input features, x_1 and x_2.

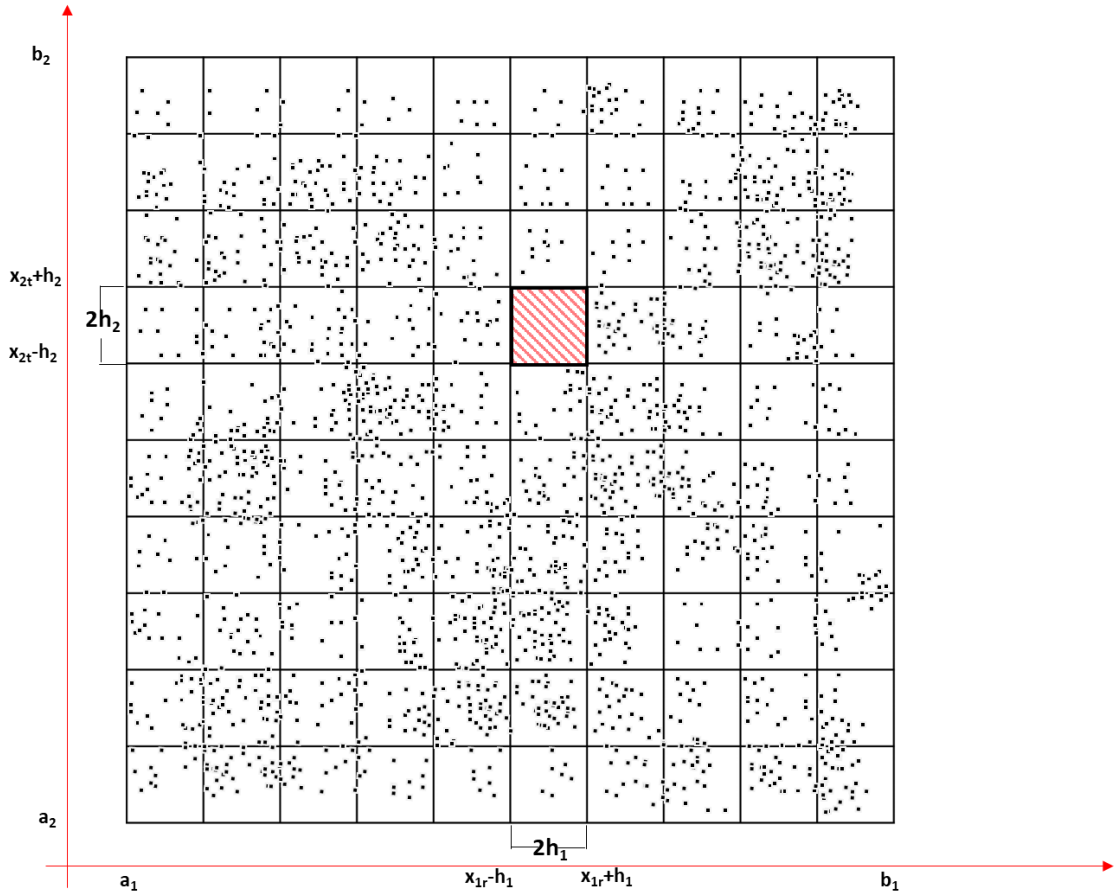


Figure 1. Example of problem of insufficient data density with respect to the fuzzy partitions.

In the example shown in Fig.1 there are two basic functions, A_{1r} and A_{2t} such that, for each data point $p_j = (p_{j1}, p_{j2})$ $A_{1r}(p_{j1}) \cdot A_{2t}(p_{j2}) = 0$. Then, the data are not sufficiently dense with respect to the fuzzy partitions. The fuzzy partitions of the domains of the two input variables are too fine and fuzzy partitions coarser grained must be set.

To solve this problem in [7] a technique has been proposed that allows to optimize the selection of the cardinality of the fuzzy partitions while respecting the constraint of sufficient data density.

The low diagram in Fig.2 schematizes this technique.

