electronics

MDPI

Review

# A Summary of F-Transform Techniques in Data Analysis

Ferdinando Di Martino [1,2,*], Irina Perfilieva [3] and Salvatore Sessa [1,2]

1 Dipartimento di Architettura, Università degli Studi di Napoli Federico II, Via Toledo 402, 80134 Napoli, Italy; sessa@unina.it
2 di Ricerca "Alberto Calza Bini", Università degli Studi di Napoli Federico II, Via Toledo 402, 80134 Napoli, Italy
3 Institute for Research and Applications of Fuzzy Modeling, 708 00 Ostrava, Czech Republic; irina.perfilieva@osu.cz
* Correspondence: fdimarti@unina.it; Tel.: +39-081-253-8907; Fax: +39-081-253-8905

**Abstract:** Fuzzy transform is a technique applied to approximate a function of one or more variables applied by researchers in various image and data analysis. In this work we present a summary of a fuzzy transform method proposed in recent years in different data mining disciplines, such as the detection of relationships between features and the extraction of association rules, time series analysis, data classification. After having given the definition of the concept of Fuzzy Transform in one or more dimensions in which the constraint of sufficient data density with respect to fuzzy partitions is also explored, the data analysis approaches recently proposed in the literature based on the use of the Fuzzy Transform are analyzed. In particular, the strategies adopted in these approaches for managing the constraint of sufficient data density and the performance results obtained, compared with those measured by adopting other methods in the literature, are explored. The last section is dedicated to final considerations and future scenarios for using the Fuzzy Transform for the analysis of massive and high-dimensional data.

**Keywords:** direct F-transform; inverse F-transform; multi-dimensional F-transform; fuzzy partition; dependency between attributes time series; data classification

**Citation:** 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

Academic Editor: Javid Taheri

Received: 28 June 2021
Accepted: 22 July 2021
Published: 24 July 2021



## 1. Introduction

Fuzzy Transform (for short, F-transform) [1,2] is a recent soft computing approximation technique, successfully used in numerous applications in image and data analysis (see, e.g., [3] for an in-depth discussion on this matter).

In particular, the properties of the F-transform in the information aggregation and function approximation favors its use in many data analysis and data mining problems.

The aim of this paper is to provide an in-depth overview of soft computing data analysis techniques based on the use of the F-transform proposed in the literature.

Some variations of basic functions used to constrict the F-transform are proposed in [4], in which they are given by B-spline functions, and in [5] where the basic functions are given by block pulse functions.

Recently an extension of the basic F-transform on higher-degree F-transform was introduced in [6] by generalizing the case of constant (zero-order) components to the case of m-order polynomial components. In [7,8] the applicability of the m-order F-transform is discussed and an application of the one-degree F-transform in seasonal time series forecasting is presented in [9]. However, while increasing the performance in terms of accuracy and precision of the results compared to basic F-transforms, the higher-degree fuzzy transforms are computationally more complex to manage and this makes them unsuitable for use in data analysis applications, especially in the presence of datasets of high cardinality and size.

In this work we focus on the application of the basic (zero-order) F-transform in data analysis. We will discuss the techniques proposed in the literature that employ the direct and inverse zero-order F-transform in data mining problems, such as dependencies between attributes, time series analysis and data classification, analyzing their critical points and performance benefits.

F-transform techniques were initially applied in image analysis in which the constraint of sufficient density described in Section 2 is always respected. In data analysis, however, the application of the F-transform necessarily requires the management of this constraint and the choice of suitable fuzzy partitions of the domains of the input variable and the choice of the appropriate dimensionality of the fuzzy partitions which cannot be too fine, to guarantee sufficient data density, nor too coarse grained, to guarantee high performance levels.

In Section 2 we introduce the one-dimensional and multi-dimensional F-transforms, providing a summary of their characteristics. In particular, the constraint of sufficient density of the data will be analyzed, which is of extreme importance in the use of F-transform techniques in data analysis. In Section 3 are discussed the methods proposed in the literature applying the multidimensional F-transform in the analysis of dependencies between attributes in the data and in detecting association rules. Section 4 focuses on the F-transform techniques applied in time series analysis. In Section 5 a classification method based on the multi-dimensional F-transform is discussed. Final considerations are contained in Section 6. A list with descriptions of all acronyms and abbreviations in the text is given in Appendix A.

## 2. Preliminaries

### *2.1. Basic Functions*

Let X = [a,b] be a close interval in *R* and $\{x_1, x_2, \ldots, x_n\}$ be a set of n fixed points in [a,b] such that $3 \le n$ and $a = x_1 < x_2 < \ldots < x_n = b$.

In [1,2] the following definition of fuzzy partition of X was introduced: the fuzzy sets $A_1, \ldots, A_n$: [a,b] → [0,1] form a *(generalized) fuzzy partition* of [a,b], if for each k = 2, …, n − 1, 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. $A_k$ is continuous in $[x_{k-1}, x_{k+1}]$ (continuity)
4. $A_k$ is strictly decreasing in $(x_{k-1}, x_k)$ and strictly increasing in $(x_k, x_{k+1})$
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 an h-uniform fuzzy partition the following additional properties hold:

1. $A_k(x_k - x) = A_k(x_k + x)\ \forall x \in [0, h]$
2. $A_k(x) = A_{k-1}(x - h)$ 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, e.g., [2]) by an 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:

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

*2.2. One-Dimensional Direct and Inverse F-Transform*

Let {$A_1$, $A_2$, …, $A_n$} be a fuzzy partition of [a,b] and *f*(*x*) be a continuous function on [a,b]. The n-tuple $[F_1, F_2, \dots, F_n]$ with components:

$$F_k = \frac{\int_a^b f(x)A_k(x)dx}{\int_a^b A_k(x)dx} \quad k = 1, \dots, n \tag{2}$$

is called the *fuzzy transform* of *f* with respect to {$A_1$, $A_2$, …, $A_n$}. The $F_k$ are called *components* of the F-transform.

If the fuzzy partition {$A_1$, $A_2$, …, $A_n$} is uniform with nodes $x_1$, $x_2$, …, $x_n$, the components are given (cfr. [2] Lemma 1) by the formula:

$$F_k = \begin{cases} \frac{2}{h}\int_{x_1}^{x_2} f(x)A_k(x)dx & \text{if } k = 1 \\ \frac{1}{h}\int_{x_{i-1}}^{x_i} f(x)A_k(x)dx & \text{if } k = 2,\dots,n-1 \\ \frac{2}{h}\int_{x_{n-1}}^{x_n} f(x)A_k(x)dx & \text{if } k = n \end{cases} \tag{3}$$

Now we define the following function on [a,b] given by a weighted average of the basic functions in which the weights are the F-transform components:

$$f_{F,n}(x) = \sum_{k=1}^{n} F_k\, A_k(x) \quad x \in [a, b] \tag{4}$$

It is called *inverse F-transform* of *f* with respect to the uniform fuzzy partition {$A_1$, $A_2$, …, An}. An important theorem proves that the function $f_{F,n}$ approximates the continuous function *f* on [a,b] with arbitrary precision. We enunciate below this theorem and its proof is given in [2] Theorem 2.

**Theorem 1.** *Let f(x) be a continuous function on [a,b]. For every* $\varepsilon > 0$*, then there exist an integer* $n(\varepsilon)$ *and a related fuzzy partition {*$A_1$*,* $A_2$*, …,* $A_{n(\varepsilon)}$ *} of [a,b] such that for all* $x \in [a, b]$ *results* $|f(x) - f_{F,n(\varepsilon)}(x)| < \varepsilon$.

Theorem 1 concerns the approximation of a known continuous function *f*, but in many cases we only know that the function *f* assumes determined values in a set of m points $p_1,\dots,p_m \in [a,b]$.

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 ∈ {1, …,m} such that $A_k(p_j) > 0$. Then we can define the n-tuple [$F_1$, $F_2$, …,$F_n$] as the *discrete F-transform* of *f* with respect to {$A_1$, $A_2$, …, $A_n$ }, where each $F_k$ is given by

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

Then we call the *discrete* inverse F-transform of *f* with respect to {$A_1$, $A_2$, …, $A_n$} to be the following function defined in the same points $p_1$,..., $p_m$ of [a,b]:

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

Analogously to Theorem 1, we have the following approximation theorem (its proof is given in [2] Theorem 5.

**Theorem 2.** *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 exists 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, holds* $\left|f(x) - f_{F,n(\varepsilon)}(x)\right| < \varepsilon$.

Theorem 2 states that the inverse F-transform (6) approximates the original continuous function *f* in a point with an arbitrary precision.

### *2.3. Multi-Dimensional Direct and Inverse F-Transform*

The one-dimensional F-transform can be extended to approximate continuous functions defined in a N-dimensional domain given by the Cartesian product $[a_1,b_1] \times [a_2,b_2] \times\ldots \times [a_s,b_s]$ of s real intervals $[a_i,b_i] \subseteq R$ (i = 1, …, s).

Let $f$: $[a_1,b_1] \times [a_2,b_2] \times\ldots \times [a_s,b_s] \rightarrow R$ be a continuous function on the universe of discourse. Let $\{A_{11}, A_{12}, \ldots, A_{1n_1}\}$, $\{A_{21}, A_{22}, \ldots, A_{2n_2}\}$, ... $\{A_{s1}, A_{s2}, \ldots, A_{sn_s}\}$ uniform fuzzy partitions of $[a_1,b_1]$, …,$[a_s,b_s]$, respectively.

The F-transform of the function f with respect to $\{A_{11}, A_{12}, \ldots, A_{1n_1}\}, \{A_{21}, A_{22}, \ldots, A_{2n_2}\}, \ldots, \{A_{s1}, A_{s2}, \ldots, A_{sn_s}\}$ ,are the functions given by

$$F_{k_1k_2\ldots k_s} = \frac{\int_{a_s}^{b_s}\cdots\int_{a_2}^{b_2}\int_{a_1}^{b_1} f(x_1, x_2, \ldots, x_s) A_{k_1}(x_1) A_{k_2}(x_2)\ldots A_{k_s}(x_s)\, dx_1 dx_2 \ldots dx_s}{\int_{a_s}^{b_s}\cdots\int_{a_2}^{b_2}\int_{a_1}^{b_1} A_{k_1}(x_1) A_{k_2}(x_2)\ldots A_{k_s}(x_s)\, dx_1 dx_2 \ldots dx_s} \quad (7)$$

being $k_1 = 1, \ldots, n_1$, $k_2 = 1, \ldots, n_2$, …, $k_s = 1, \ldots, n_s$.

