Annales Geophysicae, 23, 3035–3042, 2005
SRef-ID: 1432-0576/ag/2005-23-3035

Annales Geophysicae

One-step ahead prediction of foF2 using time series forecasting techniques

K. Koutroumbas and A. Belehaki

National Observatory of Athens, Institute for Space Applications and Remote Sensing, Metaxa and V. Pavlou, Palaia Penteli, 15 236, Athens, Greece

Received: 24 February 2005 – Revised: 9 September 2005 – Accepted: 4 October 2005 – Published: 22 November 2005

Part of Special Issue "1st European Space Weather Week (ESWW)"

Abstract. In this paper the problem of one-step ahead prediction of the critical frequency (foF2) of the middle-latitude ionosphere, using time series forecasting methods, is considered. The whole study is based on a sample of about 58 000 observations of foF2 with 15-min time resolution, derived from the Athens digisonde ionograms taken from the Digisonde Portable Sounder (DPS4) located at Palaia Penteli (38° N, 23.5° E), for the period from October 2002 to May 2004. First, the embedding dimension of the dynamical system that generates the above sample is estimated using the false nearest neighbor method. This information is then utilized for the training of the predictors employed in this study, which are the linear predictor, the neural network predictor, the persistence predictor and the k-nearest neighbor predictor. The results obtained by the above predictors suggest that, as far as the mean square error is considered as performance criterion, the first two predictors are significantly better than the latter two predictors. In addition, the results obtained by the linear and the neural network predictors are not significantly different from each other. This may be taken as an indication that a linear model suffices for one step ahead prediction of foF2.

Keywords. Ionosphere (Modelling and forecasting) – History of geophysics (Instruments and techniques)

1 Introduction

The accurate prediction of ionospheric conditions is critical for several applications affected by the space weather, including HF communications, satellite positioning and navigation applications. Ionospheric storms can cause large-scale, drastic changes to the usable range of HF frequencies. Large solar flares cause short-wave fadeouts, resulting in blackouts of HF signals. Also, protons emitted from the Sun result in polar cap absorption events and consequently in blackouts of HF signals propagating through the Earth's polar regions. Ionospheric effects can also generate time-varying ionospheric currents, especially in the northern latitudes causing problems in ground systems, such as systems for power generation and supply; oil and gas pipeline distribution; aerial surveying for minerals, oil and gas; drilling for oil and gas; railways.

To handle the complexity of the problem, the development of accurate models to forecast the ionospheric conditions is of crucial importance, especially during disturbed conditions. Considerable effort has been devoted on the development of physical models (Anderson et al., 1998) based on the coupling between the thermosphere and the ionosphere. However, the use of such models is not suitable for real-time applications due to both the input data requirements for the simulation of the thermosphere (Fuller-Rowell and Rees, 1980) and the computational load.

The empirical models driven by magnetic indices are a second group of ionospheric models. Fuller-Rowell et al. (2001) developed the empirical storm-time ionospheric correction model driven by the previous time-history of the geomagnetic index, ap, and it is designed to scale the quiet-time F-layer critical frequency (foF2) to account for storm-time changes in the ionosphere. The model provides a useful, yet simple tool for estimating the changes to ionosphere in response to geomagnetic activity.

Another well-known statistical model for the prediction of ionospheric parameters was introduced by Muhtarov and Kutiev (1999), which makes use of the auto correlation function of the parameter under consideration without using any geomagnetic index. Muhtarov et al. (2002) further improved the prediction capability of the autocorrelation model by adding a geomagnetic index and its statistical characteristics.

An alternative approach for predicting ionospheric conditions is based on time series forecasting techniques. Data-driven modelling techniques, such as neural networks, are used to predict the behaviour of the ionosphere under the assumption that non linear processes are the dominant mechanisms that generate foF2 variability (Tulunay et al., 2004a,

Correspondence to: K. Koutroumbas
(koutroum@space.noa.gr)

b; Wintoft and Cander, 2000; McKinnell and Poole, 2000).

Wintoft and Cander (2000) used time-delay, feed-forward neural networks to predict the hourly values of the ionospheric $F2$ layer critical frequency, foF2, 24 h ahead. The 24 measurements of foF2 per day are reduced to five coefficients with principal component analysis. A time delay line of these coefficients is then used as input to a feed-forward neural network. Also included in the input are the 10.7-cm solar flux and the geomagnetic index A_p. The network is trained using foF2 data from 1965 to 1985 gathered at the Slough ionospheric station and validated on an independent validation set from the same station for the periods 1987–1990 and 1992–1994.