 doi:10.20944/preprints202501.2127.v1

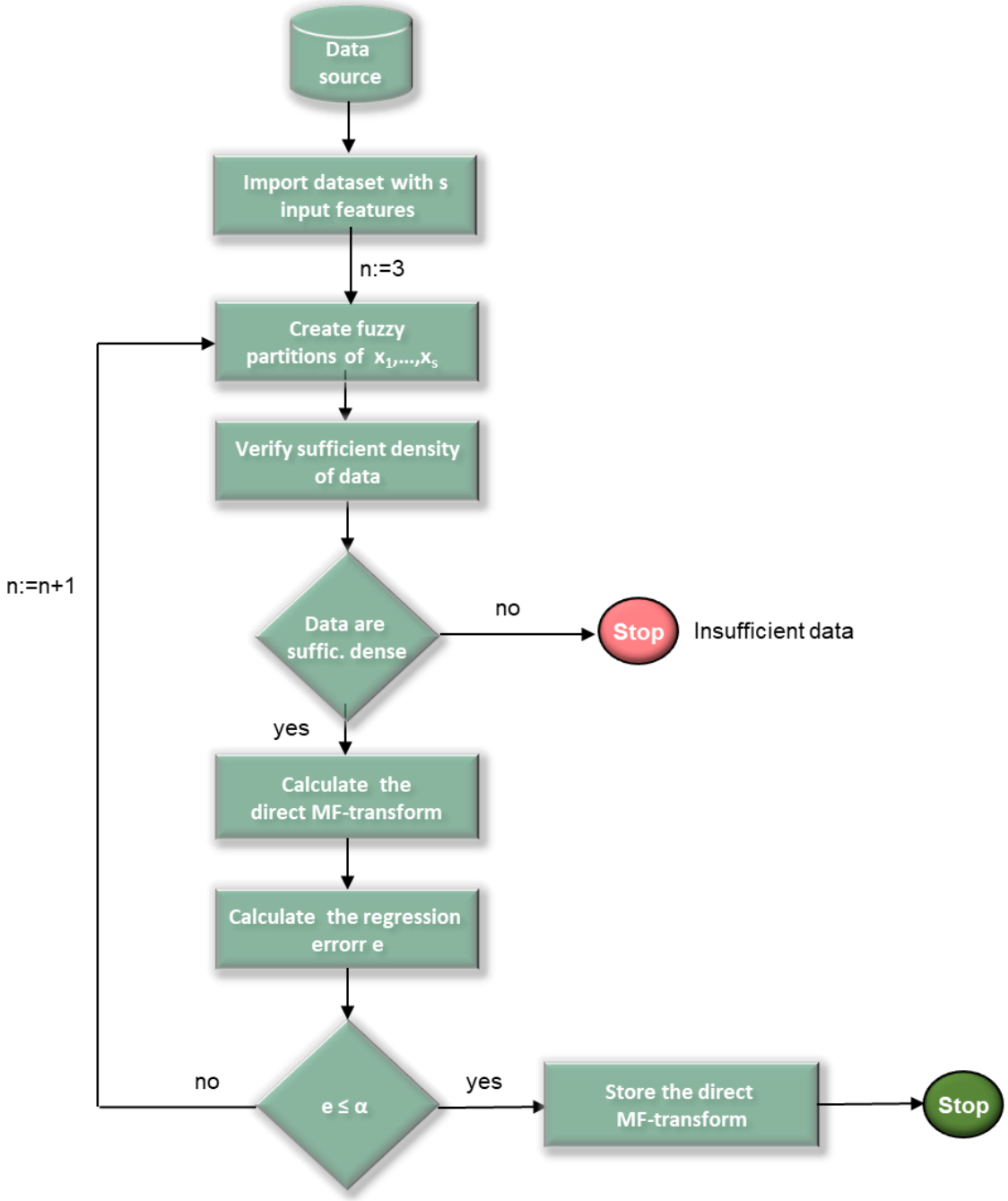


Figure 2. Flow diagram MF-transform regression method [7].

Initially, the lowest cardinality of the fuzzy partitions is set at n = 3. After creating the fuzzy partitions, the sufficient data density is analyzed; if the data are not sufficiently dense with respect to the fuzzy partitions, the algorithm end, with the error message that the data is not dense enough and the regression model based on the MF-transform cannot be used. Otherwise, the direct MF-transform components and a regression measure is used to verify the accuracy of the model. If the regression error is not higher than a threshold α, then the direct MF-transform components are stored, and the algorithm ends. Otherwise, finer fuzzy partitions are generated (n = n + 1) and the process is iterated.

The limitation of this method is that, in cases where the choice of the α threshold implies the need for fine partitions, the data points may not be dense enough with respect to the fuzzy partitions; in these cases, the use of coarser grained fuzzy partitions would require a reduction of the threshold and, therefore, a lower accuracy of the model results.

To address this criticality, this work proposes a variation of the linear regression model based on MF-transforms [7], in which the IDW data interpolation algorithm is used when the data is not dense enough with respect to the fuzzy partitions. The flow diagram in Fig. 3 schematizes the IDWMF-Transform method.