The inverse F-transform of the function $f$ with respect to $\{A_{11}, A_{12}, \ldots, A_{1n_1}\}$, $\{A_{21}, A_{22}, \ldots, A_{2n_2}\}$, ... $\{A_{s1}, A_{s2}, \ldots, A_{sn_s}\}$ are the following functions defined on $[a_1,b_1] \times [a_2,b_2] \times\ldots\times [a_s,b_s]$:

$$f^F_{n_1n_2\ldots n_s}(x_1, x_2, \ldots, x_s) = \sum_{k_1=1}^{n_1}\sum_{k_2=1}^{n_2}\cdots\sum_{k_s=1}^{n_s} F_{k_1k_2\ldots k_s} A_{k_1}(x_1)\, A_{k_2}(x_2)\ldots A_{k_s}(x_s) \quad (8)$$

Let the function $f(x_1,x_2,\ldots, x_s)$ be known in N points $p_j = (p_{j1}, p_{j2}, \ldots, p_{js}) \in [a_1,b_1] \times [a_2,b_2] \times\ldots\times [a_s,b_s]$ being j = 1,2, …,N.

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 called *sufficiently dense with respect to the partitions* $\{A_{11}, A_{12}, \ldots, A_{1n_1}\}$, …, $\{A_{s1}, A_{s2}, \ldots, A_{sn_s}\}$ if, for any combination $(h_1, \ldots, h_s) \in \{1, \ldots, n_1\} \times\ldots\times \{1, \ldots, n_s\}$ there is some $p_v = (p_{v1}, p_{v2}, p_{vs}) \in P$, v ∈ {1, …,N}, such that $A_{1h_1}(p_{v1}) \cdot A_{2h_2}(p_{v2}) \cdot \ldots \cdot A_{s_{h_s}}(p_{vs}) > 0$. So we can define the $(h_1,h_2, \ldots,h_s)$th components $F_{h_1h_2\ldots h_s}$ of the direct F-transform of *f* with respect to the basic functions $\{A_{11}, A_{12}, \ldots, A_{1n_1}\}$, …,$\{A_{s1}, A_{s2}, \ldots, A_{sn_s}\}$ as

$$F_{h_1h_2\ldots h_s} = \frac{\sum_{j=1}^{N} f(p_{j1}, p_{j2}, \ldots p_{js}) \cdot A_{1h_1}(p_{j1}) \cdot A_{2h_2}(p_{j2}) \cdot\ldots\cdot A_{sh_s}(p_{js})}{\sum_{j=1}^{N} A_{1h_1}(p_{j1}) \cdot A_{2h_2}(p_{j2}) \cdot\ldots\cdot A_{sh_s}(p_{js})} \quad (9)$$

If the set P is sufficiently dense with respect to the fuzzy partition we can define the *inverse multi-dimensional F-transform* of $f$ with respect to the basic functions $\{A_{11}, A_{12}, \ldots, A_{1n_1}\}$, $\{A_{21}, A_{22}, \ldots, A_{2n_2}\}$, ... $\{A_{s1}, A_{s2}, \ldots, A_{sn_s}\}$ to be the following functions by setting for each point $p_j = (p_{j1}, p_{j2}, \ldots, p_{js}) \in [a_1,b_1] \times\ldots\times [a_s,b_s]$:

$$f^F_{n_1n_2\ldots n_s}(p_{j1}, p_{j2}, \ldots, p_{js}) = \sum_{h_1=1}^{n_1}\sum_{h_2=1}^{n_2}\cdots\sum_{h_s=1}^{n_s} F_{h_1h_2\ldots h_s} \cdot A_{1_{h_1}}(p_{j1}) \cdot\ldots\cdot A_{s_{h_s}}(p_{js}) \quad (10)$$

for j = 1, …,N. The following theorem, which is an extension of Theorem 2, holds:

**Theorem 3.** *Let* $f(x_1,\ldots, x_s)$ *be a function assigned on the set of points P = $\{(p_{11},p_{12}, \ldots,p_{1s}),(p_{21}, p_{22}, \ldots, p_{2s}), \ldots,(p_{m1}, p_{m2}, \ldots,p_{ms})\} \subset [a_1,b_1] \times [a_2,b_2] \times\ldots\times [a_k,b_s]$ and assuming values in [0,1]. Then for every $\varepsilon > 0$, there exist k integers $n_1(\varepsilon)$, …, $n_s(\varepsilon)$ and related fuzzy partitions*

$\{A_{11}, A_{12}, \ldots, A_{1n_1(\varepsilon)}\}$, …,$\{A_{s1}, A_{s2}, \ldots, A_{sn_s(\varepsilon)}\}$ *such that the set P is sufficiently dense with respect to this fuzzy partitions. Moreover, for every* $\mathbf{p}_j = (p_{j1}, p_{j2}, \ldots, p_{js}) \in P$, $j = 1, \ldots, m$, *the following inequality holds.*

$$\left| f(p_{j1}, p_{j2}, \ldots, p_{js}) - f^F_{n_1(\varepsilon)n_2(\varepsilon)\cdots n_s(\varepsilon)}(p_{j1}, p_{j2}, \ldots, p_{js}) \right| < \varepsilon \tag{11}$$

The inverse multi-dimensional F-transform $f^F_{n_1 n_2 \ldots n_s}$ can be used in regression analysis only if the input dataset is sufficiently dense with respect to the set of fuzzy partitions $\{A_{11}, A_{12}, \ldots, A_{1n_1}\}$, $\{A_{21}, A_{22}, \ldots, A_{2n_2}\}$, … $\{A_{s1}, A_{s2}, \ldots, A_{sn_s}\}$.

In Figure 1 an example of data points not sufficiently dense with respect to the fuzzy partition is shown. Let $\{A_{11}, A_{12}, \ldots, A_{1n_1}\}$ be a fuzzy partition of the domain $[a_1, b_1]$ and $\{A_{21}, A_{22}, \ldots, A_{2n_2}\}$ be a fuzzy partition of the domain $[a_2, b_2]$. The data points are shown in red. No data points are located within the subset $[a_{1\ h-1}, \ldots, a_{1\ h+1}] \times [a_{2\ k-1}, \ldots, a_{2\ k+1}]$, corresponding to the dark yellow area in Figure 1. Consequently, for each data point $\mathbf{p}_j = (p_{j1}, p_{j2})$ $j = 1, \ldots, N$ we have $A_{1_h}(p_{j1}) = 0$ and $A_{2_k}(p_{j2}) = 0$. Then, the data are not sufficiently dense with respect to this set of two fuzzy partitions.

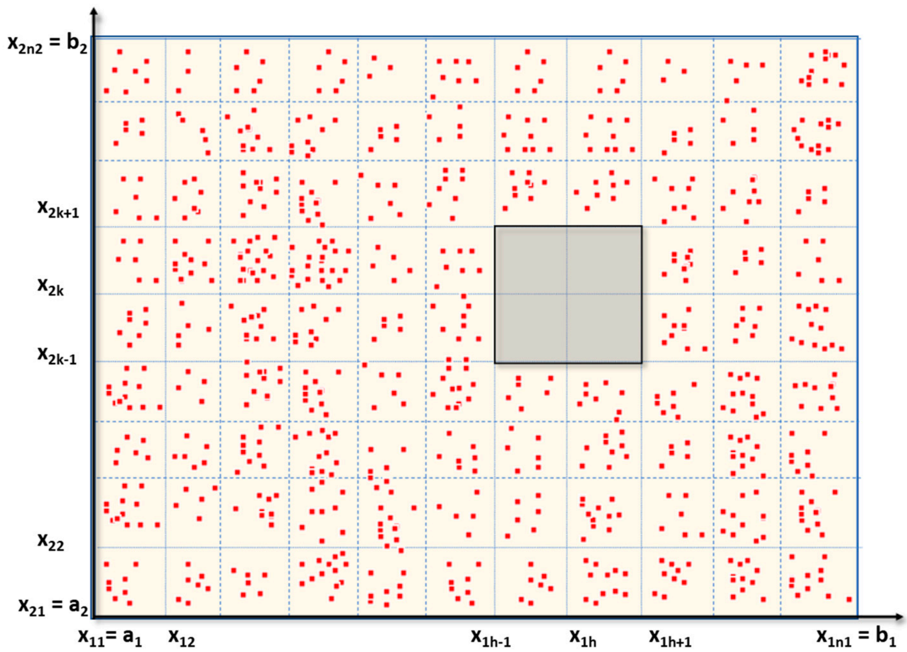


**Figure 1.** Example of non sufficiently dense data points with respect to the fuzzy partitions.

In Figure 2 two examples of fuzzy partitions that are more coarse-grained with respect to the fuzzy partitions are shown in Figure 1. In both cases the data points are sufficiently dense with respect to the fuzzy partitions.

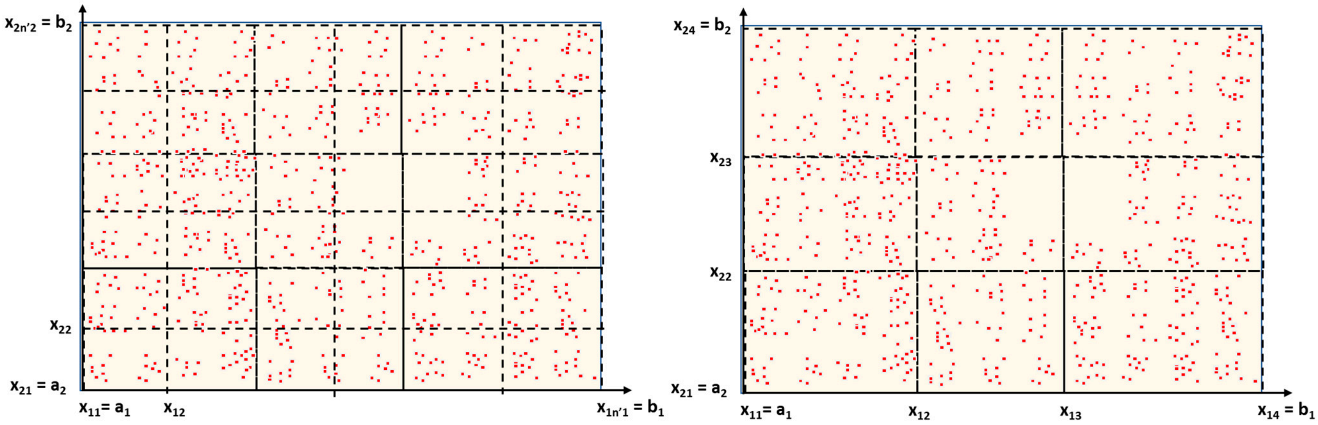


**Figure 2.** Examples of data points sufficiently dense with respect to the fuzzy partitions.

It is necessary to properly set the size of the fuzzy partitions. In fact, the use of fuzzy partitions that are too thin can make the data points not sufficiently dense with respect to them; on the contrary, fuzzy partitions that are too coarse grained, while guaranteeing the sufficient density of the data points, can significantly reduce the performances of the regression analysis methods in which the inverse multi-dimensional fuzzy transform is used as a regression function.

