Peer-reviewed version available at Symmetry 2017, 9, 281; doi:10.3390/sym9110281

Type of the Paper (Article)

Time Series Seasonal Analysis Based on Fuzzy Transforms

Ferdinando Di Martino[1,*] and Salvatore Sessa[1]

[1] Università degli Studi di Napoli Federico II, Dipartimento di Architettura, via Toledo 402, 80134 Napoli (Italy); fdimarti,sessa@unina.it

* Correspondence: fdimarti@unina.it; Tel.: +39-081-253-8907; Fax: +39-081-253-8905

Abstract: We define a new seasonal forecasting method based on fuzzy transforms. We use the best interpolating polynomial for extracting the trend of the time series and generate the inverse fuzzy transform on each seasonal subset of the universe of discourse for predicting the value of a an assigned output. Like first example, we use the daily weather dataset of the municipality of Naples (Italy) starting from data collected from 2003 till to 2015 making predictions on the following outputs: mean temperature, max temperature and min temperature, all considered daily. Like second example, we use the daily mean temperature measured at the weather station "Chiavari Caperana" in the Liguria Italian Region. We compare the results with our method, the average seasonal variation, ARIMA and the usual fuzzy transforms concluding that the best results are obtained under our approach in both examples.

Keywords: ARIMA; forecasting; fuzzy partition; fuzzy transform; time series

1. Introduction

Time series forecasting methods are quantitative techniques that analyse historical data of a variable for predicting its values. Traditional forecasting methods [1, 2, 3, 4, 14, 19, 20] use statistical techniques to estimate the future trend of a variable starting from numerical datasets. Many time series data contain seasonal patterns, that is regular, repetitive and predictable changes that happen sequentially in a period of time which could be a year, a season, a month, a week, etc.

Different approaches have been developed to deal with trend and seasonal time series. Traditional approaches, such the moving average method, additive and multiplicative models, Holt-Winters exponential smoothing, etc. [3, 4, 14, 20], use statistical methods for removing the seasonal components: they decompose the series into trend, seasonal, cyclical and irregular components [32]. Other statistical approaches are based on the Box–Jenkins model, called Autoregressive Integrated Moving Average (ARIMA) [3, 14, 20, 25]. In the first phase of the model ARIMA an auto-correlation analysis is performed for verifying if the series is non-stationary, then the series is transformed into a stationary series formed by the differences between the value at the actual moment and the value at the previous moment.

The main limitation of the seasonal ARIMA model is the fact that the process is considered to be linear. Many soft computing models have been presented in literature for capturing nonlinear characteristics in seasonal time series like, e. g., Support Vector Machine (SVM) [21, 24] used for wind speed prediction [22], air quality forecasting [18], rainfall forecasting [13]. SVM utilizes a kernel function to transform the input variables into a multi-dimensional feature space, then the Lagrange multipliers are used for finding the best hyperplane to model the data in the feature space. The main advantage of an SVM method is that that the solution is unique and there are no risk to move towards local minima, but some problems remain as the choice of the kernel parameters which influences the structure of the feature space, affecting the final solution. Another method is based on

an Artificial Neural Network (ANN) [5, 12, 32, 31]. The most widely used ANN architectures for forecasting problems are given by multi-layer Feed Forward Network (FNN) architectures [29, 30], where the input nodes are given by the successive observations of the time series, that is the target y_t is a function of the values y_{t-1}, $y_{t-2,\ldots}$, y_{t-p}, where p is the number of input nodes.

A variation of FNN is the Time Lagged Neural Network (TLNN) architecture [9,16], where the input nodes are the time series values at some particular lags. For example, in a time series having the month as seasonal period, the neural network used for forecasting the parameter value at the time t can contain input nodes corresponding to the lagged values at the time t-1, t-2,..,t-12. The key point is that an ANN can be considered a nonlinear auto-regression model. ANN's are inherently nonlinear and can model accurately complex characteristics in data patterns with respect to linear approaches such as ARIMA models. One of the main problems in the ANN forecasting models is the selection of appropriate network parameters. This operation is crucial since it affects strongly the final results. Furthermore, the presence of a high number of network parameters in the model can produce overtraining of data, giving rise to incorrect forecasting solutions.

In order to reduce the problems present in the SVM and ANN approaches, some authors have recently developed some hybrid models like, e.g., genetic algorithms and tabu search (GA/TS) [24] and the Modified Firefly Algorithm (MFA).

In [15] a Discrete Wavelet Transform (DWT) algorithm is used to decompose the time series into linear and nonlinear components, afterward the ARIMA and the ANN models are used to forecasting separately the two components.

In [27] the authors present four seasonal forecasting models: an adaptive ANN model that uses a genetic algorithm for evolving the ANN topology and the back-propagation parameter, a SVM seasonal forecasting model and two hybrid ANN and SVN models based on linguistic fuzzy rules. The authors compare the four methods with the traditional ARIMA method, showing that the results under the four methods are comparable with the ones obtained using ARIMA. The best results are obtained by using the adaptive ANN algorithm based on linguistic fuzzy rules but ANN-based methods are complex to manage and require more computational effort.

In order to overcome this high computational complexity and the heavy dependence on the input parameters, a soft computing forecasting method based on fuzzy transforms [26] (for short, F-transforms) is presented in [6]. There this method was applied to the North Atlantic Oscillation (NAO) data time series; the results are better than the ones obtained by using the well known Wang-Mendel method [28] and Local Linear Wavelet Neural Network (LLWNN) techniques.

The F-transforms are a powerful flexible method to be applied in various domains such as image compression [8], detection of attribute dependencies in data analysis [7] and extraction of the trend cycle of time series [23].

Strictly speaking, in [6] a forecasting index is calculated with respect to the data of the training set: if this index is smaller than or equal to an assigned threshold, the algorithm is stopped, otherwise it is iterated by taking a finer fuzzy partition of the variable domain. When a fuzzy partition is settled, the algorithm controls that the training dataset is sufficiently dense with respect to the fuzzy partition, that is each fuzzy set of the fuzzy partition has at least a non-zero membership degree in some point.

Here we give a forecasting method based on F-transforms for seasonal time series analysis called Time Series Seasonal F-transform (TSSF). We give a partition of the time series dataset into seasonal pattern components. A seasonal pattern is related to a fixed period of the time series fluctuations: the seasonality is set on a fixed period such as year, month, week, etc. In the TSSF model we assume that the different components affect the time series. We use the best polynomial fit for estimating the trend of the time series. After de-trending the data by subtracting the trend from the time series dataset, we find a fuzzy partition of the dataset into seasonal subsets to which we apply the F-transforms by checking that the chosen partition is optimal for the density of the training data. In [6, 7] the authors have developed this process: in particular, four forecasting indexes are proposed for assessing the quality of the results: Mean Absolute Deviation (MAD), Mean Absolute Percentage Error (MAPE), Root Mean Square Error (RMSE) and Mean Absolute Deviation Mean

Peer-reviewed version available at Symmetry 2017, 9, 281; doi:10.3390/sym9110281

(MADMEAN). An optimal fuzzy partition is found by calculating the corresponding RMSE and MADMEAN: if at least one of the two indices does not exceed a respective threshold, then the process stops and the given fuzzy partition is considered optimal.

In the TSSF method we essentially adopt the same procedure but we prefer to use the MADMEAN index because that this index is more robust than other forecasting indexes, as proved in [17]. We emphasize that the dimension of the optimal fuzzy partition can vary with the seasonal subset. In Fig. 1 the TSSF method is synthetized in detail.

In the assessment phase the best polynomial fit is applied for determining the trend in the training data. After de-trending the dataset, the time series is decomposed into S seasonal subsets to which the F-transform forecasting iterative method is applied. Initially, a coarse grained uniform fuzzy partition is fixed. If the subset is not sufficiently dense with respect to the fuzzy partition, the F-transform sub-process applied on the seasonal subset stops, else the MADMEAN index is calculated and if it is greater than an assigned threshold, the F-transform sub-process is iterated considering a finer uniform fuzzy partition. The output is given by the inverse F-transforms obtained for each seasonal subset at the end of the corresponding sub-process.

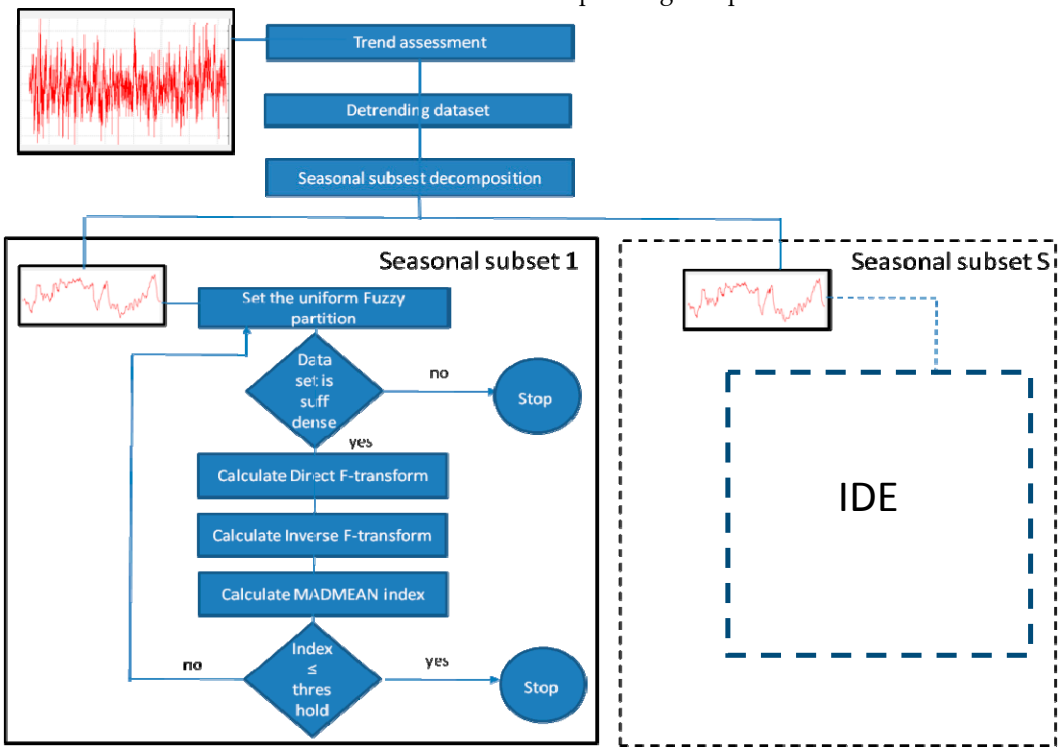


Figure 1. Schema of the TSSF method.