In a recent study, Tulunay et al. (2004) presented the application of the Middle East Technical University Neural Network (METUNN) to forecast the foF2 values one hour in advance, based on hourly resolution data. The input parameters are year, month, coded season, day, hour, coded hour, foF2 value observed one hour ago, first and second relative difference, station code. The method was applied to data from Poitier, Slouth and Uppsala, and the mean square errors were within reasonable limits (0.11353 to 0.21145 MHz).

The problem considered in this study is the estimation of the next value of foF2 using time series forecasting methods. The available data sample $X=\{x_1, x_2, \ldots, x_p\}$ consists of about p=58 000 observations of foF2 derived from the Athens digisonde ionograms taken from the ionospheric station located at Palaia Penteli, for the period from October 2002 to May 2004. The sampling rate is 15 min. There is a small fraction of missing observations that have been neglected from the subsequent prediction stages.

If we denote the current value of foF2 by $x(n)$, then the estimation of the next value $x(n+1)$ is based on

$$\mathbf{y}(n) = [x(n),\ x(n-1), \ldots, x(n-(d-1))]^T. \quad (1)$$

The first problem to be faced is the estimation of d, the so-called *embedding dimension*. This is estimated using the *false nearest neighbor* method. After the determination of d, two sets of pairs of the form $(\mathbf{y}(n),\ x(n+1))$ are created. The first one, denoted by S_1 and called the *training set*, will be used for the training of the predictors, while the second one, denoted by S_2 and called the *test set*, will be used for the evaluation of the performance of the predictors. The evaluation criterion for the above predictors is the *mean square error* (MSE), that is the mean value of the squared difference between the actual and the predicted values of foF2.

In this study, both parametric and non-parametric predictors are used. Specifically, from the first category the linear predictor, as well as neural network predictors, are considered, while from the second category the persistence predictor, as well as the k nearest neighbor predictor, are considered. The experimental results show that all predictors exhibit a less than 13% error on the test set, in terms of the MSE criterion. This issue will be discussed further in the simulation results section.

The rest of the paper is organized as follows. In Sect. 2, the definition of the embedding dimension, d, is given, together with a short description of the false nearest neighbor method that estimates d. In Sect. 3 a short description of the predictors considered in this study is given. In Sect. 4 the procedure that generates the training and the test sets, S_1 and S_2, is described. In addition, the results of the predictors followed by a short discussion are provided. Finally, concluding remarks, as well as future research directions, are included in Sect. 5.

2 The embedding dimension

Let $\mathcal{A}$ denote the d-dimensional dynamical system that produces the available time series of observations and let $\boldsymbol{s}(n)$ denotes its state vector at time n. Assuming that $\mathcal{A}$ is a discrete dynamical system, it is described by the following equation (also called map)

$$\boldsymbol{s}(n+1) = h(\mathbf{s}(n)). \quad (2)$$

Clearly, this system is unknown, that is we do not know the dimension d, nor the function h. The only available information about it is through the sequence of observations $\{x(n)\}$, which are related with the state vector $\boldsymbol{s}(n)$ via the following equation:

$$x(n) = g(\mathbf{s}(n)). \quad (3)$$

Since, in general, the available sequence of observations[1] does not represent properly the multi-dimensional phase space of the dynamical system, one has to employ some technique to unfold the multi-dimensional structure using the available data series (Hegger et al., 1999).

The most important technique for the phase space reconstruction is the method of delays (see, e.g. Tsonis, 1992; Hegger et al., 1999). According to this method, the vectors in the new space (the embedding space) are formed from time delayed values of the scalar measurements, i.e.[2]

$$\boldsymbol{y}(n)=[x(n), x(n-1), \ldots, x(n-(d-1))]^T, \quad (4)$$

and d is the dimension of the embedding space, called the embedding dimension. Knowledge of d is of crucial importance, but, of course, it is unavailable in real world situations and has to be estimated from the available data series. Specifically, d should be chosen large enough to allow for the unfolding of the multi-dimensional structure of the system, but not too large, in order to avoid the undesirable effects of the

[1] which is one-dimensional in most cases

[2] In general, $\mathbf{y}(n)$ is defined as $\mathbf{y}(n)=[x(n),\ x(n-T), \ldots, x(n-(d-1)T)]^T$, where T is the so-called *time delay* parameter. However, in the present study we assume that $T=1$ since we are interested in one-step ahead predictions. However, allowing values greater than 1 for T, we may obtain interesting results. For example, choosing a value of T equal to 26, which corresponds to 6.5 h, since the sampling rate for the data at hand is 15 min (this is the first time where the autocorrelation function for the data set at hand becomes zero, see Tsonis, 1992), we obtain very good 6.5-h ahead prediction. Nevertheless, the last issue deserves more investigation.

noise encountered in the measurements, as well as the unnecessary increase in computational complexity (Kennel et al., 1992).