## 3. Multi-Dimensional F-Transform Methods to Explore Dependency and Rules in the Data

### *3.1. Multi-Dimensional F-Transform Techniques to Detect Dependency between Attributes in Datasets*

The multi-dimensional F-transform was applied by many researchers to detect dependency among numerical features in datasets.

In [10,11] the multi-dimensional discrete F-transform is applied to find dependency between attributes in the data.

Following [10,11] a dataset with r features can be schematized as a relation with r attributes and m instances as in Table 1.

**Table 1.** Schema of a relation with r attributes and m instances.

| | $X_1$ | … | $X_i$ | … | $X_r$ |
|---|---|---|---|---|---|
| $O_1$ | $p_{11}$ | . | $p_{1i}$ | . | $p_{1r}$ |
| . | . | . | . | . | . |
| . | . | . | . | . | . |
| . | . | . | . | . | . |
| $O_j$ | $p_{j1}$ | . | $p_{ji}$ | . | $p_{jr}$ |
| . | . | . | . | . | . |
| . | . | . | . | . | . |
| . | . | . | . | . | . |
| $O_m$ | $p_{m1}$ | . | $p_{mi}$ | . | $p_{mr}$ |

where $X_1, \ldots, X_i, \ldots, X_r$ are the attributes and, $O_1, \ldots, O_j, \ldots, O_m$ (m>r) are the objects in the dataset; each object $O_j$ is given by an r-dimensional data point ($p_{j1}, \ldots, p_{ji}, \ldots, p_{jr}$) where $p_{ji}$ is the value assumed by $O_j$ of the attribute $X_i$.

The attribute $X_i$ is a variable assuming values in the real interval $[a_i,b_i]$ defined by setting $a_i = \min\{p_{1i}, \ldots, p_{mi}\}$ and $b_i = \max\{p_{1i}, \ldots, p_{mi}\}$.

In [10,11] the dependency is studied among attributes in the form:

$$X_z = H(X_1, \ldots, X_K) \tag{12}$$

where $H$: $[a_1,b_1] \times [a_2,b_2] \times\ldots\times [a_k,b_k] \rightarrow [a_z,b_z]$ is a continuous function of k variables.

In [10] the multi-dimensional inverse F-transform was applied as a regression function to assess the functional dependency (12). The given function $H(X_1, \ldots, X_k)$ is known in m points $P_j = (p_{j1},p_{j2}, \ldots,p_{jk})$, j = 1, …,m, by setting $H(p_{j1},p_{j2}, \ldots,p_{jk}) = p_{jz}$ for j = 1,2, …,m.

For any interval $[a_i,b_i]$, i = 1, …,k, a fuzzy partition $\{A_{i1}, A_{i2}, \ldots, A_{in_i}\}$ is created with $n_i \geq 3$. If the set of m points is sufficiently dense with respect to these fuzzy partitions, we can define the multi-dimensional direct F-transform of $H$ with $(h_1,h_2, \ldots,h_k)$th components given by

$$F_{h_1 h_2 \ldots h_k} = \frac{\sum_{j=1}^{m} p_{jz} \cdot A_{1_{h_1}}(p_{j1}) \cdot \ldots \cdot A_{k_{h_K}}(p_{jk})}{\sum_{j=1}^{m} A_{1_{h_1}}(p_{j1}) \cdot \ldots \cdot A_{k_{h_K}}(p_{jk})} \tag{13}$$

Using formula (10), the inverse F-transform $H^F_{n_1 n_2 \ldots n_k}$ of H in the point $P_j$ is given by

$$H^F_{n_1 n_2 \ldots n_k}(p_{j1}, p_{j2}, \ldots p_{jk}) = \sum_{h_1=1}^{n_1} \sum_{h_2=1}^{n_2} \ldots \sum_{h_k=1}^{n_k} F_{h_1 h_2 \ldots h_K} \cdot A_{1h_1}(p_{j1}) \cdot \ldots \cdot A_{kh_K}(p_{jk}) \tag{14}$$

In [10] a measure of the dependency of $X_z$ from $X_1, \ldots, X_k$ evauated by (14) is given by the statistical index of determinacy:

$$\mathrm{r}_c^2 = \frac{\sum_{j=1}^{m} \left( \mathrm{H}^F_{n_1 n_2 \ldots n_k}(p_{j1}, p_{j2}, \ldots p_{jk}) - \hat{p}_z \right)^2}{\sum_{j=1}^{m} \left( p_{jz} - \hat{p}_z \right)^2} \tag{15}$$

where $\hat{p}_z$ is the average of values $p_{1z}, p_{2z}, \ldots, p_{mz}$ of the attribute $X_z$.

The index of determinacy $r_c^2$ ranges in the interval [0,1], where $r_c^2 = 0$ means that $H^F_{n_1 n_2 \ldots n_k}$ does not fit to the data and, conversely, $r_c^2 = 1$ means that $H^F_{n_1 n_2 \ldots n_k}$ fits perfectly to the data.

A variation of the formula (15) used in multiple regression analysis to take into account the number of independent variables k and the scale of the data sample is given by (Johnson and Wichern, 1998):

$$\mathrm{r}'^2_c = 1 - \left[ (1 - \mathrm{r}_c^2) \cdot \frac{\mathrm{m} - 1}{\mathrm{m} - \mathrm{k} - 1} \right] \tag{16}$$

This formula includes both the number of independent variables k and the scale of the data sample. The function H in the point $(x_1, x_2, \ldots,x_k)$ is approximated by the following formula:

$$H^F_{n_1 n_2 \ldots n_k}(x_1, x_2, \ldots x_k) = \sum_{h_1=1}^{n_1} \sum_{h_2=1}^{n_2} \ldots \sum_{h_k=1}^{n_k} F_{h_1 h_2 \ldots h_K} \cdot A_{1h_1}(x_1) \cdot \ldots \cdot A_{kh_K}(x_k) \tag{17}$$

In [10] the inverse multi-dimensional F-transform is applied to find dependency among attributes in the dataset containing economic data measured in the Czech Republic in quarters starting from 1997. The two indices of determinacy (15) and (16) are used to evaluate the existence of such dependency.

The results obtained show that the inverse multi-dimensional F-transform provides good performance used as a regression function for the analysis of the dependency between numerical attributes in the datasets. However, it is necessary to determine the optimal fuzzy partitions of the domains of the input attributes and check when the data points are not sufficiently dense. In [11] an algorithm has been proposed that finds the optimal fuzzy partitions and checks that the data points are sufficiently dense with respect to the fuzzy partition. This algorithm is schematized in Figure 3.

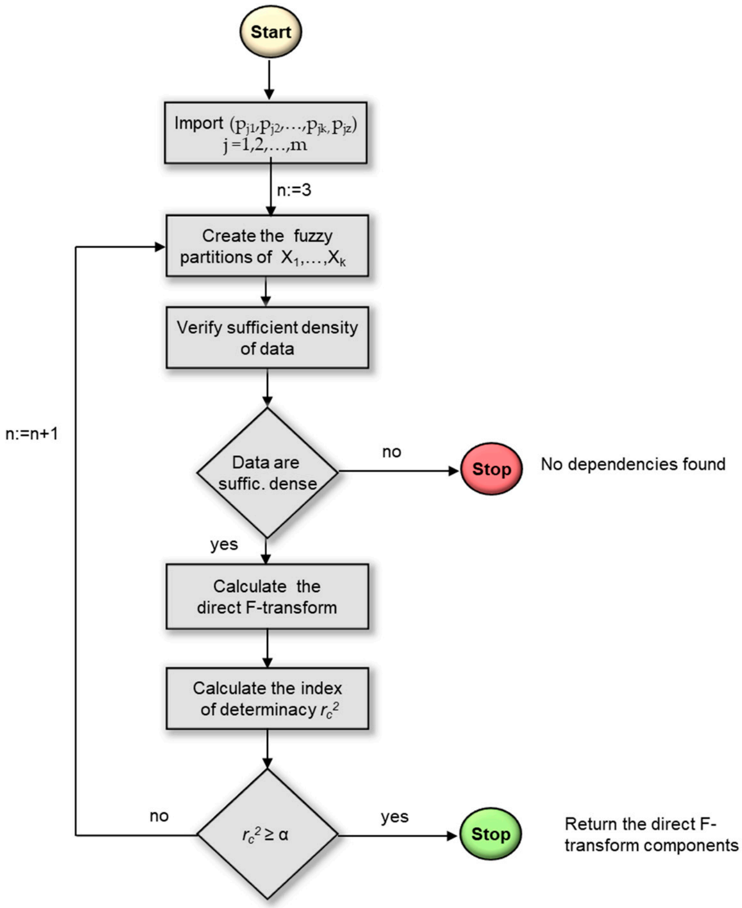


**Figure 3.** Flow diagram of the algorithm proposed in [11].

To reduce the computational costs, the same number n of fuzzy sets is assigned to each of the fuzzy partitions of the input attribute domains. Initially the minimum value n = 3 is set; in each cycle the algorithm checks that the data points are sufficiently dense with respect to the fuzzy partitions and, successively, calculates the direct multi-dimensional F-transform and, for each data point, the inverse multi-dimensional F-transform, finally measuring the value of the index of determinacy. If this value exceeds a predetermined $\alpha$ threshold, the algorithm ends by returning the components of the direct F-transform, otherwise a successive iteration is performed in which the number n is increased by one unit. If, during an iteration, the data points are not sufficiently dense with respect to the fuzzy partition, the algorithm terminates by reporting that it has not found the dependency of $X_z$ on the attributes $X_1$, ..., $X_k$.

In [11] this algorithm is executed to explore dependency between oceanographic and surface meteorological attributes of a dataset containing data measured from a series of buoys positioned throughout the Equatorial Ocean Pacific and used to analyze the El Nino/Southern Oscillation (ENSO) cycles.

The application of the multi-dimensional F-transform as a machine learning regression function can become expensive in the presence of massive datasets in which the number of data points and the number of features become higher. In a recent work [12] an extension of the algorithm is proposed in [11], called MFAD (Massive F-transform Attribute Dependency) aimed to find dependencies between numerical attributes in massive datasets. MFAD apply a uniform sampling algorithm to partition the dataset in subsets having the same cardinality. The F-transform attribute dependency algorithm [11] is executed on each subset returning the multi-dimensional direct F-transform components (13) and the index of determinacy (16). Let $\mathbf{F_q}$ be the direct F-transform vector obtained applying the F-transform attribute dependency algorithm on the qth subset, where q = 1, …,s.