To forecast a value of a parameter y_0 at the time t we calculate the *sth* seasonal subset of cardinality n(s) in which the time t is inserted. Then we consider the *sth* inverse F-transform $f^F_{n(s)}(t)$. The forecasted value of the parameter is given by the formula $f^F_{n(s)}(t) + trend\ (t)$, where *trend(t)* is the value of the polynomial trend at the time t. Fig. 2 illustrates the application of the TSSF for forecasting analysis.

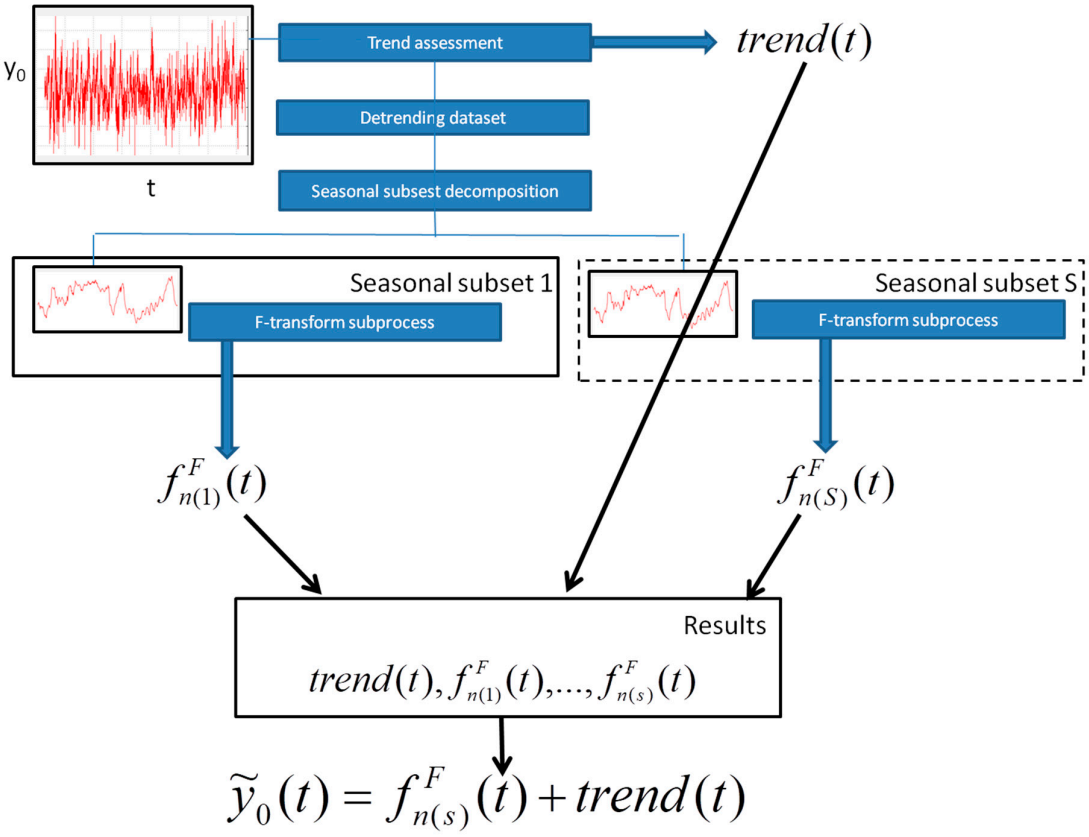


Figure 2. The TSSF method for forecasting analysis.

For sake of completeness, we recall the concept of F-transform in Section 2. Section 3 contains the F-transform-based prediction method, in Section 4 we describe our TSSF method applied on the climate dataset of the municipality of Naples (www.ilmeteo.it/portale/archivio-meteo/Napoli) whose results are given in Section 5 that containin also comparisons with average seasonal variation, ARIMA and traditional prediction F-transform methods. Section 6 reports the final conclusions.

2. Direct and inverse f-transform

Following the definitions and notations of [26], let $n \geq 2$ and $x_1, x_2, \ldots, x_n$ be points of [a,b], called nodes, such that $x_1 = a < x_2 < \ldots < x_n = b$. The family of fuzzy sets $A_1,\ldots,A_n : [a,b] \rightarrow [0,1]$, called basic functions, is a fuzzy partition of [a,b] if the following hold:

- $A_i(x_i) = 1$ for every $i = 1,2,\ldots,n$;
- $A_i(x) = 0$ if $x \in]x_{i-1}, x_{i+1}[$ for $i = 2,\ldots,n-1$;
- $A_i(x)$ is a continuous function on [a,b];
- $A_i(x)$ strictly increases on $[x_{i-1}, x_i]$ for $i = 2,\ldots, n$ and strictly decreases on $[x_i, x_{i+1}]$ for $i = 1,\ldots, n-1$;
- $A_1(x) + \ldots + A_n(x) = 1$ for every $x \in [a,b]$.

The fuzzy sets $\{A_1(x),\ldots,A_n(x)\}$ form an uniform fuzzy partition if

- $n \geq 3$ and $x_i = a + h\cdot(i-1)$, where $h = (b-a)/(n-1)$ and $i = 1, 2, \ldots, n$ (that is the nodes are equidistant);
- $A_i(x_i - x) = A_i(x_i + x)$ for every $x \in [0,h]$ and $i = 2,\ldots, n-1$;
- $A_{i+1}(x) = A_i(x - h)$ for every $x \in [x_i, x_{i+1}]$ and $i = 1,2,\ldots, n-1$.

Here we only interested in the discrete case, that is to functions f defined on the set P of points $p_1,...,p_m$ of [a,b]. If P is sufficiently dense with respect to the fixed partition $\{A_1, A_2, \ldots, A_n\}$, that is for each $i \in \{1,\ldots,n\}$ there exists an index $j \in \{1,\ldots,m\}$ such that $A_i(p_j) > 0$, we can define the n-tuple $\{F_1, F_2, \ldots, F_n\}$ as the discrete direct F-transform of f with respect to the basic functions $\{A_1, A_2, \ldots, A_n\}$, where F_i is given by

$$F_i = \frac{\sum_{j=1}^{m} f(p_j) A_i(p_j)}{\sum_{j=1}^{m} A_i(p_j)} \tag{1}$$

for i=1,…,n. Similarly we define the discrete inverse F-transform of f with respect to the basic functions $\{A_1, A_2, \ldots, A_n\}$ by setting

$$f_n^F(p_j) = \sum_{i=1}^{n} F_i A_i(p_j) \tag{2}$$

for every $j \in \{1,\ldots,m\}$. We can extend the above concepts to functions in k (≥2) variables. In the discrete case, we assume that the function $f(x_1,x_2,\ldots,x_k)$ is defined on m points $p_j = (p_{j1}, p_{j2},\ldots,p_{jk}) \in [a_1,b_1]\times[a_2,b_2]\times\ldots\times[a_k,b_k]$ for j=1,…,m. We say that $P=\{(p_{11}, p_{12}, \ldots, p_{1k}), \ldots, (p_{m1}, p_{m2}, \ldots, p_{mk})\}$ is sufficiently dense with respect to the fuzzy partitions

$$\{A_{11}, A_{12},\ldots, A_{1n_1}\},\ldots,\{A_{k1}, A_{k2},\ldots, A_{kn_k}\} \tag{3}$$

of $[a_1,b_1],\ldots, [a_k,b_k]$, respectively, if for each k-tuple $\{h_1,\ldots,h_k\} \in \{1,\ldots,n_1\}\times\ldots\times\{1,\ldots,n_k\}$, there exists a point $p_j = (p_{j1}, p_{j2},\ldots,p_{jk})$ in P, j=1,…,m, such that $A_{1h_1}(p_{j1})\cdot\ldots\cdot A_{kh_k}(p_{jk}) > 0.$

In this case we define the $(h_1,h_2,\ldots,h_k)$-th component $F_{h_1h_2\ldots h_k}$ of the discrete direct F-transform of f with respect to the basic functions as

$$F_{h_1h_2\ldots h_k} = \frac{\sum_{j=1}^{m} f(p_{j1}, p_{j2},\ldots p_{jk}) \cdot A_{1h_1}(p_{j1}) \cdot A_{2h_2}(p_{j2}) \cdot\ldots\cdot A_{kh_k}(p_{jk})}{\sum_{j=1}^{m} A_{1h_1}(p_{j1}) \cdot A_{2h_2}(p_{j2}) \cdot\ldots\cdot A_{kh_k}(p_{jk})} \tag{4}$$

Now we define the discrete inverse F-transform of f with respect to the above basic functions to be the following function by setting for each point $p_j=(p_{j1}, p_{j2},\ldots,p_{jk}) \in [a_1,b_1]\times\ldots\times[a_k,b_k]$:

$$f^F_{n_1\ldots n_k}(p_{j1},\ldots, p_{jk}) = \sum_{h_1=1}^{n_1}\sum_{h_2=1}^{n_2}\ldots\sum_{h_k=1}^{n_k} F_{h_1h_2\ldots h_k} \cdot A_{1h1}(p_{j1})\ldots A_{kh_k}(p_{jk}) \tag{5}$$

for j=1,…,m. The following theorem holds [26]:

Theorem. Let $f(x_1,x_2,\ldots,x_k)$ be given on the set of points $P = \{(p_{11},p_{12}, \ldots,p_{1k}), (p_{21}, p_{22}, \ldots, p_{2k}), \ldots,(p_{m1}, p_{m2}, \ldots,p_{mk})\} \in [a_1,b_1]\times[a_2,b_2]\times \ldots\times[a_k,b_k]$. Then for every $\varepsilon > 0$, there exist k integers $n_1=n1(\varepsilon),\ldots,n_k=n_k(\varepsilon)$ and k related fuzzy partitions (3) of $[a_1,b_1], \ldots, [a_k,b_k]$, respectively, such that the set P is sufficiently dense with respect to them and for every $p_j = (p_{j1}, p_{j2},\ldots,p_{jk})$ in P, j=1,…,m, the following inequality holds:

$$\left| f(p_{j1},\ldots, p_{jk}) - f^F_{n_1\ldots n_k}(p_{j1},\ldots, p_{jk}) \right| < \varepsilon \tag{6}$$