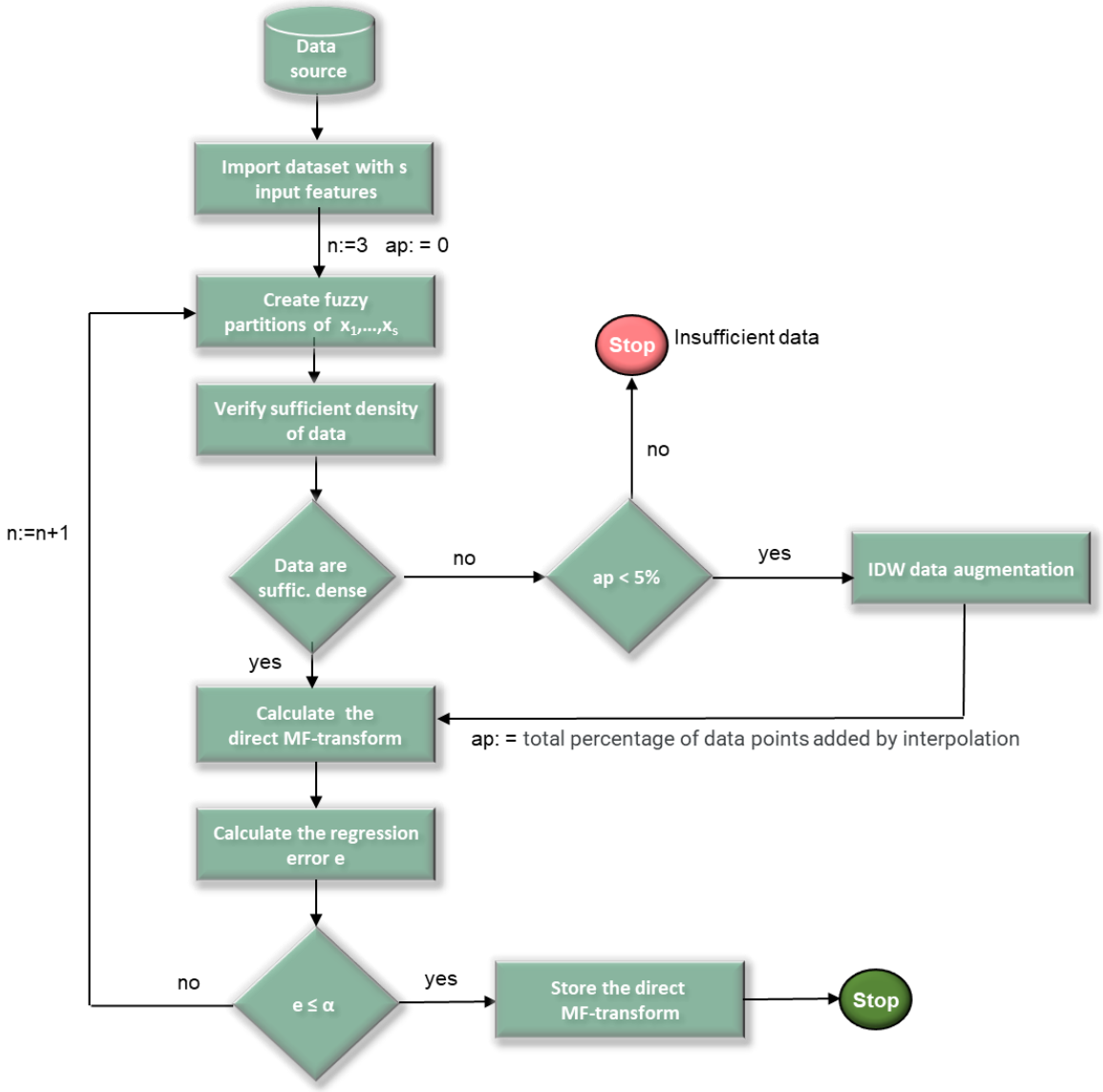


Figure 3. Flow diagram of the IDWMF-transform regression method.

Initially, in addition to setting the size of the fuzzy partitions n to 3, the variable ap, which contains the percentage of new data points, is initialized to 0. If the density of data points is insufficient and the number of added data points does not exceed 5% of the overall cardinality of the data points, then the IDW interpolation method is used to insert new data points in the feature space regions between nodes, where no data points are present, such as the red-lined region in the example in Fig. 1.

The IDW data augmentation component uses the IDW interpolation algorithm to add a new point in each of these empty regions to make the new dataset sufficiently dense with respect to the fuzzy partitions. Then, the direct MF-transform and the regression error are calculated; the algorithm ends if the regression error is less or equal to the threshold α, otherwise the cardinality of the fuzzy partitions is incremented (n = n+1) and the process is iterated.

The algorithm ends with the error message that the data is not dense enough only if the percent of new data points added by the IDW interpolator ap exceeds the 5%

threshold. Beyond this threshold, the percentage of simulated data points would become non-negligible and would significantly distort the original dataset.

To add a new data point in the space of the features the data augmentation component uses the (10) by considering the K closest sample points of the original dataset and neglecting neighboring data points added via interpolation.

A regression error index can be used to calculate the regression error e. To find the best values of the number of closest data points K and the power parameter p in (10), cross-validation techniques can be adopted.

4. Test and results

In order to measure the performance of the IDWMF-transform regression method, a set of tests were performed on well-known time series datasets.

Let $\{(x_{11}, x_{12}, \ldots, x_{1s}, y_1), (x_{21}, x_{22}, \ldots, x_{2s}, y_2), \ldots, (x_{N1}, x_{N2}, \ldots, x_{Ns}, y_N)\}$ a dataset of measures. Each data point is given by s numerical input features and one output feature y.