The functional dependency of $X_z$ from $X_1, X_2, \ldots, X_k$ in the form $X_z = H(X_1, X_2, \ldots, X_k)$ in a point $(x_1, x_2, \ldots, x_k)$ is evaluated computing the following weighted average:

$$H^F(x_1,x_2,\ldots,x_k) = \frac{\sum_{q=1}^{s} w_p(x_1,x_2,\ldots,x_k) \cdot H_{nq}^F(x_1,x_2,\ldots,x_k)}{\sum_{p=1}^{s} w_q(x_1,x_2,\ldots,x_k)} \quad (18)$$

where $H_{nq}^F(x_1,x_2,\ldots,x_k)$ q = 1, …,s is the value of the inverse multi-dimensional F-transform in the point $(x_1, x_2, \ldots, x_k)$ obtained by (17) using the qth direct F-transfom $\mathbf{F_q}$ and the weighted term $w_q(x_1,x_2,\ldots,x_k)$, q = 1, …,s, is given by the formula:

$$w_q(x_1,x_2,\ldots,x_k) = \begin{cases} r_{cq}^2 & \text{if } (x_1,x_2,\ldots,x_k) \in D_q \\ 0 & \text{otherwise} \end{cases} \quad (19)$$

The closed set $D_q = [a_{q1},b_{q1}] \times [a_{q2},b_{q2}] \times \ldots \times [a_{qk},b_{qk}]$ is the domain in which is defined the qth subset and $r_{cq}^2$ is the index of determinacy obtained by executing the F-transform attribute dependency algorithm on the qth subset.

The greater the index of determinacy $r_{cq}^2$, the greater the weight of the inverse multi-dimensional F-transform $H_{nq}^F(x_1,x_2,\ldots,x_k)$ in the approximation of the function $H$ in the point $(x_1, x_2, \ldots, x_k)$. The weighted term (19) is null if the point $(x_1, x_2, \ldots, x_k)$ is outside the domain $D_q$.

In Figure 4 the MFAD method is schematized. Each subset is treated separately by applying the F-transform attribute dependency algorithm. The regression function is constituted by the weighted average of the single inverse-fuzzy transforms where the weights are the values of the index of determinacy obtained for each subset.

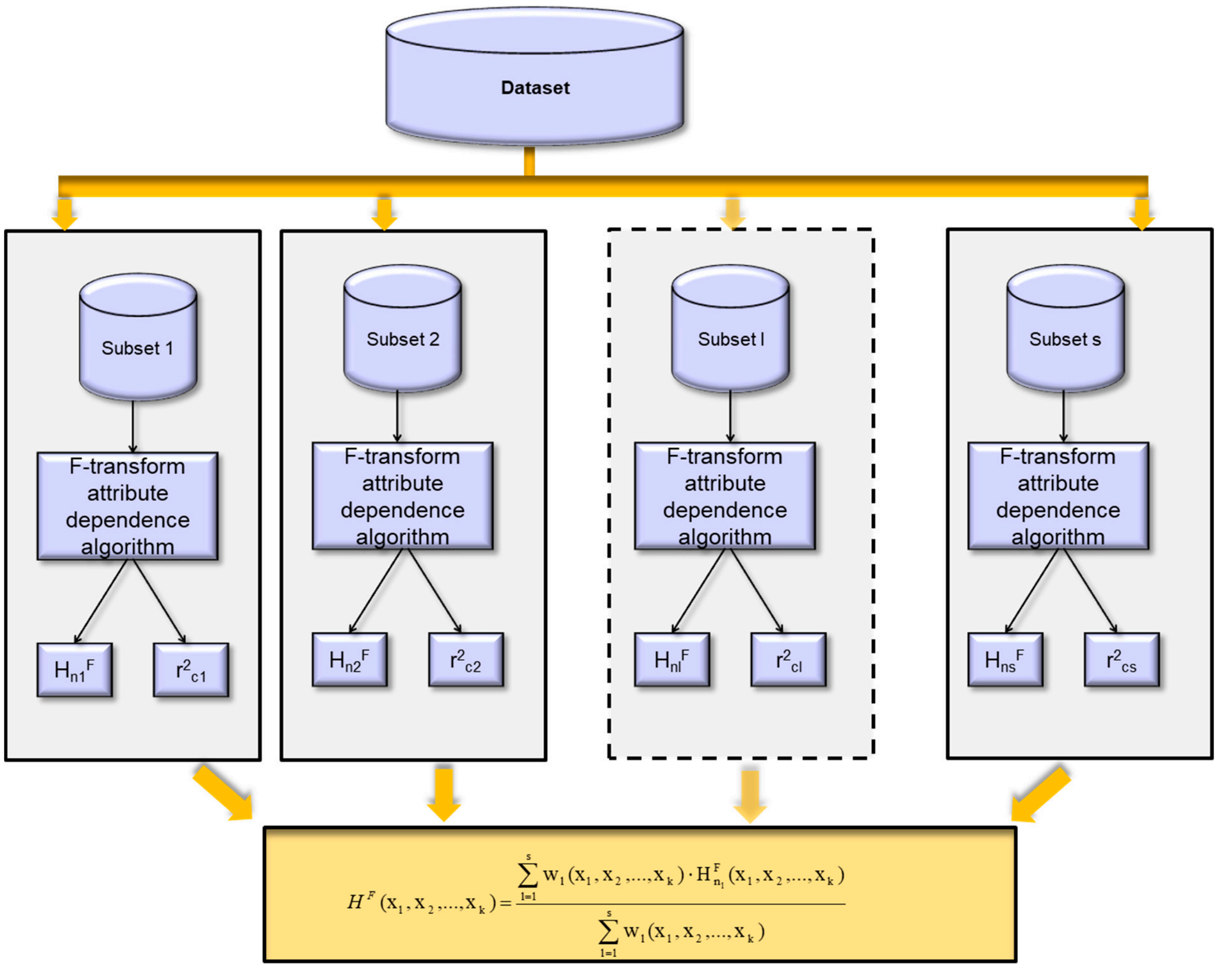


**Figure 4.** Schema of the MFAD method proposed in [12].

To test the MFAD algorithm in [12] it was applied on a large dataset given by the Italian National Statistical Institute census database with 140 numerical features related to census characteristics and measured for all the 402,678 Italian census tracts enclosed. In their tests the authors execute the MFAD algorithm by varying the number s of subsets and compare the results with those ones obtained by applying the classical F-transform attribute dependency algorithm [11] to the entire dataset (s = 1). Table 2 show the final index of determinacy obtained by applying MFAD to explore the dependency of $X_z$ = Families in owned residences on the attribute $X_1$ = Resident population with job or capital income, setting a threshold $\alpha = 0.8$.

**Table 2.** Values of the index of determinacy applying MFAD for different values of the parameter s [12].

| s | Index of Determinacy |
|---|---|
| 1 | 0.881 |
| 8 | 0.872 |
| 9 | 0.872 |
| 10 | 0.874 |
| 11 | 0.875 |
| 13 | 0.877 |
| 16 | 0.878 |
| 20 | 0.878 |
| 26 | 0.875 |
| 40 | 0.872 |

Table 1 shows the value of the index of determinacy obtained for different values of the parameter s. The value of the index of determinacy obtained by running the attribute dependency algorithm on the entire dataset (s = 1) is 0.881. All the values of the resulting index of determinacy obtained applying MFAD with different number of datasets (from s = 8 to s = 40) are comparable with this value.

The results of tests performed in [12] on large datasets show that the performances are comparable with the ones obtained using the well-known Support Vector Regression (SVM) and Multilayer Perceptron (MLP) regression methods.

### *3.2. Multi-Dimensional F-Transform Techniques for Mining Association Rules*

In [10] a method based on the multi-dimensional F-transform for mining association rules in the data is proposed. The inverse multi-dimensional F-transform (14) applied to find a dependency of the attribute $X_z$ to the attribute $X_1, \ldots, X_k$ in the form $X_z = H(X_1 \ldots X_k)$ can be used to mine association rules.

However, unlike to the functions describing dependency between attributes, mining associations are fuzzy functions which establish a correspondence between universes of fuzzy sets.

Let $U_1, \ldots, U_k$ be the domains of k attributes partitioned by fuzzy sets: a mining association functionally joins some fuzzy sets from partitions of $U_1 \ldots U_k$ with fuzzy sets over respective F-transform components.

Let $\{A_{ih_1}, \ldots, A_{ih_i}, \ldots, A_{ih_i}\}$ be an uniform fuzzy partition of the domain of the i*th* attribute $X_i$ constructed as basic functions of this domain. The fuzzy partition is obtained on the $n_i$ nodes $x_{i1}, \ldots, x_{in_i}$ in the domain $U_i$.

Each association is supported by two parameters, namely the degrees of support r and confidence $\gamma$ defined below. In [10] the multi-dimensional F-transform is applied in order to discover associations rules in the following form:

$$(X_1 \ is \ A_{1h_1}) AND (X_2 \ is \ A_{2h_2}) AND \ ..AND (X_k \ is \ A_{kh_k}) \backsim^F mean(X_z) \ is \ C \quad (20)$$

where $A_{ih_i}$, i = 1, …, k, models the meaning of the linguistic expression "*approximately* $x_{h_i}$". The corresponding logic clause can be read as "*$X_i$ is approximately* $x_{h_i}$".

The label C in the consequent is one of the following linguistic expressions characterizing the ($h_1$, …,$h_k$)th component of the F-transform: Sm (small), Me (medium), Bi (big); it is eventually combined with one of the following linguistic hedges: Ex (extremely), Si (significantly), Ve (very), empty hedge, ML (more or less), Ro (roughly), QR (quite roughly), VR (very roughly). Let $O_j$, j = 1,2, …,m, be the jth data point with component ($p_{j1}$, $p_{j2}$, …, $p_{jk}$, $p_{jz}$).