3. F-transform forecasting method

We now describe the F-transform forecasting algorithm presented in [6]. Let M be assigned input-output pairs data ($x^{(j)}$, $y^{(j)}$) in the following form:

$$\begin{aligned}
&(\mathbf{x}^{(1)}, y^{(1)}) = (x_1^{(1)}, \ldots, x_i^{(1)} \ldots, x_n^{(1)}, y^{(1)}) \\
&\ldots \\
&(\mathbf{x}^{(j)}, y^{(j)}) = (x_1^{(j)}, \ldots, x_i^{(j)} \ldots, x_n^{(j)}, y^{(j)}) \\
&\ldots \\
&(\mathbf{x}^{(M)}, y^{(M)}) = (x_1^{(M)}, \ldots, x_i^{(M)}, \ldots, x_n^{(M)}, y^{(M)})
\end{aligned} \tag{7}$$

for j = 1,2, …,M. The task is to generate a fuzzy rule-set from the M pairs ($x^{(j)}$, $y^{(j)}$) in order to determine a mapping f from the input-space R^n into the output-space R. We assume that the $x_i^{(1)}, \ldots, x_i^{(j)}, \ldots, x_i^{(M)}$ lie in $[x_i^-, x_i^+]$ for every i=1,…,n, and $y^{(1)}$, $y^{(2)}$,…, $y^{(M)}$ in $[y^-, y^+]$. The F-transform forecasting method calculates a function $y = f(x_1, \ldots, x_n)$ which approximates the data. Like in [7], we create a partition of n_i fuzzy sets for each domain $[x_i^-, x_i^*]$, hence we construct the respective direct and the inverse F-transforms (3) and (4) in order to estimate an approximation of f. We illustrate this forecasting method in the following steps:

1) give an uniform partition composed by n_i fuzzy sets ($n_i \geq 3$) $\{A_{i1}, \ldots, A_{in_i}\}$ of the domain $[x_i^-, x_i^*]$ of each variable x_i, i = 1,…,n. If $x_{i1}, \ldots, x_{is}, \ldots, x_{in_i}$ are the nodes of each interval $[x_i^-, x_i^*]$, each function A_{is} is defined in the following way for s=1,…,n_i:

$$\begin{aligned}
A_{i1}(x) &= \begin{cases} 0.5 \cdot (1 + \cos\frac{\pi}{s_i}(x - x_{i1})) & \text{if } x \in [x_{i1}, x_{i2}] \\ 0 & \text{otherwise} \end{cases} \\
A_{is}(x) &= \begin{cases} 0.5 \cdot (1 + \cos\frac{\pi}{s_i}(x - x_{is})) & \text{if } x \in [x_{i(s-1)}, x_{i(s+1)}] \\ 0 & \text{otherwise} \end{cases} \\
A_{in_i}(x) &= \begin{cases} 0.5 \cdot (1 + \cos\frac{\pi}{s_i}(x - x_{i(n_i-1)})) & \text{if } x \in [x_{i(n_i-1)}, x_{in_i}] \\ 0 & \text{otherwise} \end{cases}
\end{aligned} \tag{8}$$

where $x_{i1} = x_i^-$, $x_{in_i} = x_i^+$, $s_i = (x_i^+ - x_i^-)/(n_i - 1)$ and $x_{is} = x_i^- + s_i \cdot (s-1)$;

2) if the dataset is not sufficiently dense with respect to the fuzzy partition, that is if there exists a variable x_i and a fuzzy set A_{is} of the corresponding fuzzy partition such that $A_{is}(x_i^{(r)}) = 0$ for each r = 1,…,M, the process stops otherwise calculate the $n_1 \cdot n_2 \cdot \ldots \cdot n_k$ components of the direct F-transform of f with (3) by setting k=n, $p_{j1} = x_1^{(j)}, \ldots, p_{jn} = x_n^{(j)}$ and $y(j) = f(x_1^{(j)}, x_2^{(j)}, \ldots, x_n^{(j)})$, obtaining the following:

$$F_{h_1 h_2 \ldots h_n} = \frac{\sum_{j=1}^{M} y^{(j)} \cdot A_{1h_1}(x_1^{(j)}) \cdot \ldots \cdot A_{nh_n}(x_n^{(j)})}{\sum_{j=1}^{M} A_{1h_1}(x_1^{(j)}) \cdot \ldots \cdot A_{nh_n}(x_n^{(j)})} \tag{9}$$

3) calculate the discrete inverse F-transform as

$$f^F_{n_1 \ldots n_n}(x_1^{(j)}, \ldots, x_n^{(j)}) = \sum_{h_1=1}^{n_1} \sum_{h_2=1}^{n_2} \ldots \sum_{h_n=1}^{n_n} F_{h_1 h_2 \ldots h_n} \cdot A_{1h_1}(x_1^{(j)}) \cdot \ldots \cdot A_{nh_n}(x_n^{(j)}) \tag{10}$$

to approximate the function f;

4) calculate the forecasting index as

$$MADMEAN = \frac{\sum_{j=1}^{M} \left| f^{F}_{n_1 n_2 \ldots n_n}(x_1^{(j)},\ldots,x_n^{(j)}) - y^{(j)} \right|}{\sum_{j=1}^{M} y^{(j)}} \tag{11}$$

If it is less or equal to an assigned threshold, then the process stops otherwise a finer fuzzy partition is taken and the process restarts from the step 2.

In [6] four forecasting indices RMSE, MAPE, MAD, MADMEAN were proposed, but we prefer to use MADMEAN which is the best one in terms of accuracy as proved in [17].

4. TSSF method

We consider time series of data formed from observations of a original parameter y_0 measured at different times. The dataset is formed by M measure pairs as $\left(t^{(0)}, y_0^{(0)}\right), \left(t^{(1)}, y_0^{(1)}\right), \ldots, \left(t^{(M)}, y_0^{(M)}\right)$.

Our aim is to evaluate seasonal fluctuations of a time series by using the F-transform method. First of all, we use a polynomial fitting for calculating the trend of the phenomenon with respect to the time. Afterwards we subtract the trend from the data obtaining a new dataset of the fluctuations $y^{(j)}$ being $y^{(j)} = y_0^{(j)} - trend(t^{(j)})$. After de-trending the dataset, it is partitioned into S subsets, being S seen as seasonal period. Each subset represents the seasonal fluctuations with respect to the trend.

The seasonal data subset is composed by M_s pairs, expressing the fluctuation measures of the parameter y_0 at different times: $(t^{(1)}, y^{(1)})$, $(t^{(2)}, y^{(2)})$.... $(t^{(M_s)}, y^{(M_s)})$, where $y^{(i)}$ is given by the original measure $y_0^{(j)}$ at the time $t^{(j)}$ minus the trend calculated at this time. The formulae of the corresponding one-dimensional directed and inverse F-transforms, considering n basic functions, are given as

$$F_h = \frac{\sum_{j=1}^{M_S} y^{(j)} \cdot A_h(t^{(j)})}{\sum_{j=1}^{M_S} A_h(t^{(j)})} \tag{12}$$

$$f_n^{F}(t) = \sum_{h=1}^{n} F_h \cdot A_h(t) \tag{13}$$

respectively. The inverse F-transform (13) approximates the seasonal fluctuation at the time t. In our method we start with three basic functions and verify that the subset of data is sufficiently dense with respect to this fuzzy partition. After calculating the directed and inverse F-transform using (12) and (13), respectively, we calculate the $(MADMEAN)_s$ index for the *sth* fluctuation subset of data given as

$$(MADMEAN)_s = \frac{\sum_{j=1}^{M_s} \left| f_n^{F}(t^{(j)}) - y^{(j)} \right|}{\sum_{j=1}^{M_s} y^{(j)}} \tag{14}$$

If the MADMEAN index (14) is greater than an assigned threshold, than the process is iterated considering a fuzzy partition of dimension n:= n+1, otherwise the iteration process stops. Our algorithm is illustrated in Fig. 3.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 9 October 2017 doi:10.20944/preprints201710.0053.v1

Peer-reviewed version available at Symmetry 2017, 9, 281; doi:10.3390/sym9110281

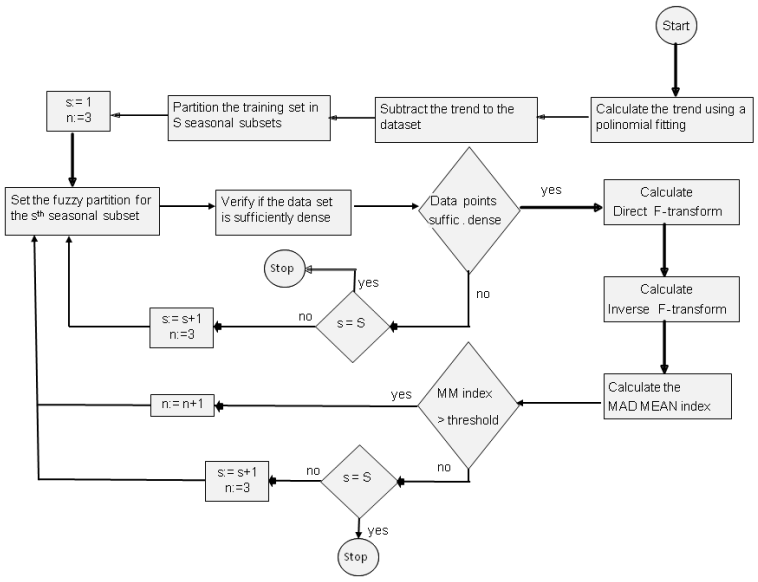


Figure 3. Flux diagram related of the TSSF method.