In choosing the regression error index, we avoided adopting scale-dependent regression error measures that cannot be used, as it is necessary that the regression error threshold α is fixed and does not depend on the unit of measurement of the output variable.

The scale-independent Symmetric Mean absolute Percentage Error (SMAPE) [25,26] is used to measure the regression error. It is given by:

$$e = \text{SMAPE} = \frac{100\%}{N} \sum_{j=1}^{N} \frac{2 \cdot \left| f^{F}_{n_1 n_2 \ldots n_s}(x_{j1}, p_{j2}, \ldots, p_{js}) - y_j \right|}{\left| f^{F}_{n_1 n_2 \ldots n_s}(x_{j1}, p_{j2}, \ldots, p_{js}) \right| + |y_j|} \quad (13)$$

SMAPE is expressed in percentage ranges where a score of 0% indicates a perfect match between the measured and predicted values. With respect to the well-known regression index Mean absolute Percentage Error (MAPE), SMAPE does not have the disadvantage of tending to infinity when the observed value y_j tends to zero; moreover, it is less sensitive to the presence of outliers.

The model was tested using a set of regression datasets in the UCI Machine Learning Repository [52]. In Tab. 1, for each dataset are shown the number of data points, the number of input features and the name of the target feature.

Table 1. Datasets used in the comparison tests.

Dataset	Data points	Input features	Target feature
Abalone	4177	8	Rings
Auto MPG	398	6	Mpg
Computer hardware	209	9	ERP
Liver disorders	345	5	Drinks
Real estate	414	6	Price

Comparison tests are performed with Ridge Regression (RR), Huber Regression (HR), Extreme Gradient Boosting XGB), Random Forest (RF), Support Vector Machine (SVM), K-nearest neighbor (KNN) and MF-transform (MFT).

Each dataset is randomly split in a training and in a testing set, containing, respectively, 80% and 20% of the data. To analyze the performance of each algorithm, for each testing set the regression error indices R^2, RMSE, MAE and MAPE were calculated.

In order to set the two IDW parameters number of neighborhoods K and power parameter, for each dataset a sample consisting of 10% of the data points was randomly extracted. Then, the IDW algorithm was executed multiple times to this sample data to assess the value of the target feature, varying in each execution the parameter K from 5 to 15 and the parameter p from 1 to 5. The RMSE between the predicted and the measured values was calculated to set the optimal values of the two parameters.

For the sake of brevity, only the the tests performed on the Abalone and Real estate datasets are discussed in detail in paragraphs 4.1 and 4.2. The complete comparison results are shown in paragraph 4.3.

4.1. Comparison results on Abalone

The IDW sample randomly extracted from the dataset is given by 420 data points.

In Fig. 4 is shown the trend oof RMSE with respect to k, for various values of p. The RMSE index is minimized for K = 12 and p = 2, then, the values of the two parameters were set, respectively, to 12 and 2.

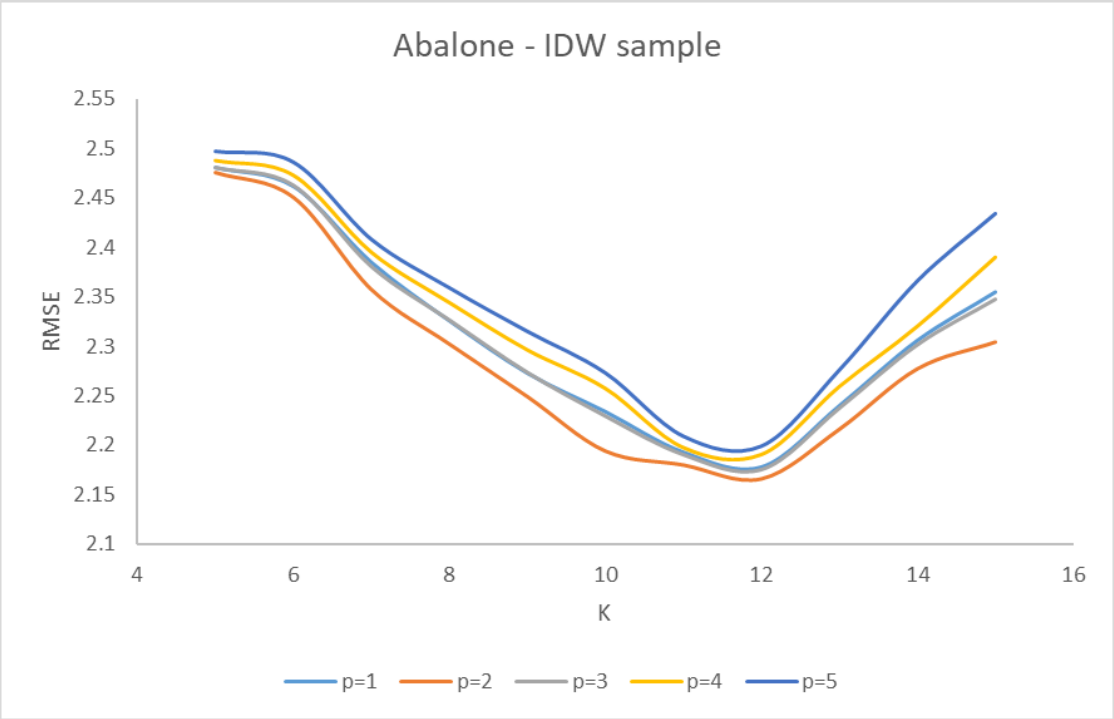


Figure 4. RMSE trend varying K and p for the Abalone IDW sample.