A method that has been extensively used for the estimation of d is the so-called method of false nearest neighbors (Kennel et al., 1992), which is described below. Let $X'=\{x(1), x(2), \ldots, x(q)\}$[3] be the set of observations on which the estimation of d will be based.

The false nearest neighbor method

- Compute the quantities

$$\overline{x}=\frac{1}{q}\sum_{n=1}^{q}x(n), \quad R_A^2=\frac{1}{q}\sum_{n=1}^{q}(x(n)-\overline{x})^2.$$

- Set R_{tot}=15 and A_{tot}=2 (as suggested in Kennel et al., 1992).

- Choose a high enough value of d, say, $d_{\max}$, and use X' to construct the set

$$Z = \{\text{y}(n)=[x(n), x(n-1), \ldots, x(n-(d_{\max}-1))]^T, \\ n = d_{\max}, \ldots, q\}.$$

- For d=1 to $d_{\max}$
  - For each $\boldsymbol{y}(n)$ in Z, determine its nearest neighbor $\boldsymbol{y'}(n)=[x'(n), \ldots, x'(n-(d_{\max}-1))]^T$ in $Z-\{\boldsymbol{y}(n)\}$, based on the last d coordinates of the y values, i.e. choose y'(n) such that $d(\boldsymbol{y}(n), \boldsymbol{y'}(n))=\min_{\boldsymbol{y}\in Z-\{\boldsymbol{y}(n)\}} d(\boldsymbol{y}(n), \boldsymbol{y})$, where the distance between two $d_{\max}$-dimensional vectors, $\boldsymbol{u}$ and $\boldsymbol{v}$, is defined as $d(\boldsymbol{u}, \boldsymbol{v})=\sum_{i=0}^{d}(u_i-v_i)^2$, where u_i and v_i are the i-th coordinates of $\boldsymbol{u}$ and $\boldsymbol{v}$, respectively.
  - For each $\boldsymbol{y}(n)$ in Z compute

$$R_d^2(n) = \sum_{k=0}^{d-1}(x(n-(d_{\max}-1)+k) \\ -x'(n-(d_{\max}-1)+k))^2$$

$$T_{d+1}^2(n) = |x(n-(d_{\max}-1)+d) \\ -x'(n-(d_{\max}-1)+d)|$$

$$R_{d+1}^2(n)=R_d^2(n)+T_{d+1}^2(n)$$

  - Count the points for which

$$(\frac{T_{d+1}(n)}{R_d(n)}>R_{\text{tot}}) \text{ OR } (\frac{R_{d+1}}{R_A}>A_{\text{tot}})$$

  - If their fraction is smaller than 1% of q (as suggested in Kennel et al., 1992), choose d as the embedding dimension and terminate the procedure.

- End {for-loop}

[3] We use q instead of p observations for the estimation of d because a part of the data will be used for the evaluation of the results obtained by the various predictors, and it is assumed to be unknown.

In words, the above method tests if d is an appropriate estimate for the embedding dimension, by utilizing the neighborhood information of the $d_{\max}$-dimensional vectors $\boldsymbol{y}(n)$ in Z. Specifically, for each vector $\boldsymbol{y}(n)$ in Z, its nearest neighbor $\boldsymbol{y'}(n)$ is determined based on the last d coordinates of the vectors. Let $R_d^2(n)$ be the distance between $\boldsymbol{y}(n)$ and $\boldsymbol{y'}(n)$ when only the last d coordinates are taken into account. Then $R_{d+1}^2(n)$ is computed and the difference between $R_d^2(n)$ and $R_{d+1}^2(n)$ is considered. If $R_d^2(n)$ and $R_{d+1}^2(n)$ differ significantly, then we say that $\boldsymbol{y'}(n)$ is a false nearest neighbor of $\boldsymbol{y}(n)$ [4]. If this happens for a significant number of points $\boldsymbol{y}(n) \in Z$, it is an indication that the multi-dimensional structure of the system does not "unfold" well in the d dimensional space, i.e. a larger value of d must be considered.

Finally, it is worth noting that the above algorithm may also be used by considering not only the nearest neighbor of each vector $\boldsymbol{y} \in Z$ but also its k-nearest neighbors.