For the s*th* fluctuation subset, we obtain the inverse F-transform by using the following n(s) basic functions:

$$f_{n(s)}^{F}(t) = \sum_{h=1}^{n(s)} F_h \cdot A_h(t) \cdot \tag{15}$$

For evaluating the value of the parameter y_0 at the time t in the s*th* seasonal period, we add to $f_{n(s)}^{F}(t)$ the trend calculated at the time t obtaining the following value:

$$\tilde{y}_0(t) = f_{n(s)}^{F}(t) + trend(t) \tag{16}$$

For evaluating the accuracy of the results, we can use the MADMEAN index, given by:

$$MADMEAN = \frac{\sum_{j=1}^{M} \left| \tilde{y}_0(t^{(j)}) - y_0^{(j)} \right|}{\sum_{j=1}^{M} y_0^{(j)}} \tag{17}$$

For sake of completeness, we describe also the values of the above cited indices:

- the Root Mean Square Error (RMSE) defined as

$$RMSE = \left(\frac{\sum_{j=1}^{M} \left(\tilde{y}_0(t^{(j)}) - y_0^{(j)} \right)^2}{M} \right)^{1/2} \tag{18}$$

- the Mean Absolute Percent Error (MAPE) defined as

$$MAPE = \frac{1}{M} \sum_{j=1}^{M} \left| \frac{\tilde{y}_0(t^{(j)}) - y_0^{(j)}}{y_0^{(j)}} \right| \tag{19}$$

- the Mean Absolute Deviation (MAD), defined as

Peer-reviewed version available at Symmetry 2017, 9, 281; doi:10.3390/sym9110281

$$MAD = \frac{\sum_{j=1}^{M} \left| \tilde{y}_0(t^{(j)}) - y_0^{(j)} \right|}{M} \tag{20}$$

To evaluate the MADMEAN threshold we apply the k-cross-validation technique in order to control the presence of overfitting in the learning data. We set a seasonal subset and partition it randomly in k folds. The union of k-1 folds is used as a training set and the other fold is used as a validation set. Then we consider a fuzzy partition size n = 3 and apply the TSSF algorithm to the training set calculating the MADMEAN index; then we use the validation test to calculate the RMSE index. We repeat this process for the k combination of k-1 folds used as a training test and a fold used as validation set. Then, we obtain the mean MADMEAN and RMSE indexes as:

$$MADMEAN(n) = \frac{\sum_{i=1}^{k} MADMEAN_i(n)}{k} \tag{21}$$

$$RMSE(n) = \frac{\sum_{i=1}^{k} RMSE_i(n)}{k} \tag{22}$$

where MADMEAN(n) and RMSE(n) are the mean MADMEAN and RMSE by using the fuzzy partition size n.

We repeat this process by using many values of n; then we plot in a graph the RMSE(n) by varying the fuzzy partition size n. The threshold value is set as the mean MADMEAN value obtained in correspondence of the fuzzy partition size n in which there is a plateau in the RMSE graph. In our tests we set k = 10.

5. Experiments on time series data

We compare the results obtained by using the TSSF method with the one obtained by using three other seasonal forecasting methods: a simple average variation method, the seasonal ARIMA and the F-transforms prediction method. The training dataset is composed from climate data (mean, maximum and minimum temperature, pressure, speed of the wind, etc., measured every day) of the municipality of Naples collected at web pages: www.ilmeteo.it/portale/archivio-meteo/Napoli. The main climate parameters are measured every thirty minutes and are the following: Min temperature (°C), Mean temperature (°C), Max temperature (°C), Dew point (°C), Mean humidity (%), Mean view (km), Mean wind speed (km/h), Max wind speed (km/h), Gust of wind (km/h), Mean pressure on sea level (mb), Mean pressure (mb), Millimetres of rain (mm).

For sake of brevity, we limit the results for the parameters mean, max and min temperature. As training dataset, we consider these data recorded in the months July and August from 01/07/2003 to 16/08/2015, hence for overall 806 days represented as abscissas (via a number ID) in Fig.s 4 and 6, respectively. The daily mean and max temperature is represented on the ordinate axis in Fig.s 4 and 6, respectively. We obtain the best fit polynomial of nine degree ($y = \sum_{i=1}^{9} a_i \cdot x^i$) (red color in Fig.s 4 and 6) whose coefficients are given in Appendix.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 9 October 2017 doi:10.20944/preprints201710.0053.v1

Peer-reviewed version available at Symmetry 2017, 9, 281; doi:10.3390/sym9110281

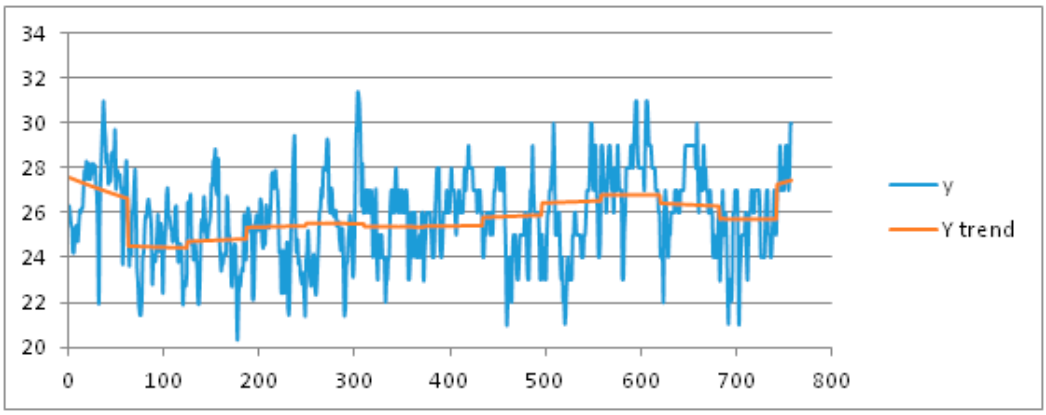


Figure 4. Trend of the mean temperature in the months of July and August (from 01/07/2003 till to 16/08/2015) obtained by using a ninth-degree polynomial fitting.

We consider the week as seasonal period, partitioning the data set into 9 seasonal subsets: week 27 (01/07) to the week 36 (31/08). We apply the TSSF method by setting the threshold of the MADMEAN index to 6%. For the daily mean and max temperature, we have also used the below other three methods, whose the final results have been plotted in Fig.s 5a÷5d and Fig.s 7a÷7d, respectively.

1) Average seasonal variation method (labelled as avgSV). This method calculates the mean seasonal variation for each seasonal period and adds the mean seasonal variation to the trend value.

2) Seasonal ARIMA method. In our experiments we used the forecasting tool ForecastPro [11]. As highlighted in [27], this tool is is well known by ensuring the best performance in the use of forecasting ARIMA models.

3) F-transform prediction method [6] applied to the complete dataset (labelled as F-transforms).

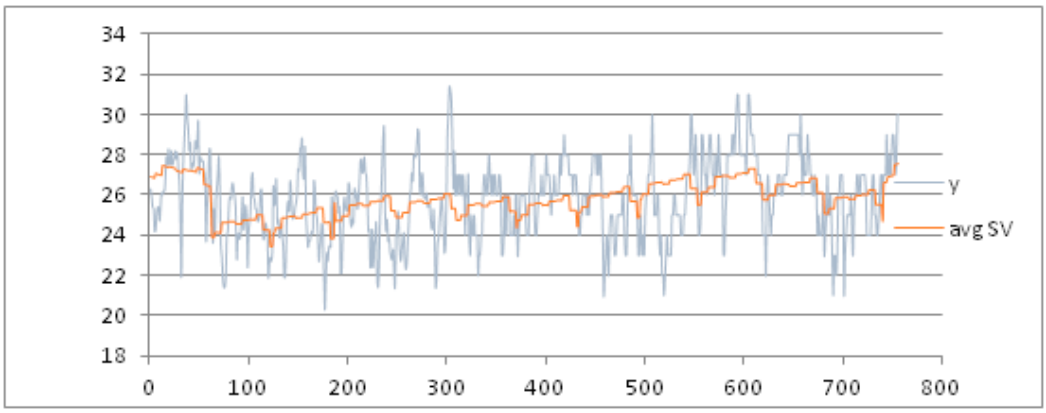
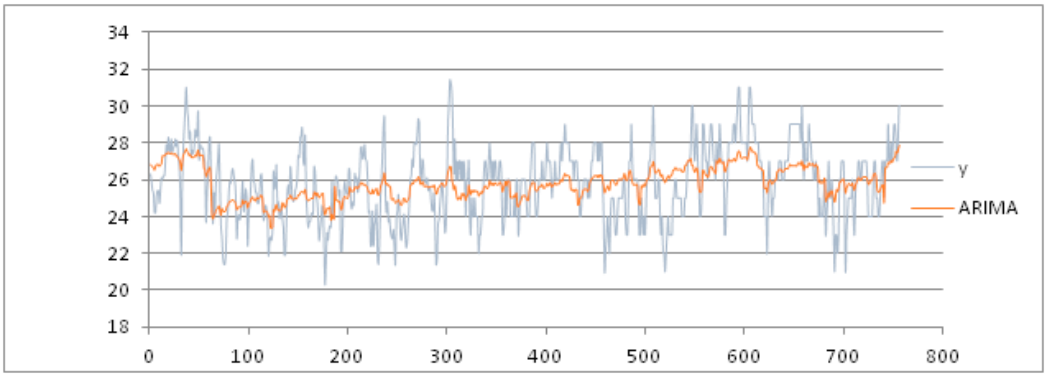


Figure 5a. Results obtained for the mean temperature by using the avgSV method.



Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 9 October 2017 doi:10.20944/preprints201710.0053.v1

Peer-reviewed version available at Symmetry 2017, 9, 281; doi:10.3390/sym9110281

Figure 5b. Results obtained for the mean temperature by using the ARIMA method.

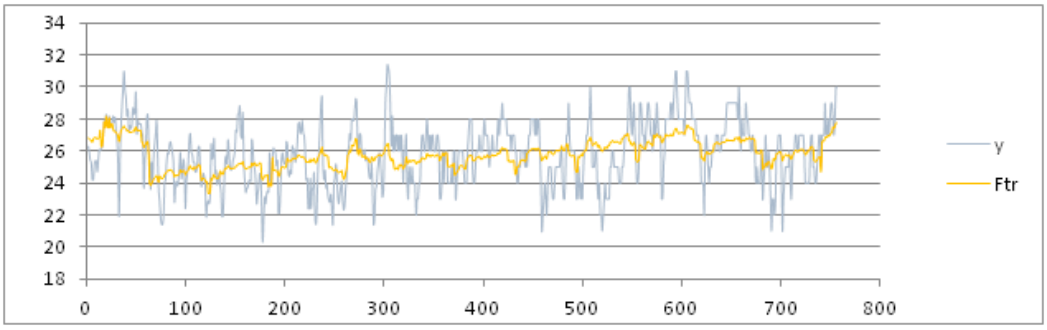


Figure 5c. Results obtained for the mean temperature by using the F-transforms method.