To measure the strength of the fuzzy rule (20), in [10] a membership function of an induced fuzzy set on the set of m data points {$O_1$, …, $O_m$} is defined by considering the antecedent of the hth rule (20):

$$\mathrm{A_h(O_j) = A_{1h_1}(p_{j1}) \cdot \ldots \cdot A_{kh_k}(p_{jk})} \quad (21)$$

where $A_{ih_i}(p_{ji})$ is the membership degree to the fuzzy set $A_{ih_i}$of the ith attribute in the jth data point. The following value

$$r = \frac{card\{O_j | A_h(O_j) > 0\}}{m} \quad (22)$$

is called *degree of support* of the association rule (20). If $F_{h_1 h_2 \ldots h_k}$ is the ($h_1$, …,$h_k$)th component of the direct F-transform (13) and

$$f^F_{n_1 n_2 \ldots n_k}(O_j) = \sum_{h_1=1}^{n_1} \sum_{h_2=1}^{n_2} \cdots \sum_{h_k=1}^{n_k} F_{h_1 h_2 \ldots h_K} \cdot A_{1h_1}(p_{j1}) \cdot \ldots \cdot A_{kh_K}(p_{jk}) \quad (23)$$

is the inverse F-transform on the point ($p_{j1}$, …, $p_{jk}$), in (Perfilieva et al., 2008) the degree of confidence of the association rule (20) is defined as

$$\gamma = \sqrt{\frac{\sum_{j=1}^{m} \left(f^F_{n_1 n_2 \ldots n_k}(O_j) - F_{h_1 h_2 \ldots h_K}\right)^2 \cdot A_{1h_1}(p_{j1}) \cdot \ldots \cdot A_{kh_K}(p_{jk})}{\sum_{j=1}^{m} \left(p_{jz} - F_{h_1 h_2 \ldots h_K}\right)^2 \cdot A_{1h_1}(p_{j1}) \cdot \ldots \cdot A_{kh_K}(p_{jk})}} \quad (24)$$

The strength of the hth association rule is evaluated by measuring the degree of support r and the degree of confidence γ. If both the two parameters are greater or equal to a degree of support threshold and a degree of confidence threshold, respectively, the association is found.

In [10] this method is tested on a dataset of measures of air pollution produced on a road related to traffic volumes and weather conditions, collected by the Norwegian Public Roads Administration.

## 4. F-Transform Techniques for Time Series Analysis

Time series forecasting involves methods for fitting over historical data referring to measures of an observable series and using them to predict future observations.

A time series is given by a set of data measured at different times listed in time order. Let y be a measured parameter and y(t) the measure performed at the time t. A *time series* is a function y: t ∈ N → y(t)∈ R known in n regular time steps y(1),y(2), …,y(n), where y(i), i = 1,2, …,n, is the measured value of y at the ith time step.

Time series forecasting techniques assess the value of y in the n future time steps y(n + 1), ..., y(n + m), where the value y(t + 1) at the step t + 1 is evaluated as a function of the previous p+1 measured values y(t), y(t − 1), ..., y(t − p). Let y(t), t = 1,2, …,T, be a time series. It can be decomposed by following two terms:

$$\mathrm{y(t) = f(t) + r(t)} \quad (25)$$

The term *f*(*t*) is a deterministic part, called *trend*; the term *r*(*t*) is an additional random function called *residuals*, giving the random error with respect to the trend at the time t. A general model of a stationary time series y(t) as a linear function of the p+1 measured values y(t), y(t − 1), ..., y(t − p) is the *Auto-Regressive* of order p model AR(p), given by ([13,14]):

$$y(t) \;=\; \alpha_1 y(t-1) \;+\ldots+\alpha_p y(t-p) \;+\varepsilon_t \tag{26}$$

The p coefficients $\alpha_1$, …, $\alpha_p$ must satisfy some constraints and the term $\varepsilon_t$ is the statistical white noise giving the fluctuations in the observations that cannot be explained by the model.

### 4.1. One-Dimensional F-Transform Time Series Models

In [15,16] the one-dimensional F-transform is applied to approximate the trend *f*(*t*) in (25). Let {y(t), t = 1,2…,T} be a time series given by a set of data y(t) measured in T regular time intervals. Let {$t_1$ = 1, $t_2$, …, $t_n$ = T} be a set of n nodes of the interval [1,T], where $3 \le n \le T$, and {$A_1$,...,$A_n$} be the basic functions of a uniform fuzzy partition of the interval [1,T].

If the dataset given by the time series {y(t), t = 1,2…,T} is sufficiently dense with respect to this fuzzy partition, then there exists the direct one-dimensional F-transform of *f* with components

$$F_k \;=\; \frac{\sum_{i=1}^{T} y(t)\, A_k(t)}{\sum_{i=1}^{T} A_k(t)} \qquad k = 1,2\ldots,n \tag{27}$$

Let $P_k$, k = 1,...,n, be a subset of {1,2, …,T} given by the time steps t, being $A_k(t) > 0$, as

$$P_k = \{t \le T | A_k(t) > 0\} \tag{28}$$

We can decompose y(t) as:

$$y(t) = \bigvee_{k=1}^{n} (F_k + r_{tk}) \tag{29}$$

where $r_{tk}$ is the kth residual of $y_t$ with respect to $A_k$ given by

$$r_{tk} = \begin{cases} y_t - F_k & \text{if } t \in P_k \\ -\infty & \text{otherwise} \end{cases} \tag{30}$$

Based on the autoregressive model (26), in [15,16] the kth component $F_k$ is given by a linear combination of the p previous components. The trend at the kth time step is assessed by

$$F_k = \alpha_1 F_{k-1} + \alpha_2 F_{k-2} + \ldots + \alpha_p F_{k-p} \qquad k = \; p+1,\ldots,n \tag{31}$$

In [15,16] p = 3 is set as well. The calculated value for $F_n$ are used to forecast the unknown value $F_{n+1}$ as

$$F_{n+1} = \tilde{\alpha}_1 F_n + \tilde{\alpha}_2 F_{n-1} + \tilde{\alpha}_3 F_{n-3} \tag{32}$$

The values $\tilde{\alpha}_1, \tilde{\alpha}_2, \tilde{\alpha}_3$ chosen for the three coefficients $\alpha_1, \alpha_2, \alpha_3$ minimize the absolute difference between the predicted and the calculated values of $F_n$. In [15] a numerical method and a Multilayer Perceptron neural network are used to find the optimal values of the coefficients $\alpha_1$, $\alpha_2$, $\alpha_3$. In [16] a method based on fuzzy relations is proposed to find the best values of the three coefficients.

In [16] comparisons with the autoregressive model ARIMA and with other time series fuzzy-based models are performed; the MAPE and SMAPE indexes are used to measure the forecast errors; the authors showed that their F-transform-based time series prevision model has the best performances.

In [17] the one-dimensional F-transform is proposed to filter the high frequencies in the time series. A time series can be additively decomposed into three components: trend cycle, a seasonal component, and noise. The authors prove that the one-dimensional F-transform acts as a low-pass filter, removing or significantly reducing the seasonal and noise components; then, the inverse F-transform optimally approximates the trend component.

*4.2. Multi-Dimensional F-Transform Time Series Model*

In [17] a time series forecasting model based on the multi-dimensional F-transform is proposed. The authors applied their method to the well-known Mackey-Glass time series generated by the differential equation:

$$\frac{dy}{dt} = \frac{0.2 \cdot y(t-\tau)}{1 + y^{10}(t-\tau)} - 0.1 \cdot y(t) \tag{33}$$

In [18] the function $y(t)$ is approximated by previous t-6 values $y(t-6)$, $y(t-5)$, …, $y(t-1)$ by constructing a multi-dimensional F-transform to approximate the output variable y as a function of six variable $x_i = y(t-i)$, i = 1, …,6.

To construct the components of the direct multi-dimensional F-transform the N points $\left(x_1^{(j)}, x_2^{(j)}, \ldots, x_6^{(j)}, y^{(j)}\right)$ are considered, where j = 1, …, N. They are given by

$$F_{h_1 h_2 \ldots h_6} = \frac{\sum_{j=1}^{N} y^{(j)} \cdot A_{1h_1}(x_1^{(j)}) \cdot \ldots \cdot A_{6h_6}(x_6^{(j)})}{\sum_{j=1}^{N} A_{1h_1}(x_1^{(j)}) \cdot \ldots \cdot A_{6h_6}(x_6^{(j)})} \tag{34}$$

The inverse F-transform is given by

$$f^F_{n_1 n_2 \ldots n_6}(x^{(j)}_{1\ \ 1}, x_2^{(j)}, \ldots, x_6^{(j)}) = \sum_{h_1=1}^{n_1} \sum_{h_2=1}^{n_2} \cdots \sum_{h_6=1}^{n_6} F_{h_1 h_2 \ldots h_6} \cdot A_{1h_1}(x_1^{(j)}) \cdot \ldots \cdot A_{6h_6}(x_6^{(j)}) \tag{35}$$

To assess the value of the function y(t) at the time t considering the value obtained in the six previous time steps: $x_i = y(t-i)$ i = 1, …,6, the formula (35) is applied by obtaining the following:

$$\tilde{y} = f^F_{n_1 n_2 \ldots n_6}(x_1, x_2, \ldots, x_6) = \sum_{h_1=1}^{n_1} \sum_{h_2=1}^{n_2} \cdots \sum_{h_6=1}^{n_6} F_{h_1 h_2 \ldots h_6} \cdot A_{1h_1}(x_1) \cdot \ldots \cdot A_{6h_6}(x_6) \tag{36}$$

In [18] the authors compare the results obtained by applying this method to the Mackey-Glass time series with those ones obtained by using the well-known Wang and Mendel method and with the results obtained using a local Wavelet Neural Network with three layers, six input nodes, 10 hidden nodes and one output node. They measure the MAPE, RMSE and MADMEAN indices, showing that the multi-dimensional time series method has the best performances.

The multi-dimensional fuzzy transform method [18] can be generalized for any function considering a dependency on k input parameters. In [19] it is applied for forecasting problems in spatial analysis. The framework proposed in [19] is schematized in Figure 5.