3 The predictors

3.1 Parametric predictors

3.1.1 The linear predictor

In this framework, the estimation of $x(n+1)$, denoted by $\hat{x}(n+1)$, is assumed to depend linearly on the values $x(n), x(n-1), \ldots, x(n-(d-1))$, i.e.

$$\hat{x}(n+1)=\sum_{i=0}^{d-1}w_i x(n-i)+w_d=[\boldsymbol{y}(n)\ 1]^T\boldsymbol{w}, \tag{5}$$

where $\boldsymbol{w}=[w_0,\ w_1, \ldots, w_{d-1},\ w_d]^T$ is the parameter vector of the predictor. Given a data set $Y=\{x(1), \ldots, x(q)\}$, $\boldsymbol{w}$ is chosen such that the following cost function is minimized

$$J(\boldsymbol{w}) = \sum_{n=d}^{q-d}(x(n+1)-\hat{x}(n+1))^2 \\ = \sum_{n=d}^{q-d}(x(n+1)-[\boldsymbol{y}(n)\ 1]^T\boldsymbol{w})^2. \tag{6}$$

It can be proven (see, e.g. Theodoridis et al., 2003) that the vector $\boldsymbol{w}$ that minimizes $J(\boldsymbol{w})$ is

$$\hat{\boldsymbol{w}}=(Z^TZ)^{-1}Z^T\boldsymbol{u}, \tag{7}$$

where $Z^T=[[\boldsymbol{y}(d)^T\ 1]^T,\ [\boldsymbol{y}(d+1)^T\ 1]^T, \ldots,\ [\boldsymbol{y}(q-d)^T\ 1]^T]$ and $\boldsymbol{u}=[x(d+1),\ x(d+2), \ldots,\ x(q-d+1)]^T$.

The estimated value of $x(n+1)$, $\hat{x}(n+1)$, is given by Eq. (5), where $\hat{\boldsymbol{w}}$ is used in place of $\boldsymbol{w}$.

[4] Consider, for example, the points $\boldsymbol{y_1}=[0.5,\ 0.5]^T$ and $\boldsymbol{y_2}=[0.55,\ 5.5]^T$. With respect to the first coordinate, their squared Euclidean distance is only 0.0025, while if both coordinates are taken into account, their squared Euclidean distance becomes approximately equal to 25.0025.

3.1.2 Neural networks predictor

In this study we consider only two-layer, feedforward neural networks ($2LFNN$), with m nodes in their hidden layer and a single output node[5]. These networks are modelled by the following equation

$$\hat{x} = g(\sum_{j=1}^{m} v_j f(\boldsymbol{w_j}^T [\boldsymbol{y}^T \ 1]^T) + v_0), \tag{8}$$

where $\boldsymbol{y}$ is the input vector and $\hat{x}$ is the output of the network. f is typically chosen to be equal to $\text{logi}(x)=1/(1+e^{-ax})$ or $\tanh(x)=(1-e^{-ax})/(1+e^{-ax})$, while g may be chosen to be equal to x, $\text{logi}(x)$ or $\tanh(x)$. The m $(d+1)$-dimensional vectors $\boldsymbol{w_j}$, as well as the values of v_j, $j=0,\ldots,m$ are the parameters of the network. Let $\boldsymbol{W}$ denote a vector that contains all these parameters. $\boldsymbol{W}$ is usually estimated by optimizing an appropriately defined cost function, using tools from nonlinear optimization theory. Given a data set $Y=\{x(1),\ldots x(q)\}$, a typical cost function that is frequently employed is the sum of square errors, defined as

$$J(\boldsymbol{W}) = \sum_{n=d}^{q-d} (x(n+1) - g\left(\sum_{j=1}^{m} v_j f(\boldsymbol{w_j}^T [\boldsymbol{y}^T(n) \ 1]^T) + v_0\right)\Bigg)^2. \tag{9}$$

The advantage of the above types of models is that they can describe more reliably phenomena that exhibit significant nonlinearities. However, their major disadvantage follows from the fact that the cost function to be optimized is non-convex, due to the nonlinear nature of f and (probably) g in Eq. (8). As a consequence, it is difficult to obtain the global optimum $\boldsymbol{W^*}$ of $J(\boldsymbol{W})$ that best represents the data at hand. Thus, instead of trying to determine the global optimum of $J(\boldsymbol{W})$, we seek for local optima of $J(\boldsymbol{W})$, which are (hopefully) suitable for the problem at hand. Their suitability is assessed through the test set.

3.2 Non-parametric predictors

3.2.1 The persistence predictor

In this case, the estimator of $x(n+1)$, $\hat{x}(n+1)$, is $x(n)$, that is $\hat{x}(n+1)=x(n)$. This simple predictor is expected to give satisfactory results in cases where the sample-to-sample variation is small, as is the case for periods where no significant disturbances occur in the ionosphere.

3.2.2 The k nearest neighbor predictor

In this case, for a given vector $\boldsymbol{y}(n)$, the predictor computes the estimate of $x(n+1)$ as follows. First, the k nearest neighbors, denoted by $\boldsymbol{y}(n_1)$, $\boldsymbol{y}(n_2)$, $\ldots$, $\boldsymbol{y}(n_k)$, of $\boldsymbol{y}(n)$ in S_1 are identified. Then, the estimate of $x(n+1)$ is taken to be equal to the mean of $x(n_1+1), x(n_2+1), \ldots, x(n_k+1)$. This method is met under the name "first order local approximation" in Tsonis (1992).