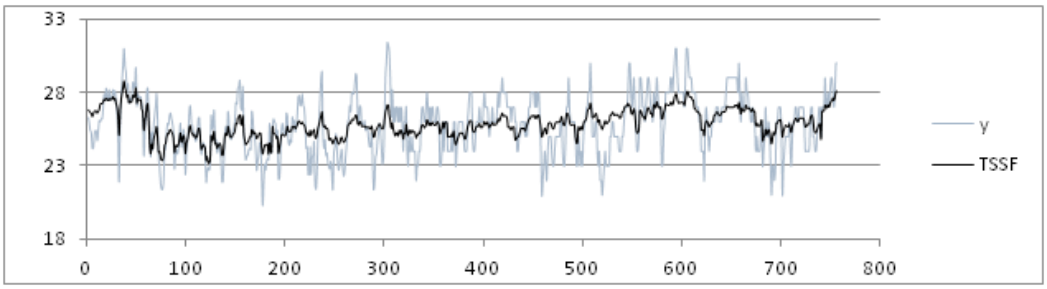


Figure 5d. Results obtained for the mean temperature by using the TSSF method.

In Table 1 we show the four indices obtained by using the four methods. The best results for the mean temperature are obtained using the TSSF method whose MADMEAN is 4.22%.

Table 1. RMSE, MAPE, MAD and MADMEAN indices for the mean temperature

Forecasting method	RMSE	MAPE	MAD	MADMEAN
avgSV	1.78	5.60%	1.42	5.50%
ARIMA	1.55	4.89%	1.24	4.80%
F-transforms	1.61	4.96%	1.28	5.05%
TSSF	1.37	4.29%	1.09	4.22%

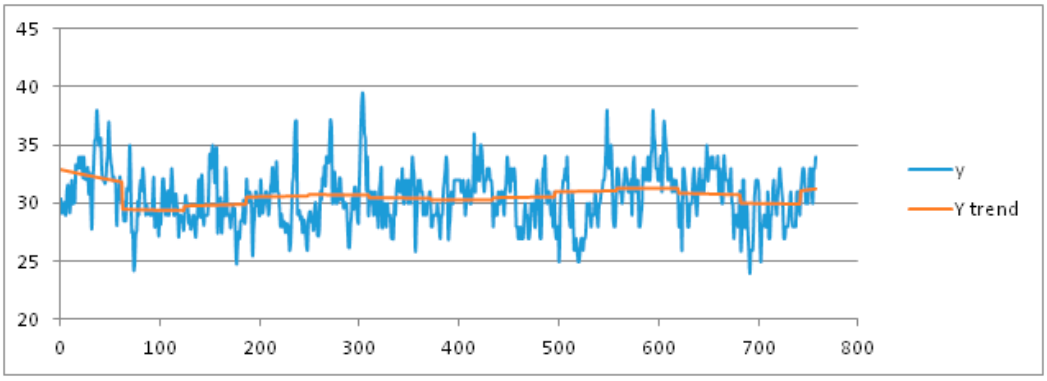
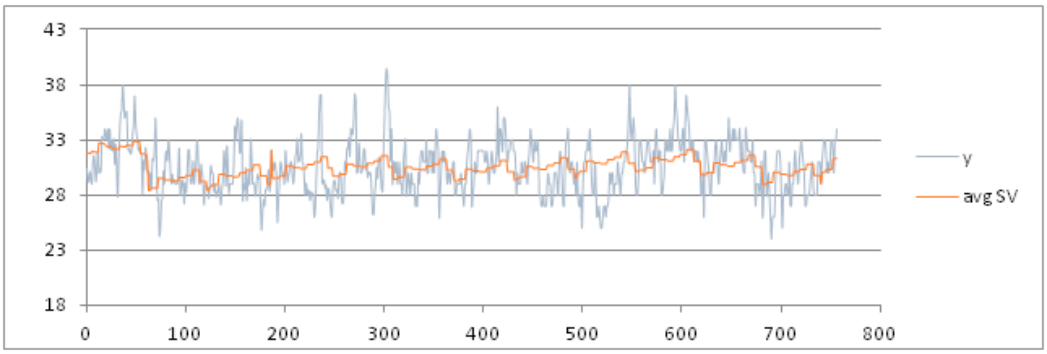


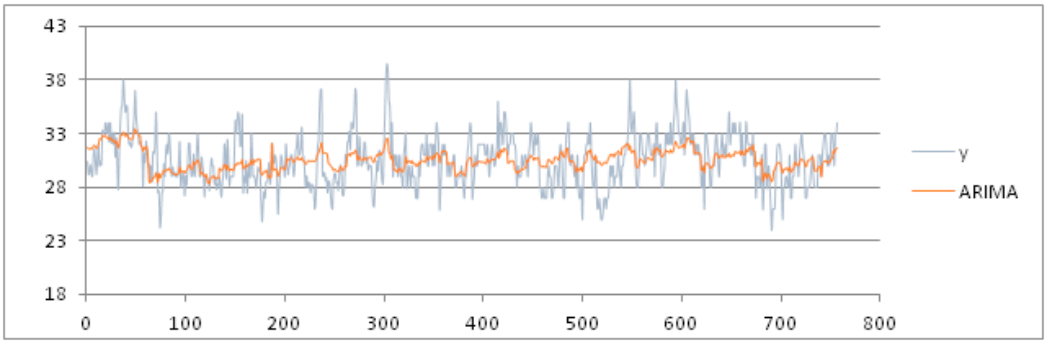
Figure 6. Trend of the max temperature in the months of July and August (from 01/07/2003 till to 16/08/2015) obtained by using a best fit polynomial of nine degree.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 9 October 2017 doi:10.20944/preprints201710.0053.v1

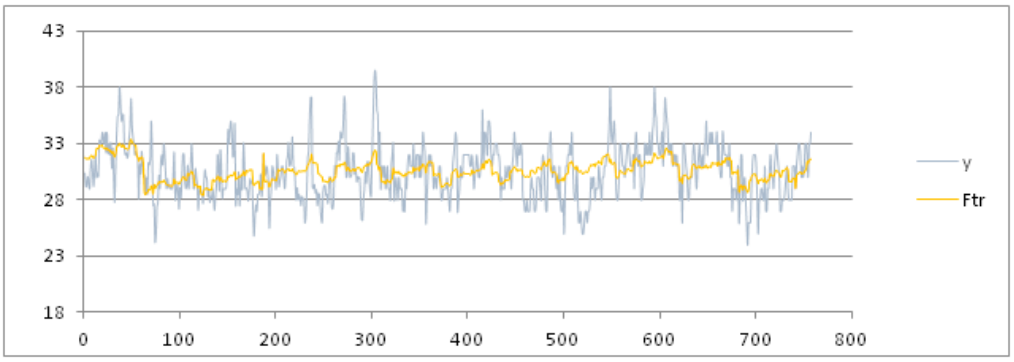
Peer-reviewed version available at Symmetry 2017, 9, 281; doi:10.3390/sym9110281



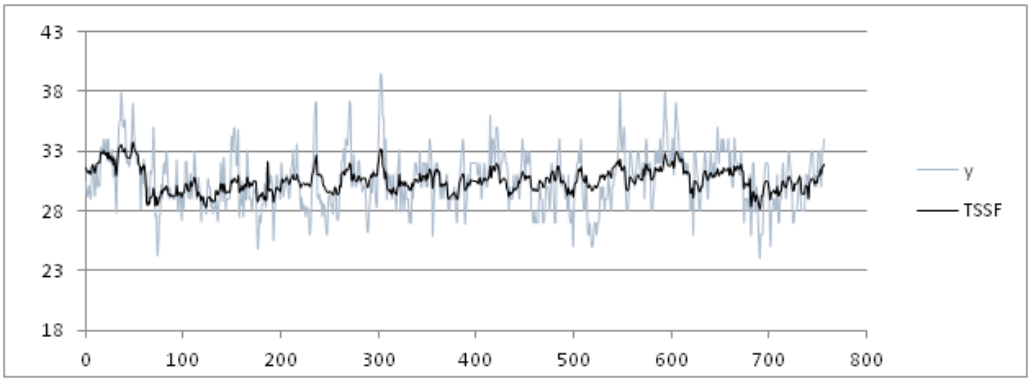
298
299 **Figure 7a.** Results for the max temperature in the months of July and August under avgSV method.



300
301 **Figure 7b.** Results for the max temperature in the months of July and August under Arima method.



302
303 **Figure 7c.** Results for the max temperature in the months of July and August under the F-transforms
304 method.



305
306 **Figure 7d.** Results for the max temperature in the months of July and August under the TSSF
307 method.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 9 October 2017 doi:10.20944/preprints201710.0053.v1

Peer-reviewed version available at Symmetry 2017, 9, 281; doi:10.3390/sym9110281

In Table 2 we show the four indexes obtained by using the four methods. The best results for the max temperature are obtained by using TSSF whose MADMEAN index is 4.56%.

Now we present the results of other experiments in which the variation of the min temperature is explored during the years. Now we assume a month as seasonal period: the training dataset is formed by all the measures recorded from 01/01/2003 to 31/12/2015. It is partitioned into 12 seasonal subsets corresponding to the 12 months of a year. In Fig. 8 we show the data and the trend obtained by using a best fit polynomial of nine degree (see Appendix).

Table 2. RMSE, MAPE, MAD and MADMEAN indices for the max temperature

Forecasting method	RMSE	MAPE	MAD	MADMEAN
avgSV	2.21	5.74%	1.74	5.70%
ARIMA	1.93	5.03%	1.53	4.99%
F-transforms	1.98	5.17%	1.56	5.13%
TSSF	1.76	4.57%	1.39	4.53%

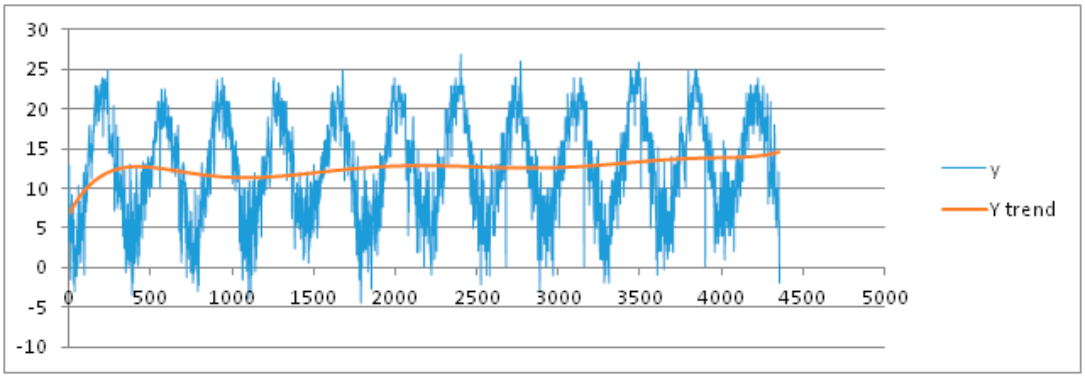


Figure 8. Trend of min temperature under a best fit polynomial of nine degree.