After splitting randomly the dataset, a training set given by 3342 data points and a testing set given by 835 data points are obtained. The eight regression methods are executed on the training set.

MFT and IDW-MFT were executed setting for the threshold error α = 0.5%. Initially the cardinality of the fuzzy partition n is fixed to three, obtaining a value of SMAPE obtained is 0.823%, greater than the threshold value. In the second iteration with n = 4, the SMAPE value is equal to 0.665%, still higher than the threshold. In the third iteration the MF-transform algorithm terminates because the fuzzy partitions are too fine, and the data is not dense enough with respect to the fuzzy partitions. Instead, the IDW-MFT algorithm, applying the IDW-based data augmentation process, terminates as SMAPE = 0.451, which is lower than the threshold. Tab. 2 shows the MFT obtained executing the two algorithms at each iteration.

Table 2. Abalone – SMAPE values obtained varying the cardinality of the fuzzy partitions.

n	MFT (%)	IDW-MFT (%)
3	0.823	0.823
4	0.634	0.665
5	/	0.451

In Tab. 3 are shown the value of the regression indices obtained for the Abalone testing set. The values obtained executing MFT are the ones calculated in the second iteration where the cardinality of the fuzzy relations is n = 4.

Table 3. Abalone - Regression results.

	R^2	RMSE	MAE	MAPE
RR	**0.540**	2.241	1,601	0.161

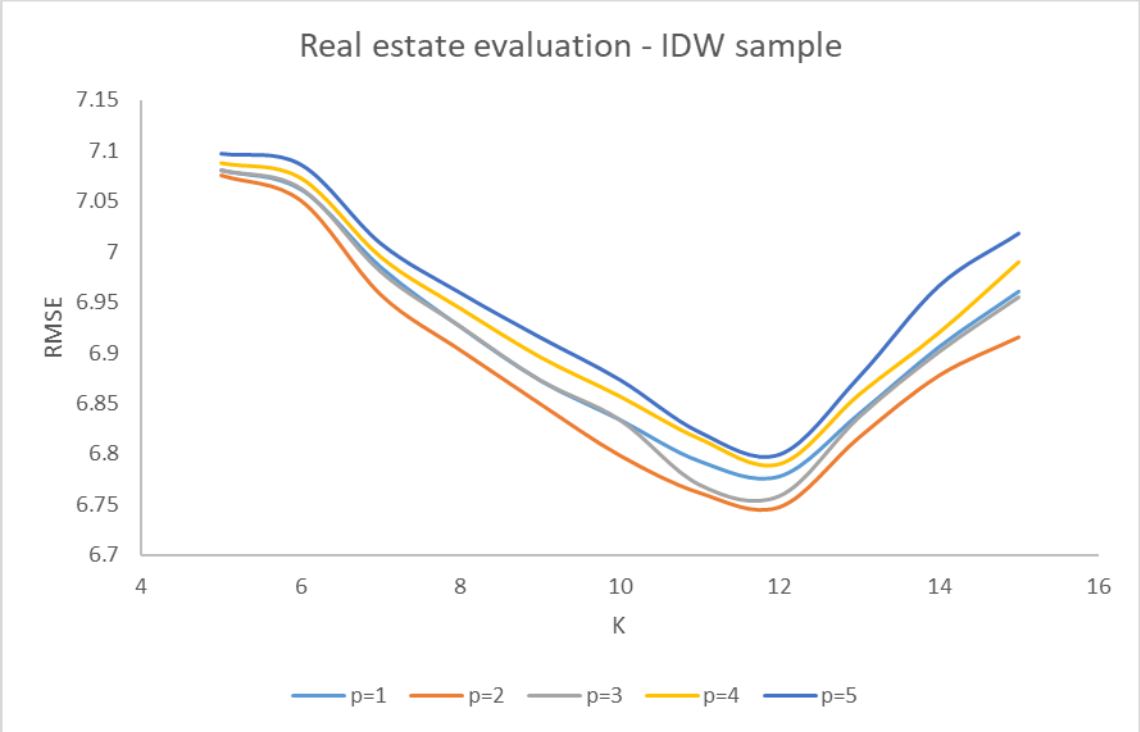
HR	0.532	2.263	1.610	0.156
XGB	0.473	2.391	1.649	0.170
RF	0.538	2.266	1.580	0.159
SVM	0.538	2.255	1.586	0.158
KNN	0.527	2.269	1.578	0.154
MFT	0.483	2.397	1.672	0.185
IDW-MFT	0.535	**2.233**	**1.554**	**0.151**

IDW-MFT provided the best performance in terms of RMSE, MAE and MAPE. The best value of R^2 was obtained executing RR. MFT and XGB produced the worst performances.