[5] See, e.g. Rummelhart et al. (1986); Pao (1989); Haykin (1994); Theodoridis et al. (2003).

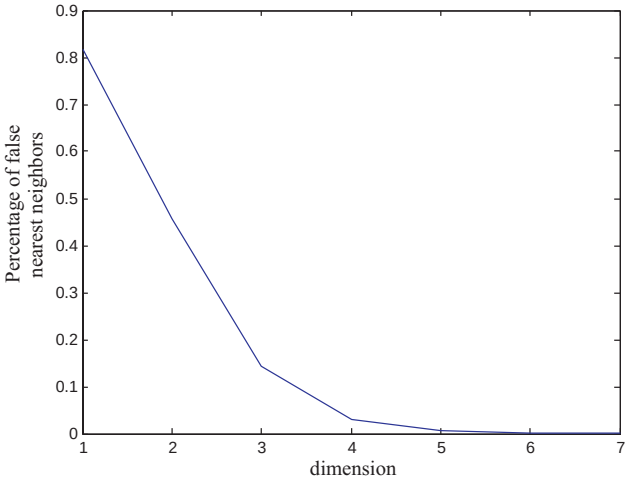

Fig. 1. Plot of the percentage of the false nearest neighbors versus the space dimension. For dimensions greater than or equal to 6 the percentage falls well below 1%.

4 Experimental results

Before we proceed with the generation of the training and the test sets, we need to estimate the embedding dimension d of the space where the dynamical system that produces the data sample X at hand "unfolds" in a satisfactory fashion its multi-dimensional structure. In applying the false nearest neighbor method described in Sect. 2 at the first half of the data sample X (that is $q=p/2$), we find that a good choice for d is 6. Specifically, for dimensions greater than or equal to 6, the percentage of false nearest neighbors falls well below 1% (see also Fig. 1)[6].

Having estimated d, we then describe the way the training and the test sets are generated. Specifically, the data sample X is split into two halves, X_1 (first half) and X_2 (second half). From each X_i, $i=1,2$, a corresponding set S_i, $i=1,2$ is generated as follows

$$S_1 = \{(\mathbf{y}(d), x(d+1)), \\ (\mathbf{y}(d+1), x(d+2)), \ldots, (\mathbf{y}(p/2-1), x(p/2))\} \tag{10}$$

and

$$S_2 = \{(\mathbf{y}(p/2+d), x(p/2+d+1)), (\mathbf{y}(p/2+d+1), \\ x(p/2+d+2)), \ldots, (\mathbf{y}(p-1), x(p))\}, \tag{11}$$

where d is chosen to be equal to 6, $\boldsymbol{y}(n)$ is defined as in Eq. (1) and the vectors $\boldsymbol{y}(n)$ with missing values are omitted.

All the predictors have been trained using S_1, and their performance has been measured on the test set S_2. The results are summarized in Table 1. Also, in Figs. 2, 3, 4 and 5 the histogram of the absolute differences between the actual and the estimated values on the test set, as well as the plot of the actual and the predicted values for a short time

[6] We note that the same value for d is taken if we consider the time delay T to be equal to 26 (see previous footnote).

Table 1. Fifteen-minute ahead prediction. The table shows the mean square error on the training set and on the test set for the linear predictor, the 2LFNN predictor with 4 nodes in the hidden layer, the persistence predictor and the k-nearest neighbor predictor, for k=12. In parentheses the standard deviation of the squared errors for each predictor on the test set is shown.

	Linear predictor	$2LFNN$ predictor (nodes=4)	Persistence predictor	k-nearest neighbor predictor (k=12)
Training set	0.1599	0.1462	0.1780	0.1311
Test set	0.1105 (0.3041)	0.1050 (0.2879)	0.1253 (0.3144)	0.1272 (0.3165)