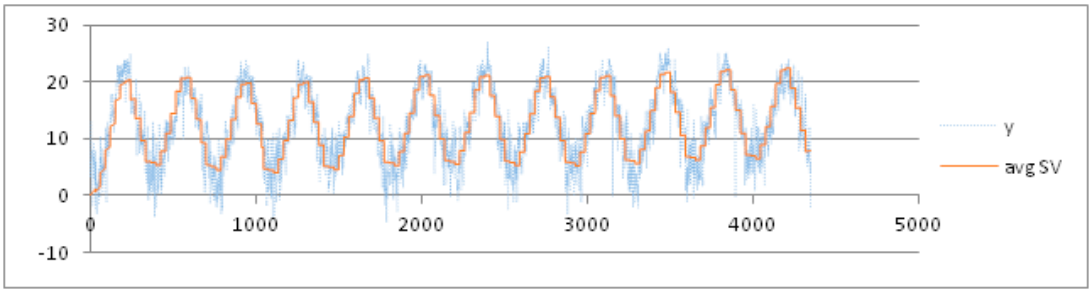
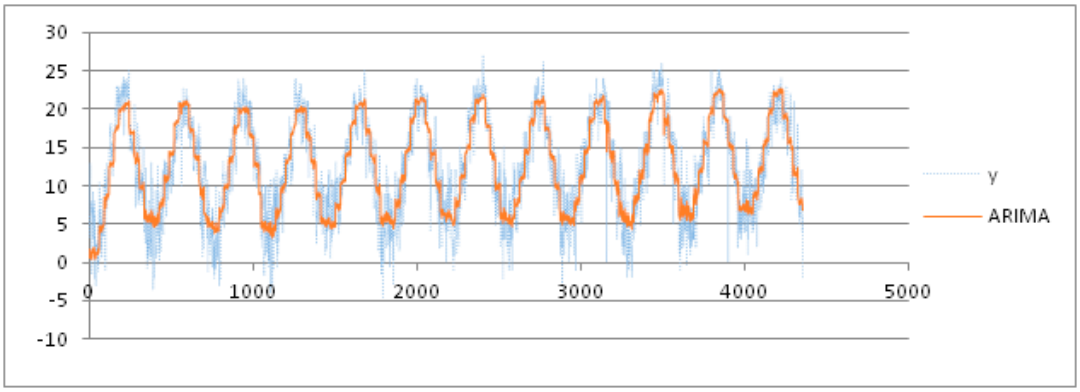


Figure 9a. Results obtained for the variation of the min temperature by using the avgSV method.



Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 9 October 2017 doi:10.20944/preprints201710.0053.v1

Peer-reviewed version available at Symmetry 2017, 9, 281; doi:10.3390/sym9110281

Figure 9b. Results obtained for the variation of the min temperature by using the ARIMA method.

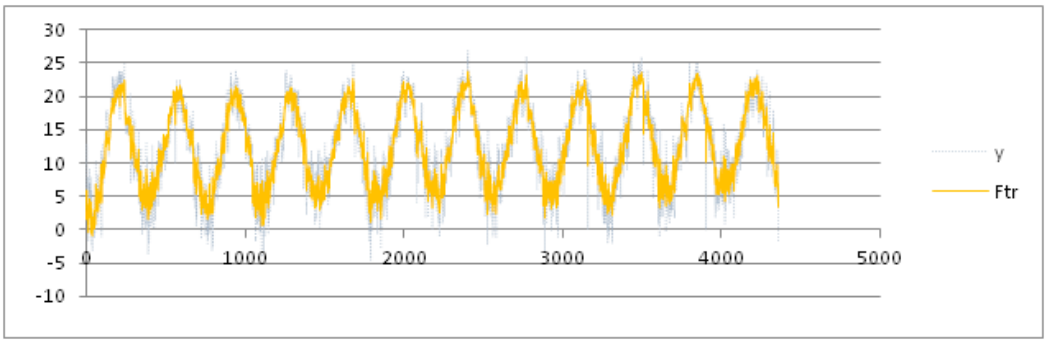


Figure 9c. Results obtained for the variation of the min temperature by using the F-transforms method.

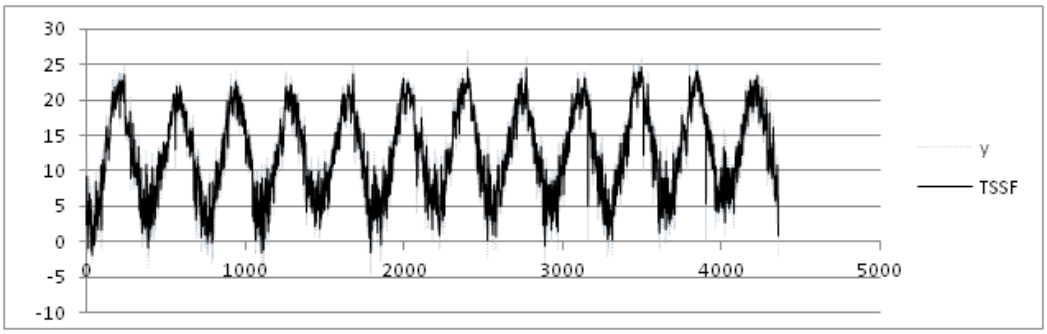


Figure 9d. Results obtained for the variation of the min temperature by using the TSSF method.

As we can observe in Fig.8, the seasonality of the data seems more evident than in the two previous examples. In Fig. 9a÷9d we have plotted the final results by using the four forecasting methods.

In Table 3 we show the indices obtained by using the four methods, the MAPE index is not measurable because there are measures of min temperature equal to 0. As for the results obtained previously, the best results for the min temperature are obtained by using the TSSF, whose MADMEAN is 5.26%.

Table 3. RMSE, MAPE, MAD and MADMEAN indices for the min temperature

Forecasting method	RMSE	MAPE	MAD	MADMEAN
avgSV	2.97	-	2.34	18.66%
ARIMA	1.25		0.97	7.46%
F-transforms	1.56	-	1.09	8.73%
TSSF	0.87	-	0.66	5.26%

The results in Table 3 show that the TSSF is more efficient when the seasonality of the data is more pronounced as for the mean temperature time series. In order to test the reliability of the forecasting results obtained by using the four forecasting methods, we have considered test datasets containing the measure of the analysed parameter in a next time period and calculating the RMSE obtained with respect to the forecasted values. For the mean and max temperature parameters we use a test dataset formed by the mean and max daily temperatures measured in the interval 17/08/2015 – 31/08/2015. For the min temperature parameter we use a test dataset formed by the min

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 9 October 2017 doi:10.20944/preprints201710.0053.v1

Peer-reviewed version available at Symmetry 2017, 9, 281; doi:10.3390/sym9110281

daily temperatures measured in the interval 01/01/2015-31/08/2015. In Table 4 we show the RMSE measured in the four methods for each of the three parameters.

Table 4. RMSE measured in the four methods

Parameter	Training dataset dimension	Test dataset dimension	Seasonal period	RMSE			
				avgSV	ARIMA	F-transf.	TSSF
Temp. mean	757	15	Week	1.83	1.58	1.62	1.39
Temp. max	757	15	Week	2.20	1.92	1.99	1.68
Temp. min	4354	212	Month	2.90	1.27	1.54	0.94

These results confirm that the best performances are obtained by using the TSSF algorithm: the most reliable results are obtained for the third parameter in which the seasonality is more regular.

We repeated these comparison tests by using other climatic datasets. For brevity we show the results obtained analysing the trend of the daily mean temperature measured at the weather station *Chiavari Caperana* in the Liguria Italian Region; the data can be downloaded at the web site http://www.cartografiarl.regione.liguria.it/SiraQualMeteo/script/PubAccessoDatiMeteo.asp of the Liguria Region web page "Ambiente in Liguria".

The analyzed data concern the measures in °C of the daily mean temperature in the period 01/01/2006 - 31/12/2016. The month is set as the seasonal period; the dataset it is partitioned into 12 seasonal subsets corresponding to the 12 months of a year.

In Fig. 10 we show the data and the trend obtained by using a best fit polynomial of nine degree (see Appendix).

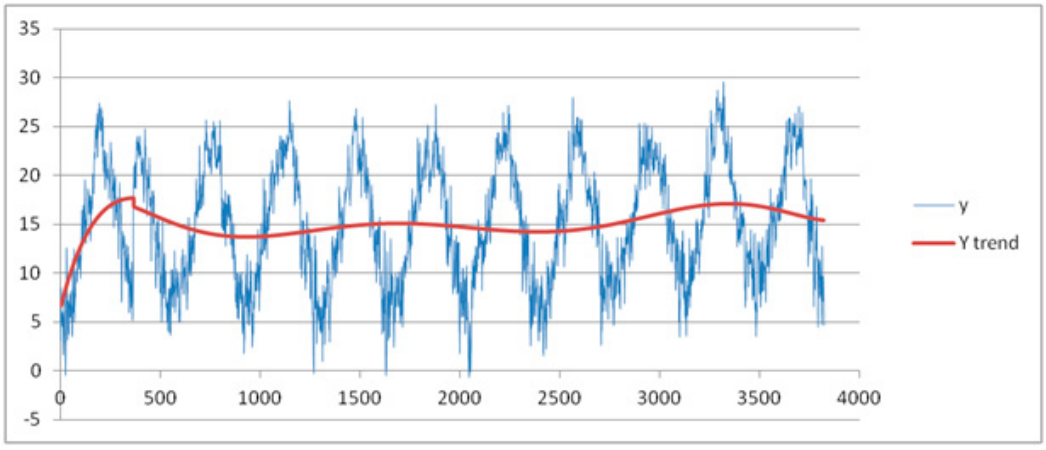


Figure 10. Trend of mean temperature under a best fit polynomial of nine degree - Chiavari Caperana weather station dataset.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 9 October 2017 doi:10.20944/preprints201710.0053.v1