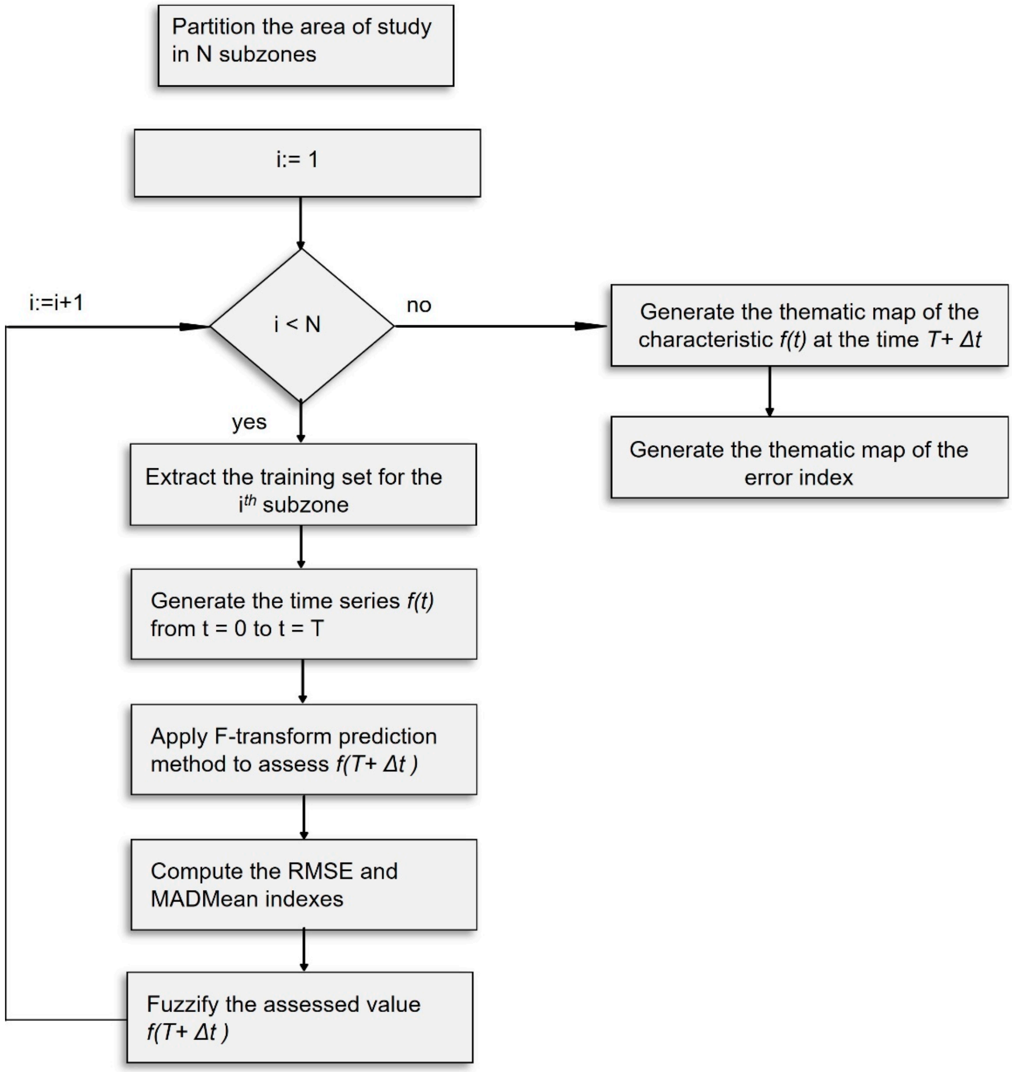


**Figure 5.** Schema of the framework proposed in [19].

The area of study is partitioned in subzones. For each subzone a training dataset with the measure of characteristics of the subzone in a specified period is extracted. Then, the time series correspondent to a measured characteristic $f$(t) from a time t = 0 to t = T is constructed and the multi-dimensional F-transform prediction method [17] is applied to assess the value of $f$ at the time T + Δt. The RMSE and the MADMEAN are used to evaluate the performances of the forecasting model. Finally, two thematic maps of the predicted value of the characteristic at the time T + Δt and of the prediction error in each subzone are given after performing a fuzzification process. This approach is encapsulated in a Geographical Information System and is tested in [19] to analyze the demographical balance data measured every month in the period 01/01/2003–31/10/2014 in the municipalities of *Cilento and Vallo di Diano National Park* located in the province of Salerno (Italy). The birth-rate and death-rate in November 2014 in each municipality are evaluated. The mean RMSE obtained is under 0.01.

### *4.3. F-Transform Seeasonal Time Series Model*

In some time series a phenomenon called *seasonality* is present, given by a repetitive and regular pattern of changes that repeats over S time periods. For example, in a monthly time series S = 12, in an hourly time series S = 24, and so on.

Some well-known statistical models as the Seasonal Auto Regressive Integrated Moving Average (SARIMA) models [20,21] are used to forecast the value of the output variable at a time t as a combination of the trend with a seasonal component.

In [22] a seasonal time-series forecasting method based on F-transforms is proposed as Time Series Seasonal F-transform (TSSF). A polynomial best fit is applied to extract the trend; then the data are de-trended, subtracting the trend from the time series and the de-treated time series is partitioned in S subsets. The one-dimensional F-transform is applied to each subset to assess the correspondent seasonality.

To assess the value of the output variable y at the time t included in the s*th* season, with s in {1,2, …,S}, we calculate the inverse F-transform $f_{n(s)}^{F}(t)$.

Let $\{(t^{(1)}, y^{(1)}), (t^{(2)}, y^{(2)}) \ldots (t^{(M_s)}, y^{(M_s)})\}$ be the de-treated s*th* subset with cardinality $M_s$, where $y^{(j)}$, j = 1, …, $M_s$, is given by difference between the original measure obtained at the time $t^{(j)}$ and the trend calculated at that time.

Let $F_h$, where h = 1,2, …,n(s), be the hth component of the one-dimensional direct F-transform calculated by using a fuzzy partition of n(s) basic functions of the domain of the sth subset. The one-dimensional inverse F-transform calculated at the time t is given by

$$f_{n(s)}^{F}(t) = \sum_{h=1}^{n(s)} F_h \cdot A_h(t) \cdot \quad (37)$$

The forecasted value $\tilde{y}_0(t)$ of the output $y_0$ at the time t included in season s is

$$\tilde{y}_0(t) = f_{n(s)}^{F}(t) + \text{trend}(t) \quad (38)$$

where the term *trend*(*t*) is the assessed value of the trend of the time series at the time t.

In the TSSF model, to verify that each subset of data is sufficiently dense with respect to the fuzzy partition and to find the best fuzzy partition, is applied the technique proposed in [11]. To find the best fuzzy partition for each subset the MADMEAN measure is calculated, being

$$MADMEAN_S = \frac{\sum_{j=1}^{M_s} |f_{n(s)}^{F}(t^{(j)}) - y^{(j)}|}{\sum_{j=1}^{M_s} y^{(j)}} \quad (39)$$

The number of fuzzy sets of the initial fuzzy partition is set to 3; then, the sufficient density of the data with respect to the fuzzy partition is verified and the direct F-transform is calculated. The inverse F- transform in each time $t^{(j)}$, where j = 1, …, $M_s$, is calculated by formula (37) and, finally, the MADMEAN index (39) is measured. If the MADMEAN index is greater than a fixed threshold, then the process stops and the direct F-transform components are stored; otherwise, the number of fuzzy sets of the fuzzy partition n(s) is increased by one unit and the previous steps are iterated. This process is executed for each seasonal subset.

In Figure 6 the flow diagram of the TSSF model is shown.

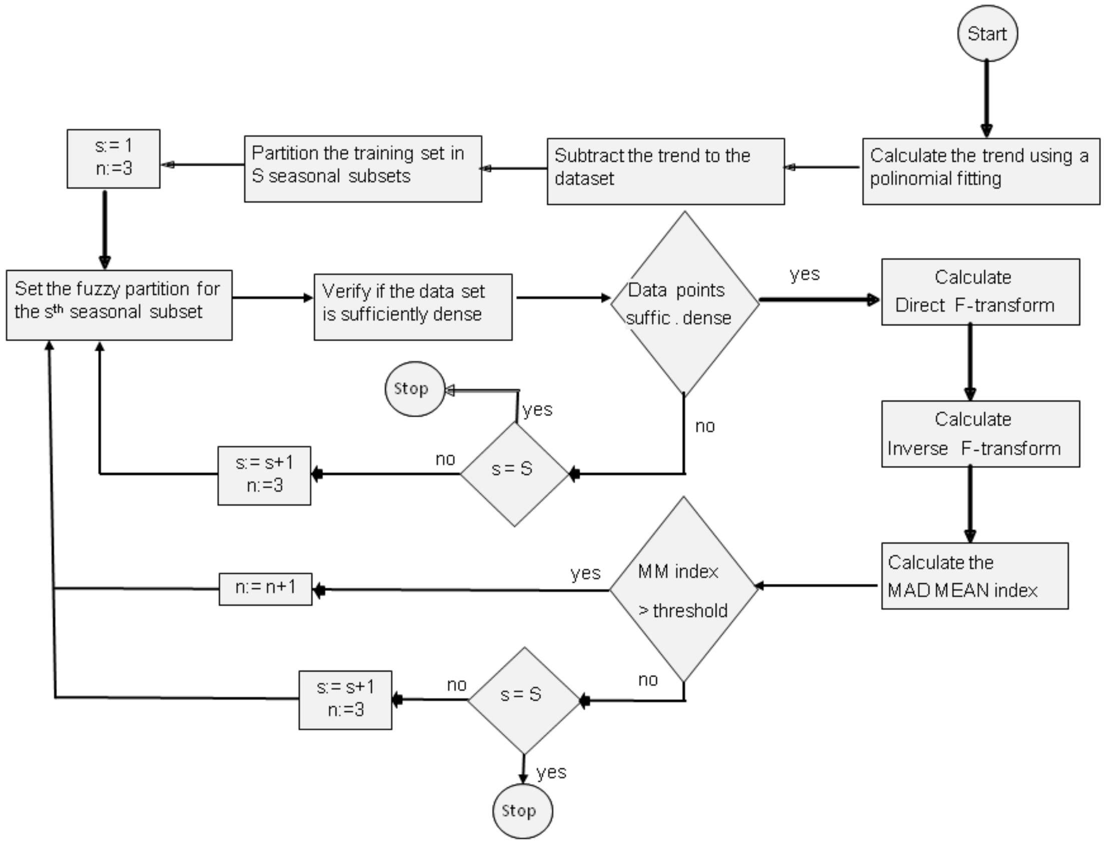


**Figure 6.** Flow diagram of the TSSF model in [22].

In [22] many comparison tests are performed comparing the performance of TSSF with the ones measured executing other forecasting algorithms applied to seasonal time series. Comparisons are executed with respect to the statistical Average Seasonal Variation (avgSV) and Seasonal ARIMA models [21], the model based on the multi-dimensional F-transform (MF-tr) [18] and the soft computing forecasting models Support Vector Machine (SVM) [23] and Automatic Design of Artificial Neural Networks (ADANN) [24]. Table 3 shows the RMSE obtained applying these models on a set of 14 seasonal time series giving the daily mean temperature measured by 14 weather monitoring stations located in the province of Genova (Italy). In each experiment, the month is used as seasonality and each dataset is partitioned in twelve subsets.