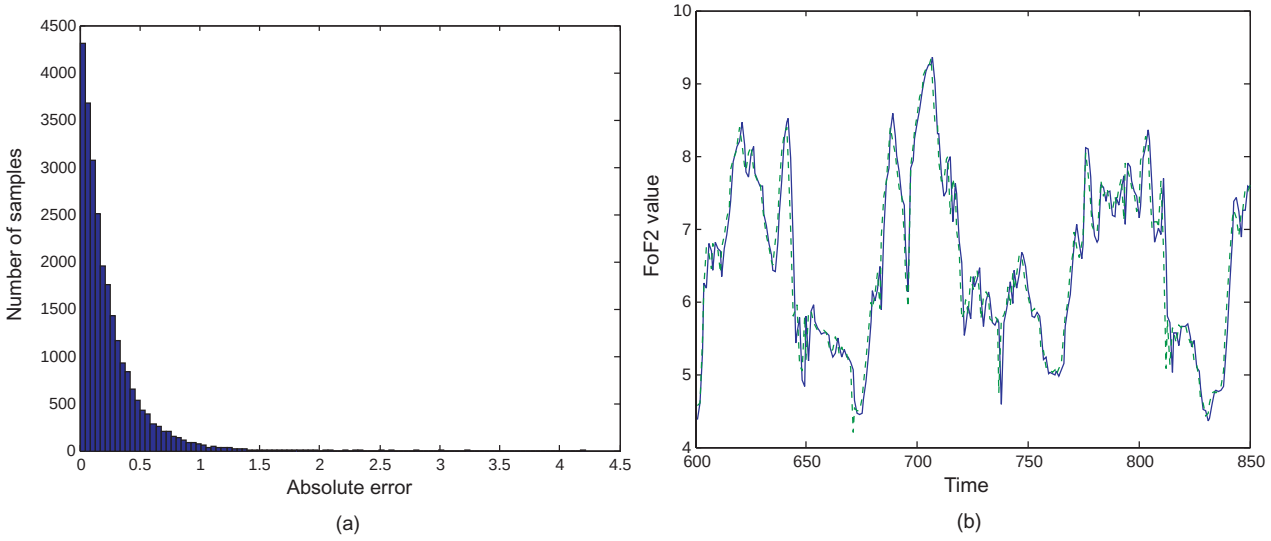

Fig. 2. One-step ahead (15 min) prediction, with the linear predictor. **(a)** The histogram of the absolute differences of the computed and the actual outputs, for the test set. **(b)** The actual (solid line) and the computed (dotted line) outputs for a short time interval of the test set.

interval, are given for each of the four predictors. It is noted that various $2LFNN$ architectures with different numbers of hidden layer nodes have been examined. Specifically, $2LFNN$s with up to m=50 nodes in the hidden layer have been considered. However, the best performance was obtained for m=4 nodes. We also note that the mean square error (MSE) value for the $2LFNN$s, shown in Table 1, is the average of the MSEs of 10 networks, with m=4 hidden nodes, that have been trained with the Levenberg-Marquardt algorithm starting from different initial values for the parameters. Also, for the k-nearest neighbor predictor, the results for k=1, 2, . . . , 30 have been considered. Here, only the best results are provided.

As can be seen from the results shown in Table 1 (and supported by Figs. 2, 3, 4 and 5), all predictors seem to exhibit more or less a similar performance. However, in order to quantify the significance of the differences among the mean square errors (MSE) produced by any pair of the above classifiers when they are applied on the test set, we use the t-test statistic (see, e.g. Mendenhall et al., 1995). The choice of this test is justified by the fact that the errors produced by the predictors are independent, since each one of the predictors follows a different prediction strategy from all the others. More specifically, for any two of the above predictors, $\mathcal{P}_1$ and $\mathcal{P}_2$, we test the hypothesis

$$H_0 : \text{MSE}_1 - \text{MSE}_2 = 0$$

against

$$H_1 : \text{MSE}_1 - \text{MSE}_2 > 0,$$

where MSE_i is the mean square error (MSE) for $\mathcal{P}_i$, i=1, 2. Denoting the sample MSE for $\mathcal{P}_i$ (as it is given in Table 1) by $\overline{\text{MSE}}_i$ and assuming that $\overline{\text{MSE}}_1 > \overline{\text{MSE}}_2$, we compute the quantity

$$z = \frac{\overline{\text{MSE}}_1 - \overline{\text{MSE}}_2}{\sqrt{\frac{s_1^2+s_2^2}{n}}}, \tag{12}$$

where s_i is the sample data deviation of the squared errors produced by $\mathcal{P}_i$, and n is the number of samples (in our case n=25605). The values of z for all pairs of predictors are shown in Table 2. Setting the level of significance a equal to

Table 2. Fifteen-minute ahead prediction. The table shows the values of the t-test that quantify the significance of the differences in the mean square error produced by two predictors when the test set is considered.

	Linear predictor	$2LFNN$ (nodes=4)	Persistence predictor	k-nearest neighbor predictor (k=12)
Linear predictor		2.1016	5.4143	6.0883
2LFNN predictor			7.6197	8.3027
Persistence pred.				0.6815
k-nearest neigh.				