Peer-reviewed version available at Symmetry 2017, 9, 281; doi:10.3390/sym9110281

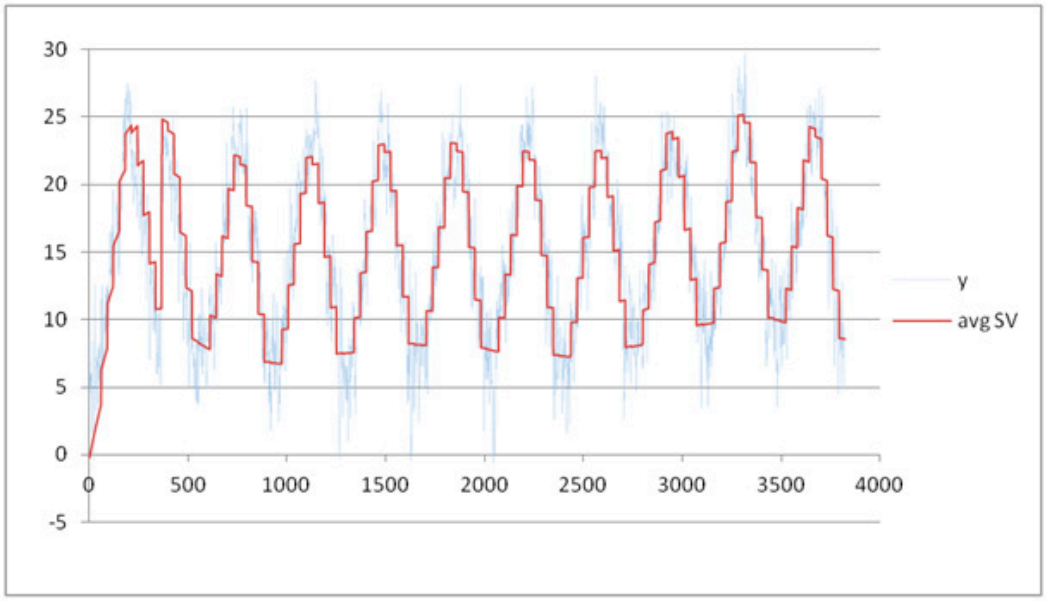


Figure 11a. Results obtained for the variation of the min temperature by using the avgSV method.

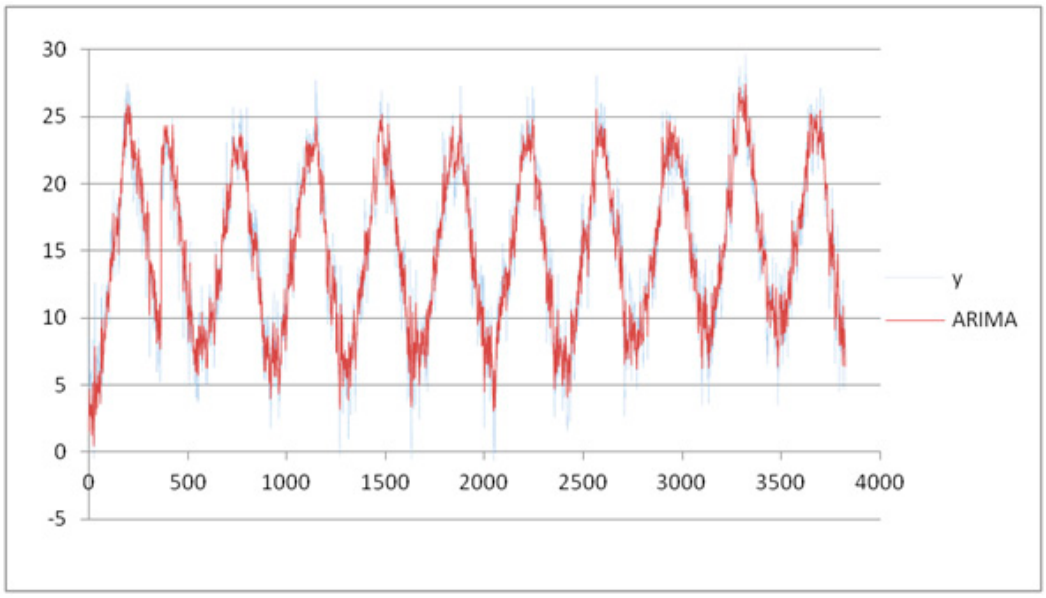


Figure 11b. Results obtained for the variation of the min temperature by using the ARIMA method.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 9 October 2017 doi:10.20944/preprints201710.0053.v1

Peer-reviewed version available at Symmetry 2017, 9, 281; doi:10.3390/sym9110281

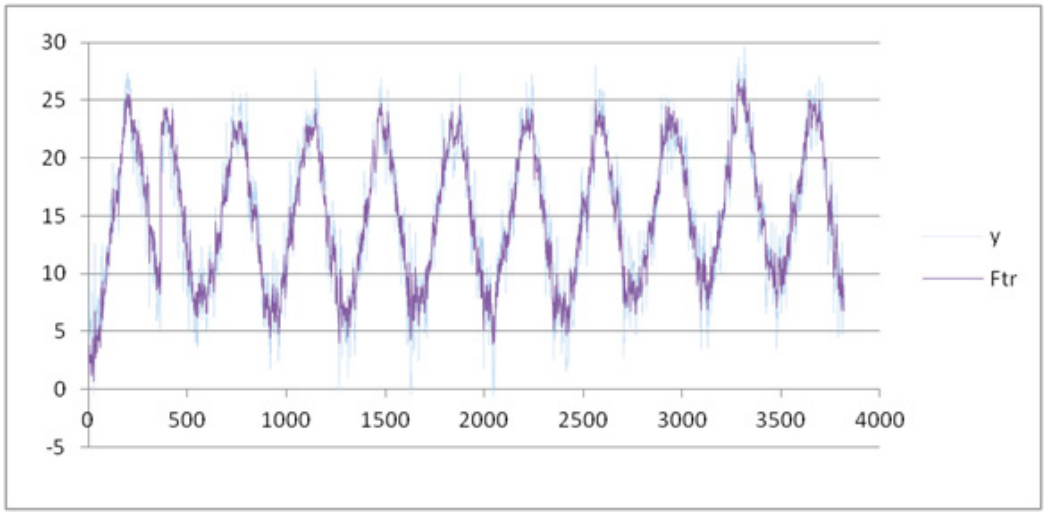


Figure 11c. Results obtained for the variation of the min temperature by using the F-transforms method.

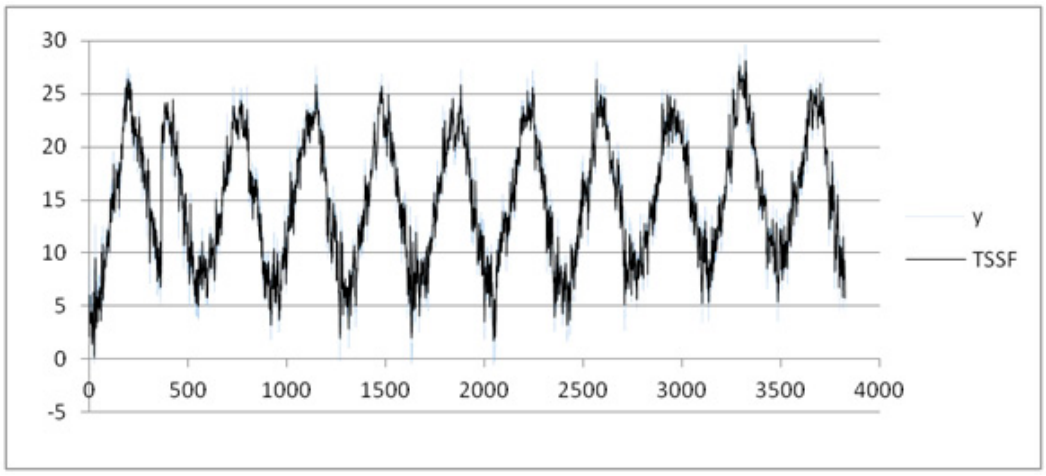


Figure 11d. Results obtained for the variation of the min temperature by using the TSSF method.

In Table 5 we show the four indexes obtained by using the four methods. The best results for the mean temperature are obtained by using TSSF whose MADMEAN index is 3.83%.

Table 5. RMSE, MAPE, MAD and MADMEAN indices for the mean temperature (Chiavari Caperana weather station)

Forecasting method	RMSE	MAPE	MAD	MADMEAN
avgSV	2.87	27.74%	2,14	16.19%
ARIMA	1.13	12.37%	0.89	5.95%
F-transforms	1.38	15.18%	1.10	7.30%
TSSF	0.73	7.96%	0.58	3.83%

The results in Table 5 confirm the ones obtained for the min temperature parameters in the weather dataset of the Municipality of Naples in Table 3: the TSSF gives better results than the ARIMA and F-transform algorithms and is more efficient when the seasonality of the data is more pronounced. In order to test the reliability of the results obtained in Table 5, we have considered a test dataset containing the measure of the daily mean temperature in the period

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 9 October 2017 doi:10.20944/preprints201710.0053.v1

Peer-reviewed version available at Symmetry 2017, 9, 281; doi:10.3390/sym9110281

01/01/2017-19/03/2017 and calculating the RMSE obtained with respect to the forecasted values. In Table 6 we show the RMSE measured in the four methods for each of the three parameters.

Table 6. RMSE measured in the four methods for the mean temperature (Chiavari Caperana weather station)

Parameter	Training dataset dimension	Test dataset dimension	Seasonal period	RMSE			
				avgSV	ARIMA	F-transf.	TSSF
Temp. mean	3822	78	Month	2.85	1.15	1.40	0.81
Temp. max	757	15	Week	2.20	1.92	1.99	1.68
Temp. min	4354	212	Month	2.90	1.27	1.54	0.94

Also the results in Table 6 confirm that the best performances are obtained by using the TSSF algorithm. In the other tests we obtain the same results; the results show that the better performances are obtained by using the TSSF algorithm and the most reliable results are obtained when the seasonality is more regular.

In Table 7 we show the tests results obtained for the mean temperature by using the seasonal datasets of the daily mean temperature measured by other stations in the district of Genova; the data were downloaded from the Liguria Region web page "Ambiente in Liguria". In each experiment the season parameter is given by the month of the year.