**Table 3.** RMSE in six methods for the mean temperature in 14stations in the province of Genova (Italy).

| **Station** | **RMSE** | | | | | |
|---|---|---|---|---|---|---|
| | **avgSV** | **SARIMA** | **MF-tr.** | **TSSF** | **SVM** | **ADANN** |
| Alpe Gorreto | 2.98 | 1.20 | 1.49 | 0.84 | 0.81 | 0.83 |
| Campo Ligure | 2.74 | 1.09 | 1.34 | 0.76 | 0.71 | 0.76 |
| Barbagelata | 3.25 | 1.30 | 1.57 | 0.89 | 0.84 | 0.90 |
| Camogli | 3.39 | 1.38 | 1.68 | 0.95 | 0.88 | 0.86 |
| Campo ligure | 3.02 | 1.20 | 1.49 | 0.83 | 0.77 | 0.79 |
| Carlasco | 2.91 | 1.15 | 1.42 | 0.80 | 0.77 | 0.76 |
| Chiavari | 2.78 | 1.12 | 1.39 | 0.78 | 0.73 | 0.77 |
| Genova Bolzaneto | 2.95 | 1.16 | 1.41 | 0.81 | 0.77 | 0.75 |
| Genova Pegli | 3.34 | 1.29 | 1.64 | 0.94 | 0.89 | 0.88 |
| Panesi | 3.20 | 1.29 | 1.56 | 0.87 | 0.84 | 0.83 |
| Rapallo | 2.71 | 1.08 | 1.33 | 0.75 | 0.78 | 0.84 |

| | | | | | | |
|---|---|---|---|---|---|---|
| Rovegno | 2.94 | 1.18 | 1.45 | 0.82 | 0.82 | 0.80 |
| Tigliolo | 3.06 | 1.24 | 1.52 | 0.85 | 0.80 | 0.85 |
| Viganego | 3.17 | 1.28 | 1.57 | 0.88 | 0.82 | 0.83 |

The results in Table 2 show that the TSSF's performances are better than the ones obtained by using the avgSV, SARIMA and F-transform and comparable with those ones obtained by using SVM and ADANN. In addition, SVM and ADANN are computationally more complex to manage than TFSS. A critical point of TSFF is its inability to manage irregular time series, in which it is complex to evaluate time series patterns in the data.

In [9] an extension of the TFSS model has been proposed, based on the use of the first-order F-transform. This model improves the performance of the TFSS model but increases its computational complexity.

## 5. F-Transform in Data Classification

In Section 3 we analyzed techniques that use the multi-dimensional F-transform as a regression function to explore dependency between data ([10,11]). In [25] a classification method based on the use of the multi-dimensional F-transform is proposed. The proposed algorithm, called MFC (Multi-dimensional F-transform Classification), compute the direct and inverse multi-dimensional F-transforms to classify data points.

The learning dataset is given by a set of data points characterized by a pair (**X**,Y), where **X** is a vector of s numerical features ($X_1$,…$X_s$) and ***Y*** is the class feature designated as class which has C categories, labelled with the values 1,2, …,C.

The multi-dimensional F-transform is applied to explore a relation between attributes in the form:

$$Y = f(x_1, \dots, x_s) \tag{40}$$

where *f* is a discrete function *f*: $[a_1,b_1] \times [a_2,b_2] \times \ldots \times [a_s,b_s] \rightarrow \{1,2,\ldots,C\}$ with $x_i \in [a_i,b_i]$ i = 1, …,s, and $Y \in \{1,2,\ldots,C\}$.

MFC uses the multi-dimensional inverse F-transform to approximate the function *f*. To avoid the over-fitting problem is applied the K-fold cross validation resampling algorithm to control this presence.

K-fold cross validation is a well-known resampling technique in which the dataset is partitioned into K subsets of equal size called folds. The classification algorithm is iterated K times. At any iteration of a fold constitutes the validation set and the union of the other K-1 folds forms the training set, used to train the classifier. With respect to other resampling techniques, K-fold is more efficient in dealing with the over-fitting problem, as in K-fold each fold is treated once as a validation set.

Let P = ($p_1$, $p_2$, …, $p_s$) be a data point. Formally, if $\boldsymbol{F_k}$ is the multi-dimensional direct F-transform calculated by using the k*th* fold and $f^{F_k}_{n_1 n_2 \dots n_s}(p_1, p_2, \ldots, p_s)$ is the value of the multi-dimensional inverse F-transform calculated in P, then, an average of the K inverse F- transforms in the point P is calculated as

$$f_{n_1 n_2 \dots n_s}(p_1, p_2, \ldots, p_s) = \frac{1}{K}\sum_{k=1}^{K} f^{F_k}_{n_1 n_2 \dots n_s}(p_1, p_2, \ldots, p_s) \tag{41}$$

The point P is classified in the class labeled c*, where

$$c^* = \arg\left\{\min_{c=1,\ldots,C}\left(\left|f_{n_1 n_2 \dots n_s}(p_1, p_2, \ldots, p_s) - c\right|\right)\right\} \tag{42}$$

To evaluate the performance of the classifier for each fold two index $CV_1^k$ and $CV_2^k$ k = 1, …,K are calculated, where

- $CV_1^k$ is the percentage of all the misclassified data points in the kth training set;
- $CV_2^k$ is the percentage of all the misclassified data points in the kth validation set.

The final index giving the average of the percentage of misclassified data points in the training sets is

$$CV_1 = \frac{1}{K}\sum_{k=1}^{K} CV_1^k \tag{43}$$

and the final index giving the average of the percentage of misclassified data points in the validation sets is

$$CV_2 = \frac{1}{K}\sum_{k=1}^{K} CV_2^k \tag{44}$$

$CV_1$ and $CV_2$ are used to evaluate the performances of MFC. If $CV_1$ is under a fixed threshold $\alpha$ and $CV_2$ is under a fixed threshold $\beta$, then the algorithm stops, else a finer set of fuzzy partitions of the domains of the s input variables is constructed and the process is iterated.

In Figure 7 we show the flow diagram of MFC.

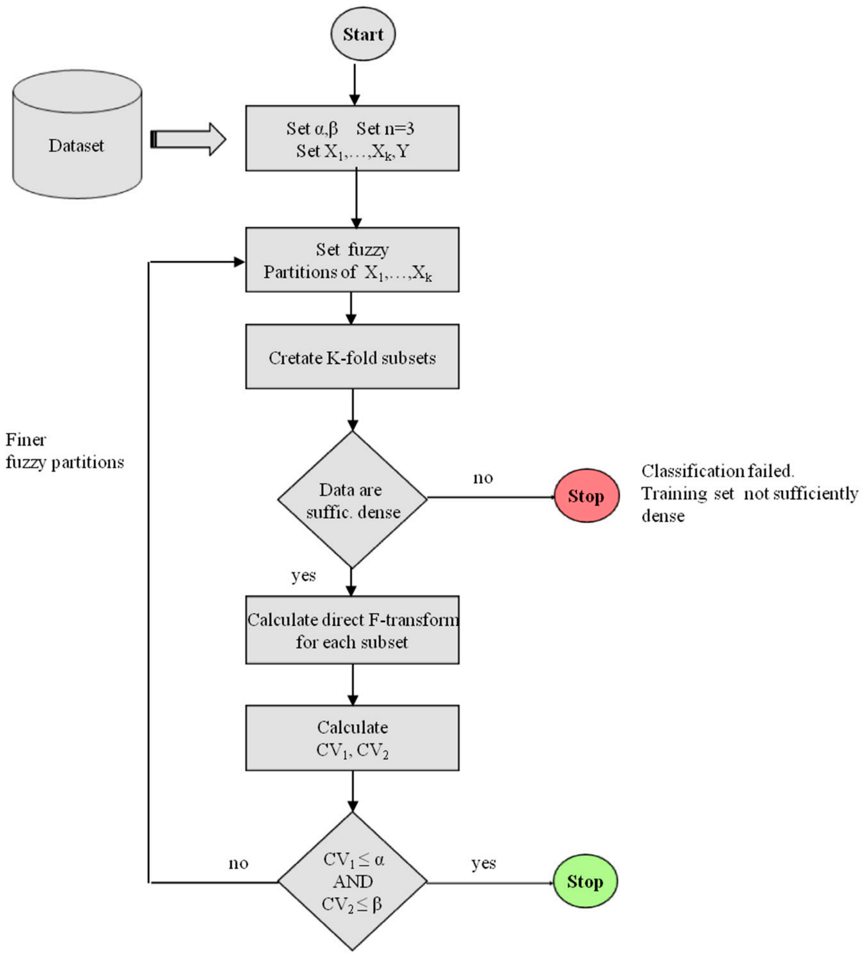


**Figure 7.** Flow diagram of the MFC algorithm [25].

In [25] comparison tests are performed on over 100 classification datasets extracted from the University of California, Irvine (for short, UCI) Machine Learning and from the Knowledge Extraction Evolution Learning repositories.

In Table 4 are shown the mean accuracy, precision and recall classification measures obtained by running MFC, Decision tree-based J48 [26], Multi-Layer Perceptron [27], naive Bayes [28] and Lazy K-Nearest Neighbor IBK [29].

**Table 4.** Mean accuracy, precision and recall with 5 classification algorithms.

| Algorithm | Accuracy | Precision | Recall |
|---|---|---|---|
| MFC Classifier | 98.15% | 98.09% | 97.36% |
| Decision tree J48 | 98.38% | 98.17% | 97.51% |
| Multilayer Perceptron | 98.22% | 98.23% | 97.48% |
| Naive Bayes | 96.55% | 91.89% | 90.65% |
| Lazy IBK | 97.17% | 93.30% | 91.44% |

These results show that MFC provides classification performance better than those ones obtained by using the naive Bayes and Lazy IBK algorithms. They are comparable with the results obtained by the Decision tree J48 and the Multilayer Perceptron algorithms.

A weak point of MFC algorithm is its high computational complexity which makes it unsuitable to manage massive and high-dimensional datasets.

The integration with data compression and feature selection approaches in the preprocessing phase can reduce these high computational costs. An approach that integrates Principal Component Analysis (PCA) feature reduction techniques with higher-degree F-transform has been proposed in [30] in image classification. A mixed model that integrates higher-degree F-transform and PCA techniques could be tested in data classification to reduce the number of features and improve the accuracy and precision of the classifier model, without significantly increasing the time consumption.

## 6. Conclusions

This paper presents a summary of the data analysis techniques proposed in the literature based on the use of the F-transform in one or more dimensions. We initially presented the definition of one-dimensional direct and inverse F-transform, showing how it can be used to approximate a continuous function on a real interval. We then extended this concept to the multi-dimensional F-transform, showing how it can be used in regression analysis. In particular, attention was paid to the constraint of sufficient data density with respect to fuzzy partitions, which is extremely important for the choice of the optimal cardinality of fuzzy partitions. Then, the methods proposed in the literature for the analysis of the dependency between attributes in the data and for the extraction of association rules through the use of direct and inverse multi-dimensional F-transforms were presented and analyzed. An extensive discussion was devoted to the different time series analysis techniques based on the F-transforms proposed in the literature. Finally, a classification method recently presented in the literature based on the multi-dimensional F-transform was described.