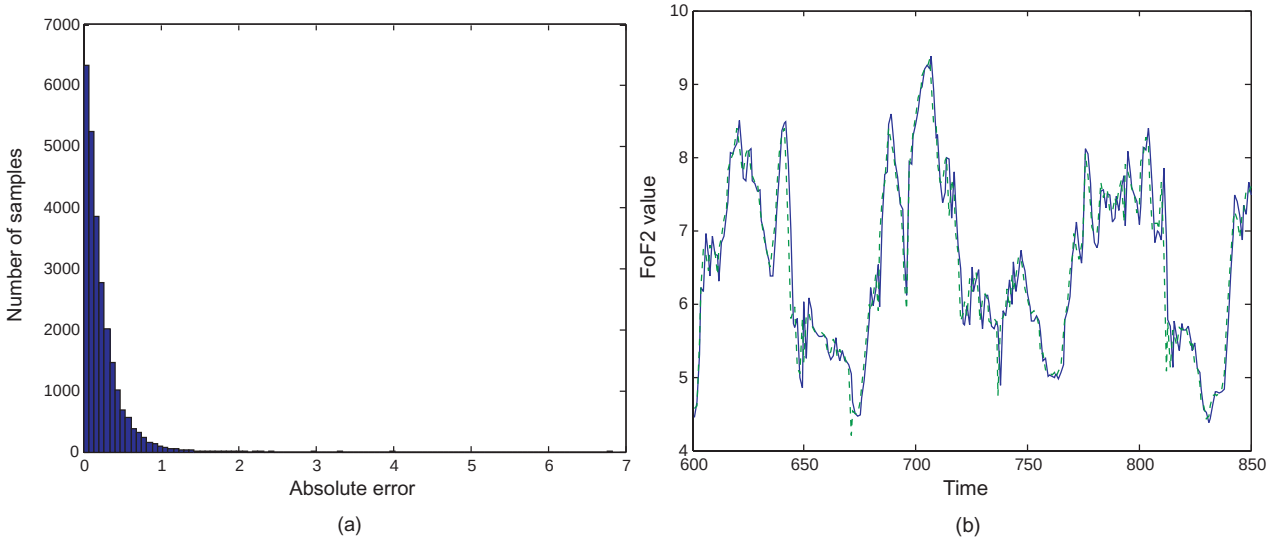

Fig. 3. One-step ahead (15 min) prediction, with the $2LFNN$ model with 4 nodes. **(a)** The histogram of the absolute differences of the computed and the actual outputs, for the test set. **(b)** The actual (solid line) and the computed (dotted line) outputs for a short time interval of the test set.

0.01 and taking into account the above value of n, the value z_a for the t-test is 2.326. Recalling that the H_0 hypothesis is rejected when $z > z_a$, the values in the table lead to the following conclusions:

- At significance level 0.01, there is not enough sufficient evidence to reject the hypothesis that the MSE for the linear predictor and neural network predictor are equal[7].
- At significance level 0.01, there is not enough sufficient evidence to reject the hypothesis that the MSE for the persistence predictor and k-nearest neighbor predictor are equal.
- At significance level 0.01, the MSEs for the non-parametric predictors differ significantly from the MSEs for the parametric predictors.

Adopting the Occam's razor principle, that is seeking for the simplest model that best describes the observed data and taking into account the above analysis, a linear model seems to be sufficient for one-step ahead predictions.

Focusing on the performance of the various predictors on the training set, we notice that the k-nearest neighbor predictor exhibits the best performance. The fact that this predictor exhibits the worst performance on the test set may be taken as an indication that the k-nearest neighbor predictor exhibits some degree of overfitting on the training set [8]. On the contrary, no such conclusion is supported from the above results for the other three classifiers.

5 Concluding remarks and future directions

In this paper we considered the problem of performing one-step ahead predictions on the $foF2$ parameter, using time

[7] However, at significance level 0.05 the H_0 hypothesis is rejected, since $z_{0.05}$=1.645.

[8] We say that a predictor *overfits* the training data, if it learns all the peculiarities of the specific training set (in other words, it memorizes the specific training set), instead of learning only its general structure. As a consequence, such a predictor does not behave well on data sets different from the one used for training.