Table 7. RMSE measured in the four methods for the mean temperature in different station in the district of Genova (Italy):

Station	RMSE			
	avgSV	ARIMA	F-transf.	TSSF
Alpe Gorreto	2.98	1.20	1.49	0.84
Campo Ligure	2.74	1.09	1.34	0.76
Barbagelata	3.25	1.30	1.57	0.89
Camogli	3.39	1.38	1.68	0.95
Campo ligure	3.02	1.20	1.49	0.83
Carlasco	2.91	1.15	1.42	0.80
Chiavari	2.78	1.12	1.39	0.78
Genova Bolzaneto	2.95	1.16	1.41	0.81
Genova Pegli	3.34	1.29	1.64	0.89
Panesi	3.20	1.29	1.56	0.87
Rapallo	2.71	1.08	1.33	0.75
Rovegno	2.94	1.18	1.45	0.82
Tigliolo	3.06	1.24	1.52	0.85
Viganego	3.17	1.28	1.57	0.88

The results in Table 7 confirm that the better performances are obtained by using the TSSF algorithm with respect to the other three algorithms.

5. Conclusions

We present a new method based on F-transforms for seasonal forecasting. The goal was to adapt the F-transform based forecasting method in [6] on the analysis of seasonal times series, improving its performances with respect to other known methods. Usually ARIMA model has better performance than models based on ANN and SVM, but these models are complex to manage: for

Peer-reviewed version available at Symmetry 2017, 9, 281; doi:10.3390/sym9110281

instance, the choice of the input parameters often affects the reliability of the final results. Our results show that the TFSS method improves the performances of ARIMA, avgSV and F-transform forecasting methods. In future works we shall try to improve TSSF into the management of time-series that also contain irregular variations, comparing it with other soft computing models.

Appendix

Coefficients of the polynomial of 9th degree used for finding the best fit to training data of the mean temperature in the period 01/07/2003÷16/08/2015.

a9 = 1,34428E-33
a8 = -2,82732E-28
a7 = 2,32945E-23
a6 = -9,06821E-19
a5 = 1,45202E-14
a4 = 0
a3 = 0
a2 = -0,066427015
a1 = 0
a0 = 17699903,37

Coefficients of the polynomial of 9th degree used for finding the best fit to training data of the max temperature in the period 01/07/2003÷16/08/2015.

a9 = 1.45368E-33
a8 = -3.06035E-28
a7 = 2.52386E-23
a6 = -9.83437E-19
a5 = 1.57619E-14
a4 = 0
a3 = 0
a2 = -0.072310129
a1 = 0
a0 = 19302936.01

Coefficients of the polynomial of 9th degree used for finding the best fit to training data of the min temperature in the period 01/01/2003÷31/12/2015.

a9 = 4.40678E-32
a8 = -1.11212E-26
a7 = 1.14529E-21
a6 = -5.94437E-17
a5 = 1.42772E-12
a4 = 0
a3 = -0.000762063
a2 = 13.0645932
a1 = 0
a0 = -1160591164

Coefficients of the polynomial of 9th degree used for finding the best fit to training data of the mean temperature measured at the weather station Chiavari Caperana in the period 01/01/2006÷31/12/2016.

a9 = 1.53589E-31
a8 = -4.0454E-26
a7 = 4.41907E-21
a6 = -2.5338E-16

a5 = 7.78147E-12
a4 = -1.06189E-07
a3 = 0
a2 = 0
a1 = 519825.0932
a0 = -7088111243

References

1. Abraham, B.; Ledolter, J. Statistical Methods for Forecasting, John Wiley & Sons, New York, 1983, 445 pages, ISBN: 978-0-471-86764-3.
2. Armstrong, J. S. (Ed.), Principles of Forecasting: a Handbook for Researchers and Practitioners, University of Pennsylvania, Wharton School, Philadelphia, 2001, Eds. Springer, 841 pp, DOI: 10.1007/9780306476303.
3. Box, G. E. P. ; Jenkins G. M.; Reinsel G. C. Time Series Analysis: Forecasting and Control, Prentice Hall, Englewood Cliffs, 5th ed., 2015, 712 pages, ISBN: 978-1-118-67502-1.
4. Chatfield C. Time Series Forecasting, Chapman & Hall/CRC, Boca Raton ,USA,20001, 267 pages, ISBN:1-58488-063-5.
5. Crone, S. F.; Hibon, M.; Nikolopoulos, K. Advances in forecasting with neural networks? Empirical evidence from the NN3 competition on time series prediction. *International Journal of Forecasting*, **2011**, 27 (3), 635–660, DOI: : 10.1016/j.ijforecast.2011.04.001.
6. Di Martino, F.; Loia, V.; Sessa, S. Fuzzy transforms method in prediction data analysis, *Fuzzy Sets and Systems*, **2011**, 180, 146-163, DOI: 10.1016/j.fss.2010.11.009.
7. Di Martino, F.; Loia, V.; Sessa, S. Fuzzy transforms method and attribute dependency in data analysis, *Inform. Sciences*, **2010**, 180, 493-505, DOI: 10.1016/j.ins.2009.10.012.
8. Di Martino, F.; Loia, V.; Perfilieva, I.; Sessa, S. An image coding/decoding method based on direct and inverse fuzzy transforms, *Internat. Journal of Approximated Reasoning*, **2008**, 48, 110–131, DOI: 10.1016/j.ijar.2007.06.008 .
9. Faraway, J. ; Chatfield, C. Time series forecasting with neural networks: a comparative study using the airline data, *J. of Roial Statistical Society. Series C: Applied Statistics,* **1998**, 47 (2), 231-250, DOI: 10.1111/1467-9876.00109.
10. Fard, A. K.; Samet, H.; Marzbani, F. A new hybrid modified firefly algorithm and support vector regression model for accurate short term load forecasting, *Expert Systems with Applications*, **2014**, 41, 6047–6056, . DOI: 10. 1016/j.eswa.2014.03.053.
11. Goodrich, R. L. The ForecastPro methodology, *International Journal of Forecasting*, **2000**, 16 (4), 533-535, DOI: 10.1016/S0169-2070(00)00086-8 .
12. Hamzacebi, C. Improving artificial neural networks performance in seasonal time series forecasting, *Information Sciences,* **2008**, 178, 4550-4559, DOI: 10.1016/j.ins.2008.07.024.
13. Hong, W. C.; Pai, P. F. Potential assessment of the support vector regression technique in rainfall forecasting, *Water Resources Management*, **2007**, 21 (2), 495–513, DOI: 10.1007/s11269-006-9026-2.
14. Hymdam, R. J.; Athanasopoulos, G. Forecasting Principles and Practice, OText Publisher, 2013. 291 pp., ISBN: 9780987507105.
15. Khandelwal, I.; Adhikari, R.; Verma, G. Time series forecasting using hybrid ARIMA and ANN models based on DWT decomposition, **2015**, *Procedia Computer Science*, 48, 173–179, DOI: 10.1016/j.procs.2015.04.167.
16. Kihoro, J. M.; Otieno, R.O.; Wafula, C. Seasonal time series forecasting: a comparative study of ARIMA and ANN models, *African Journal of Science and Technology*, **2004**, 5(2), 41-49, DOI: 10.4314/ajst.v5i2.15330.
17. Kolassa, W.; Schutz, W. Advantages of the MADMEAN ratio over the MAPE, *Foresight, International Journal of Applied Forecasting*, **2007**, 6 , 40–43.
18. Ittig, P. T. A seasonal index for business, *Decision Sciences*, **1997**, 28 (2), 335–355, DOI: 10.1111/j.1540-5915.1997.tb01314.x.
19. Lu, W. Z.; Wang, W. J. Potential assessment of the Support Vector Machine method in forecasting ambient air pollutant trends, *Chemosphere*, **2005**, 59, 693–701, DOI: 10.1016/j.chemosphere.2004.10.032.

20. Makridakis, S. G. ; Wheelwright, S. C.; Hyndman, R. J. Forecasting: methods and applications, J. Wiley & Sons, New York, 3rd edition, 1998, 656 pp., ISBN: 978-0-471-53233-0.
21. Miller, K.; Smola, A. J.; Ratsch, G.; Scholkopf, B.; Kohlmorgen, J.; Vapnik, V. Predicting time series with support vector machines, Proceedings of the 7th International Conference on Artificial Neural Networks, Lecture Notes in Computer Sciences, Springer 1998, vol. 1327, pp. 999–1004, DOI: 10.1007/BFb0020283.
22. Mohandes, M. A.; Halawani, T. O.; Rehman, S.; Hussain, A. A. Support vector machines for wind speed prediction, *Renewable Energy*, **2004**, 29(6), 939–947 DOI: 10.1016/j.renene.2003.11.009.
23. [23] Novák, V.; Pavliska, V.; Perfilieva, I., Štepnicka M., F-transform and fuzzy natural logic in Time Series Analysis, 8th Conference of the European Society for Fuzzy Logic and Technology (EUSFLAT 2013), Atlantic Press, 2013, pp. 40-47.
24. Pai, P. F.; Lin, K.P.; Lin,C. S.; Chang, P. T. Time series forecasting by a seasonal support vector regression model, *Expert Systems with Applications*, **2010**, 37, 4261–4265, DOI: 10.1016/j.eswa.2009.11.076.
25. [25] Pankratz, A. Forecasting with Dynamic Regression Models, Wiley ed., 2012, 392 pp., DOI: 10.1002/9781118150528.
26. [26] Perfilieva, I. Fuzzy transforms: theory and applications, *Fuzzy Sets and Systems*, **2006**, 157(8), 993–1023, DOI: 10.1016/j.fss.2005.11.012.
27. 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, *Expert Systems with Applications*, **2013**, 40, 1981–1992, DOI: 10.1016/j.eswa.2012.10.001.
28. Wang, L. X. Mendel, J. M. Generating fuzzy rules by learning from examples, *IEEE Transactions on Systems, Man, and Cybernetics*, **1992**, 22 (6),1414-1427, DOI: 10.1109/21.199466.
29. Zhang, G.; Patuwo, B. E.; Hu, M.Y. Forecasting with artificial neural networks: the state of the art, *International Journal of Forecasting*, **1998**, 14(1), 35-62, DOI: 10.1016/S0169-2070(97)00044-7.
30. Zhang, G.; Zhang, G. P. Time series forecasting using a hybrid ARIMA and neural network model, *Neurocomputing*, **2003**, 50, 159–175, DOI: 10.1016/S0925-2312(01)00702-0.
31. Zhang, G. P.; Kline, D. M. Quarterly time-series forecasting with neural networks, *IEEE Transactions on Neural Networks*, **2007**, 18(6), 1800–1814, DOI: 10.1109/TNN.2007.896859.
32. Zhang, G. P.; Qi, M. Neural network forecasting for seasonal and trend time series, *European Journal of Operational Research*, **2005**, 160 (2), 501–514, DOI: 10.1016/j.ejor.2003.08.037.