4.2. Comparison results on Real estate

In this paragraph are shown the results obtained for the Real estate valuation dataset. The IDW sample randomly extracted from the dataset is given by 50 data points.

In Fig. 5 is shown the trend oof RMSE with respect to k, for various values of p. Even in this case, the minimum values of RMSE ware obtained for K = 12 and p = 2.



After splitting randomly the dataset, a training set given by 331 data points and a testing set given by 83 data points are obtained. The eight regression methods are executed on the training set.

MFT and IDW-MFT were executed setting for the threshold error α = 0.5%. Initially the cardinality of the fuzzy partition n is fixed to three, obtaining a value of SMAPE obtained is 0.859%, greater than the threshold value. In the next iteration with n = 4, the MF-transform algorithm terminates because the fuzzy partitions are too fine, and the data is not dense enough with respect to the fuzzy partitions. Instead, the IDW-MFT algorithm, applying the IDW-based data augmentation process, terminates after two iterations, determining an error SMAPE = 0.451, which is lower than the threshold. Tab. 2 shows the MFT obtained by executing the two algorithms at each iteration.

Table 6. Real estate – SMAPE values obtained varying the cardinality of the fuzzy partitions.

n	**MFT (%)**	**IDW-MFT (%)**
3	0.859	0.859

4	/	0.638
5	/	0.486

In Tab. 7 are shown the value of the regression indices obtained for the Real estate testing set. The values obtained by executing MFT are the ones calculated in the first iterations where the cardinality of the fuzzy relations is n = 3.

Table 7. Real estate - Regression results.

	R^2	RMSE	MAE	MAPE
RR	0.661	7.560	5.589	0.178
HR	0.640	7.774	5.560	0.173
XGB	0.778	5.912	3.952	0.134
RF	0.793	5.833	3.901	0.122
SVM	0.708	6.033	4.328	0.156
KNN	0.649	7.772	5.939	0.185
MFT	0.612	8.065	6.534	0.196
IDW-MFT	**0.801**	**5.778**	**3.876**	**0.117**

IDW-MFT provided the best performance in terms of R^2, RMSE, MAE, MAPE and SMAPE. MFT, RR, HR and KNN produced the worst performances.

In the next paragraph are analyzed the performances of IDW-MFT with respect to the other regression models for all the five UCI Machine Learning datasets. The analysis is conducted by evaluating the gain of IDW-MFT with respect to each of the other methods for all four regression measures: R^2, RMSE, MAE and MAPE.

4.3. IDW-MFT gain with respect to other regression methods

Below, the performance of each regression model against IDW-MFT is compared for all datasets. The comparison is done by measuring, for each regression index, the gain of IDW-MFT over the other regression models.

Tab. 8 shows the gain of IDW-MFT for the R^2 index, given by $\frac{R^2_{IDW-MFT}-R^2}{R^2_{IDW-MFT}}$, where $R^2_{IDW-MFT}$ is the value of R^2 obtained executing IDW-MFT and R^2 is the value of R^2 obtained executing another model.

Gains in R^2 values over the other regression models are reported for each of the five datasets, and they fall between 0 and 0.24. IDW-MFT has the highest gain values compared to MFT and KNN.

Table 8. R^2 gain for all datasets.

	RR	HR	XGB	RF	SVM	KNN	MFT
Abalone	-0.009	0.006	0.116	0.006	0.006	0.015	0.097
Auto MPG	0.110	0.154	0.051	0.023	0.059	0.096	0.224
Computer hardware	0.151	0.102	0.098	0.095	0.067	0.089	0.251
Liver disorders	0.077	0.065	0.074	0.081	0.083	0.105	0.149
Real estate	0.175	0.201	0.029	0.010	0.116	0.190	0.236

For the other three indices the gain is calculated by the formula $\frac{I-I_{IDW-MFT}}{I_{IDW-MFT}}$, where $I_{IDW-MFT}$ is the value of the index obtained executing IDW-MFT, and I is the value of the index obtained executing another model.

The IDW-MFT gain for the RMSE index is displayed in Tab. 9. Significant RMSE gains were observed for all regression models for the datasets of computer hardware and liver disorders, for all regression models other than XGB and RF for the real estate dataset, and for XGB for the Abalone dataset and HR for the Auto MPG dataset. The increase fluctuates between 0.07 and 0.4 in relation to MFT.

Table 9. RMSE gain for all datasets.