The use of F-transform-based approaches in data analysis still remains an evolving research field. We foresee that in the future new approaches based on the use of the F-transform may be presented that reduce the time-consumption and computational complexity that currently, on the one hand, prevent the application of these techniques to massive and high dimensional data and on the other hand allow to also use high-orders F-transforms in data analysis, improving the performance obtained using the zero-order F-transform. In the future, hybrid strategies of using the high-order F-transform and reducing the data size could lead to an optimal trade-off between the quality of the results and the processing times.

In the future, the multidimensional zero and high-order fuzzy transform methods may be included into soft computing hybrid models for the analysis of risk prediction and damage assessment proposed in recent soft computing risk analysis and forecasting models such as damage assessment of existing buildings [31] and entity assessment of the damage that can be produced on them by seismic events [32]. Moreover, fuzzy transform methods can be applied for the solution of fuzzy differential equations [33] and fuzzy partial equations [34] in data analysis models for complex systems.

**Author Contributions:** The contributions of the three authors F.D.M., I.P. and S.S. are summarized below: conceptualization, F.D.M., I.P. and S.S..; methodology, F.D.M., I.P. and S.S.; software, F.D.M., I.P. and S.S.; validation, F.D.M., I.P. and S.S.; formal analysis, F.D.M., I.P. and S.S.; investigation, F.D.M., I.P. and S.S.; resources, F.D.M., I.P. and S.S..; data curation, F.D.M., I.P. and S.S.; writing—original draft preparation, F.D.M., I.P. and S.S.; writing—review and editing, F.D.M., I.P. and S.S.; visualization, F.D.M., I.P. and S.S.; supervision, F.D.M., I.P. and S.S. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Acknowledgments:** The work of Irina Perfilieva was partially supported by the project AI-Met4AI, CZ.02.1.01/0.0/0.0/17-049/0008414.

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

## Appendix A. Table of Acronyms and Abbreviations

In Table A1 are listed the acronyms and abbreviation terms used in the text.

**Table A1.** Acronyms and abbreviations.

| Acronym/Abbreviation | Explanation |
|---|---|
| F-transform | Fuzzy transform |
| Multidimensional F-transform | Multi-dimensional Fuzzy transform |
| MFAD | Massive F-transform Attribute Dependency method |
| SVM | Support Vector regression Method |
| MLP | MultiLayer Perceptron method |
| avgSV | AVeraGe Seasonal Variation model |
| SARIMA | Seasonal AutoRegressive Integrated Moving Average model |
| MF-tr | Multi-dimensional Fuzzy TRansform forecasting model |
| TFSS | Time Series Seasonal time series F-transform model |
| ADANN | Automatic Design of Artificial Neural Networks model |
| MFC | Multidimensional F-transform Classification method |
| UCI | University of California, Irvine |
| K-fold | Cross-validation K-fold resampling method applied in classification. |
| Naïve Bayes | Naïve Bayesian classification method |
| J48 | Decision tree J48 classification algorithm in the Weka data mining tool. |
| Lazy IBK | Lazy K-Nearest Neighbor Instance-Bases learning with parameter K classification method. |
| PCA | Principal Component Analysis. |

## References

1. Perfilieva, I.; Haldeeva, E. Fuzzy transformation. In *IFSA World Congress and 20th NAFIPS International Conference, Joint 9th, Vancouver, Canada, 25-28 July 2001*, IEEE, Piscataway (NJ) USA, 2001; Volume 4, pp. 1946–1948.
2. Perfilieva, I. Fuzzy transforms: Theory and applications. *Fuzzy Sets Syst.* **2006**, *157*, 993–1023.
3. Di Martino, F.; Sessa, S. *Fuzzy Transforms for Image Processing and Data Analysis. Core Concepts, Processes and Applications*; Springer: Cham, Switzerland, 2020; p. 217, doi:10.1007/978-3-030-44613-0.
4. Bede, B.; Rudas, I.J. Approximation properties of fuzzy transforms. *Fuzzy Sets Syst.* **2011**, *180*, 20–40.
5. Khastan, A. A new representation for inverse fuzzy transform and its application. *Soft Comput.* **2017**, *21*, 3503–3512.
6. Perfilieva, I.; Dankova, M.; Bede, B. Towards a higher degree f-transform. *Fuzzy Sets Syst.* **2020**, *180*, 3–19.
7. Alikhani, R.; Zeinali, M.; Bahrami, F.; Shahmorad, S.; Perfilieva, I. Trigonometric $f^m$-transform and its approximative properties. *Soft Comput.* **2017**, *21*, 3567–3577.
8. Zeinali, M.; Alikhani, R.; Bahrami, F.; Shahmorad, S.; Perfilieva, I. On the structural properties of $f^m$-transform with application. *Fuzzy Sets Syst.* **2018**, *342*, 31–52.

9. Di Martino, F.; Sessa, S. Seasonal Time Series Forecasting by F$^1$-Fuzzy Transform, Special Issue Intelligent Systems in Sensor Networks and Internet of Things. *Axioms* **2019**, *19*, 3611.
10. Perfilieva, I.; Novàk, V.; Dvoràk, A. Fuzzy transforms in the analysis of data. *Int. J. Approx. Reason.* **2008**, *48*, 36–46.
11. Di Martino, F.; Loia, V.; Sessa, S. Fuzzy transforms method and attribute dependency in data analysis. *Inf. Sci.* **2010**, *180*, 493–505.
12. Di Martino, F.; Sessa, S. Attribute dependency data analysis for massive datasets by fuzzy transforms. *Soft Comput*. **2021**, doi:10.1007/s00500-021-05760-y.
13. Wold, H.O.A. *A Study in the Analysis of Stationary Time Series*; Almqvist, Wiksel, Eds,. Stockholm:, Sweden, 1938; p. 214, doi:10.1111/j.2397-2335.1939.tb01266.x.
14. Wei, W.W.S. *Time Series Analysis Univariate and Multivariate Methods*, 2nd ed.; Pearson Addison Wesley:, Boston (MA), USA 2006; p. 605, ISBN: 0-321-32216-9.
15. Perfilieva, I.; Yarushkina, N.; Afanasieva, T. *Time Series Analysis by Discrete F-Transform, Conference Paper in IEEE International Conference on Fuzzy Systems, July 2010*; p. 4, doi:10.1109/FUZZY.2010.5584264.
16. Perfilieva, I.; Yarushkina, N.; Afanasieva, T.; Romanov, A. Time series analysis using soft computing methods. *Int. J. Gen. Syst.* **2013**, *42*, 687–705.
17. Novàk, V.; Perfilieva, I.; Kreinovich, V. Filtering out high frequencies in time series using F-transform. *Inf. Sci.* **2014**, *274*, 192–209.
18. Di Martino, F.; Loia, V.; Sessa, S. Fuzzy transforms method in prediction data analysis. *Fuzzy Sets Syst.* **2011**, *180*, 146–163.
19. Di Martino, F.; Sessa, S. Fuzzy transform prediction in spatial analysis and its application to demographic balance data*. Soft Comput*. **2017**, *21*, 3537–3550.
20. Ziegel, E.R.; Box, G.E.P.; Reinsel, G.C.; Jenkins, S. Time Series Analysis, Forecasting, and Control. *Technometrics, Taylor & Francis, Milton Park (UK),* **1995**, *37(2)*, 238-239.
21. Box, G.E.P.; Jenkins, G.M.; Reinsel, G.C. *Time Series Analysis: Forecasting and Control*, 5th ed.; Prentice Hall: Englewood Cliffs, NJ, USA, 2015; p. 712, ISBN: 978-1-118-67502-1.
22. Di Martino, F.; Sessa, S. Time series seasonal analysis based on fuzzy transforms. *Symmetry* **2017**, *9*, 281, doi:10.3390/sym911028.
23. Pai, P.F.; Lin, K.P.; Lin, C.S.; Chang, P.T. Time series forecasting by a seasonal support vector regression model. *Exp. Syst. Appl.* **2010**, *37*, 4261–4265.
24. Štepnicka, M.; Cortez, P.; Peralta Donate, J.; Štepnickova, L. Forecasting seasonal time series with computational intelligence: On recent methods and the potential of their combinations. *Exp. Syst. Appl.* **2013**, *40*, 1981–1992.
25. Di Martino, F.; Sessa, S. A classification algorithm based on multi-dimensional fuzzy transforms. *Ambient Intell. Humaniz. Comput.* **2021**, doi:10.1007/s12652-021-03336-0.
26. Bhargawa, N.; Sharma, G.; Bhargava, R.; Mathuria, M. Decision Tree Analysis on J48 Algorithm for Data Mining. *Int. J. Adv. Res. Comput. Sci. Softw. Eng.* **2013**, *3*, 1114–1119.
27. Pal, S.K.; Mitra, S. Multilayer perceptron, fuzzy sets, and classification. *IEEE Trans. Neural Netw.* **1992**, *3*, 683–697.
28. Murphy, K.P. *Machine Learning: A Probabilistic Perspective (Adaptive Computation and Machine Learning Series)*, 1st ed; The MIT Press: London, UK, 2012; p. 1070, ISBN: 978-0262018029.
29. Aha, D.W., Ed. *Lazy Learning*; Kluwer Academic Publishers: Norwell, MA, USA, 1997; p. 436, ISBN: 978-0792345848.
30. Hurtik, P.; Molek, V.; Perfilieva, I. Novel dimensionality reduction approach for unsupervised learning on small datasets. *Pattern Recognit.* **2020**, *103*, 107291.
31. Harirchian, E.; Ehsan, S.; Hosseini, A.; Jadhav, K.; Kumari, V.; Rasulzade, S.; Işık, E.; Wasif, M.; Lahmer, T. A review on application of soft computing techniques for the rapid visual safety evaluation and damage classification of existing buildings. *J. Build. Eng.* **2021**, *43*, 102536.
32. Harirchian, E.; Lhamer, T. Developing a hierarchical type-2 fuzzy logic model to improve rapid evaluation of earthquake hazard safety of existing buildings. *Structures* **2020**, *28*, 1384–1399.
33. Georgieva, A. *Application of Double Fuzzy Natural Transform for Solving Fuzzy Partial Equations, AIP Conference Proceedings*; AIP Publishing: Melville, NY, USA, 2021; Volume 2333, p. 080006, doi:10.1063/5.0041603.
34. Mazandarani, M.; Xiu, L. A Review on Fuzzy Differential Equations. *IEEE Access* **2021**, *9*, 62195–62211.