	RR	HR	XGB	RF	SVM	KNN	MFT
Abalone	0.004	0.013	0.071	0.015	0.010	0.016	0.073
Auto MPG	0.058	0.057	0.065	0.044	0.043	0.062	0.112
Computer hardware	0.147	0.122	0.096	0.082	0.069	0.101	0.180
Liver disorders	0.091	0.086	0.090	0.097	0.102	0.123	0.194
Real estate	0.308	0.345	0.023	0.010	0.044	0.345	0.396

Tab. 10 displays the improvement of IDW-MFT for the MAE metric. Across all five datasets, the improvements in MAE compared to the other regression models vary from 0 to 0.69. The maximum gain values are with respect to MFT, HR, and KNN.

Table 10. MAE gain for all datasets.

	RR	HR	XGB	RF	SVM	KNN	MFT
Abalone	0.030	0.036	0.061	0.017	0.021	0.015	0.076
Auto MPG	0.053	0.099	0.086	0.027	0.033	0.060	0.163
Computer hardware	0.092	0.096	0.085	0.069	0.057	0.083	0.231
Liver disorders	0.221	0.177	0.183	0.084	0.081	0.128	0.564
Real estate	0.442	0.434	0.020	0.006	0.117	0.532	0.686

The gain of IDW-MFT for the MAPE index is displayed in Tab. 11. The MAPE gains over the other regression models for each of the five datasets fall between 0 and 0.68. IDW-MFT has the highest gain values over MFT, HR, and KNN.

Table 11. MAPE gain for all datasets.

	RR	HR	XGB	RF	SVM	KNN	MFT
Abalone	0.066	0.033	0.126	0.053	0.046	0.020	0.225
Auto MPG	0.092	0.131	0.127	0.073	0.068	0.094	0.387
Computer hardware	0.115	0.147	0.096	0.078	0.070	0.093	0.432
Liver disorders	0.268	0.236	0.424	0.131	0.073	0.155	0.419
Real estate	0.521	0.479	0.145	0.043	0.333	0.581	0.675

These results highlight that IDW-MFT provides, in general, better performance than other well-known regression models, in terms of regression error reduction. For all datasets used in the tests, gains compared to other regression models are recorded for all five error measures.

5. Conclusions

A variation of the Multidimensional F-transform regression method based on the IDW interpolator is proposed. IDW is applied in a data augmentation process performed at each iteration in the regions of the feature space with insufficient data density with respect to the fuzzy partitions. This process allows to overcome the performance limits of MF-transform, which cannot be used when sufficient data density with respect to the fuzzy partitions is not respected.

The results of the comparative tests both with MF-transform and with other well-known regression models have shown that the IDWMF-transform provides better regression performances than MF-transform and the other regression methods for all the five datasets used in the tests.

In the future, we intend to perform further tests on many datasets of different cardinality and size in order to analyze the performance of the model when the number of features and data points varies. Furthermore, a future evolution of the research will be addressed towards an adaptation of the method aimed at managing massive data.

Author Contributions: Conceptualization, B.C. and F.D.M.; methodology, B.C. and F.D.M..; software, B.C. and F.D.M..; validation, B.C. and F.D.M..; formal analysis, B.C. and F.D.M..; investigation, B.C. and F.D.M..; resources, B.C. and F.D.M..; data curation, B.C. and F.D.M..; writing—original

draft preparation, B.C. and F.D.M..; writing—review and editing, B.C. and F.D.M..; visualization, B.C. and F.D.M..; supervision, B.C. and F.D.M.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Perfilieva, I. Fuzzy transforms: Theory and applications. *Fuzzy Sets Syst.* **2006**, *157*, 993–1023. https://doi.org/10.1016/j.fss.2005.11.012
2. Di Martino, F.; Sessa, S. Compression and decompression of images with discrete fuzzy transforms. *Inf. Sci.* **2007**, *17*, 2349–2362. https://doi.org/10.1016/j.ins.2006.12.027
3. Di Martino, F.; Loia, V.; Perfilieva, I.; Sessa, S. An image coding/decoding method based on direct and inverse fuzzy transforms. *Int. J. Approx. Reason.* **2008**, *48*, 110–131. https://doi.org/10.1016/j.ijar.2007.06.008
4. Di Martino, F.; Loia, V.; Sessa, S. Fuzzy transforms for compression and decompression of colour videos. *Inf. Sci.* **2010**, *180*, 3914–3931. https://doi.org/10.1016/j.ins.2010.06.030
5. Perfilieva, I.; De Baets, B. Fuzzy transforms of monotone functions with application to image compression. *Inf. Sci.* **2010**, *180*, 3304–3315. https://doi.org/10.1016/j.ins.2010.04.029
6. Perfilieva, I.; Novàk, V.; Dvoràk, A. Fuzzy transforms in the analysis of data. Int. J. Approx. Reason. **2008**, 48, 36–46. https://doi.org/10.1016/j.ijar.2007.06.003
7. Di Martino, F.; Loia, V.; Sessa S. Fuzzy transforms method and attribute dependency in data analysis. *Information Sciences* **2010**,*180*, 493-505. https://doi.org/10.1016/j.ins.2009.10.012
8. Stepnicka, M.; Polakovic, O. A neural network approach to the fuzzy transform *Fuzzy Sets and Systems* **2009**, 160, 1037-1047. https://doi.org/10.1016/j.fss.2008.11.02
9. Di Martino, F.; Loia, V.; Sessa S. Fuzzy transforms method in prediction data analysis. Fuzzy Sets and *Systems*, **2011**, *180*, 146-163. https://doi.org/10.1016/j.fss.2010.11.009
10. Di Martino, F.; Sessa S. Fuzzy transforms prediction in spatial analysis and its application to demographic balance data. *Soft Computing* **2017**,*21*, 3537–3550. https://doi.org/10.1007/s00500-017-2621-8
11. Di Martino, F.; Sessa S. A classification algorithm based on multi-dimensional fuzzy transforms. *J Ambient Intell Human Comput*ing **2021**, *13*, 2873–2885 (2022). https://doi.org/10.1007/s12652-021-03336-0
12. Di Martino, F.; Sessa, S. Time Series Seasonal Analysis Based on Fuzzy Transforms. *Symmetry* **2017**, *9*, 281. https://doi.org/10.3390/sym9110281
13. Loia, V.; Tomasiello, S.; Vaccaro, A.; Gao, J. Using local learning with fuzzy transform: application to short term forecasting problems, *Fuzzy Optim. Decis. Mak.* **2020**, *19*, 13–32. https://doi.org/10.1007/s10700-019-09311-x
14. Gedara, T.M.H.; Loia, V.; Tomasiello, S. Using fuzzy transform for sustainable fake news detection. *Applied Soft Computing* **2024**, *151*, 111173, 7 pp. https://doi.org/10.1016/j.asoc.2023.11117
15. Hurtik, P.; Tomasiello, S. A review on the application of fuzzy transform in data and image compression. *Soft Comput* **2019**, *23*, 12641–12653 201). https://doi.org/10.1007/s00500-019-03816-8
16. Di Martino, F.; Sessa, S. Fuzzy Transforms for Image Processing and Data Analysis—Core Concepts, Processes and Applications; Springer Nature: Cham, Switzerland, 2020; p. 217. https://doi.org/10.1007/978-3-030-44613-0
17. Di Martino, F.; Perfilieva, I.; Sessa, S. A Summary of F-Transform Techniques in Data Analysis. *Electronics* **2021**, *10*, 1771. https://doi.org/10.3390/electronics10151771
18. Patané, G. Out-of-Sample Extension of the Fuzzy Transform *IEEE Transactions on Fuzzy Systems*, **2024**, *32*(3), 1424-1434, March 2024. https://doi.org/10.1109/TFUZZ.2023.3326657
19. Cardone, B.; Martino, F.D. A Novel Classification Algorithm Based on Multidimensional F^1 Fuzzy Transform and PCA Feature Extraction. *Algorithms* **2023**, *16*, 128. https://doi.org/10.3390/a16030128
20. Shepard, D. A two-dimensional interpolation function for irregularly-spaced data. Proceedings of the 1968 23rd ACM National Conference. Washington, DC, USA, August 27-29 1968, Association for Computing Machinery Publisher, New York,United States, 1968, pp. 517–524. doi:10.1145/800186.810616.
21. Allasia, G. Some physical and mathematical properties of inverse distance weighted methodsfor scattered data interpolation. *Calcolo* 1992, *29*(1), 97–109 (1992). https://doi.org/10.1007/BF02576764
22. Lukaszyk, S. A new concept of probability metric and its applications in approximation of scattered data sets. *Comput. Mech.* 2004, *33*(4), 299–304. https://doi.org/10.1007/s00466-003-0532-2
23. Kearney, K. M.; . Harley, J. B.; Nichols J. A. Inverse distance weighting to rapidly generate large simulation datasets *Journal of Biomechanics* **2023**, *158*, 111764. https://doi.org/10.1016/j.jbiomech.2023.111764
24. Mueller, T.G.; Dhanikonda, S.R.K.; Pusuluri, N.B.; Karathanasis, A.D.; Mathias, K.K., ;Mijatovic, B.; Sears, B.G. Optimizing inverse distance weighted interpolation with cross-validation. *Soil Sci.* **2005**, *170*(7), 504–515. https://doi.org/10.1097/01.ss.0000175342.30164.89
25. Armstrong, J. S. Long-range forecasting: From crystal ball to computer. John Wiley & Sons, 1978, 630 pp. ISBN:978-0471030027
26. Nguyen, N., T.; Nguyen, B. M.; Nguyen., G. Efficient time-series forecasting using neural network and opposition-based coral reefs optimization. *International Journal of Computational Intelligence Systems*, 12(2):1144–1161, 2019.

 doi:10.20944/preprints202501.2127.v1

27. Kelly, M.; Longjohn, R.; Nottingham, K. The UCI Machine Learning Repository. 2024. Available online: https://archive.ics.uci.edu/ (accessed on 01 December 2024) .