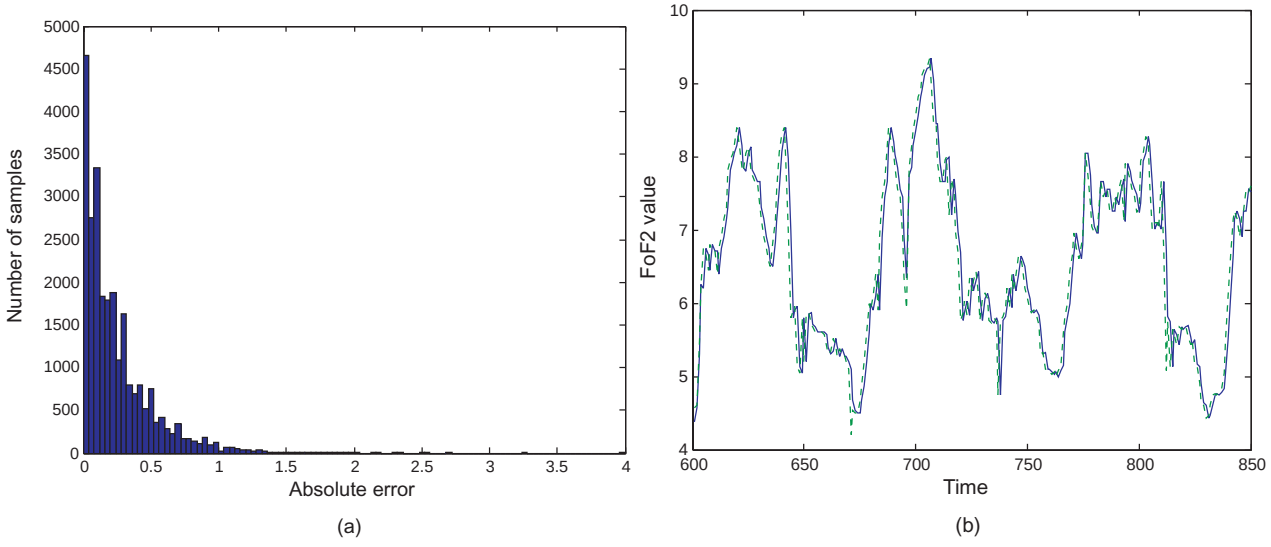

Fig. 4. One-step ahead (15 min) prediction, with the persistence predictor. **(a)** The histogram of the absolute differences of the computed and the actual outputs, for the test set. **(b)** The actual (solid line) and the computed (dotted line) outputs for a short time interval of the test set.

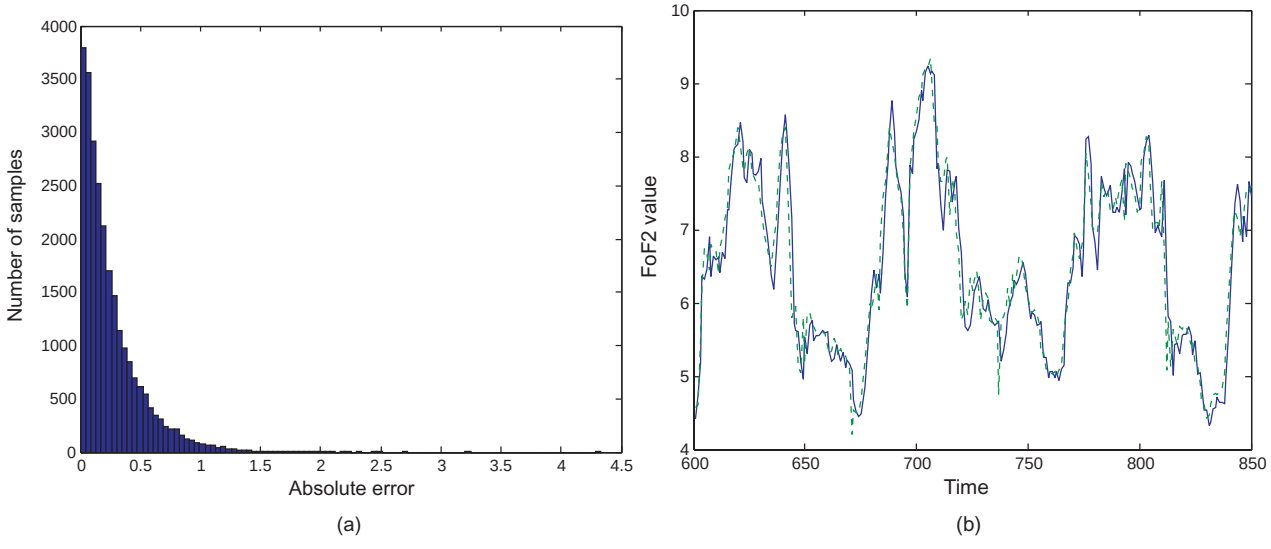

Fig. 5. One-step ahead (15 min) prediction, with the 9 nearest neighbor predictor. **(a)** The histogram of the absolute differences of the computed and the actual outputs, for the test set. **(b)** The actual (solid line) and the computed (dotted line) outputs for a short time interval of the test set.

series forecasting methods. Specifically, assuming that n denotes the current time slot, the purpose is to estimate $x(n+1)$ based on $x(n), \ldots, x(n-(d-1))$, where $\{x(n)\}$ denotes the $foF2$ time series. Our first concern was to estimate the value of d, the dimension of the space where the dynamical system that generates the observed $foF2$ measurements is embedded. This was carried out by applying the false nearest neighbor method. Then, based on the estimated value of d, we generated the appropriate training and test sets for the training and the evaluation of the performance of four well-known predictors: the linear predictor, the two-layer, feed-forward neural network predictor, the persistence predictor and the k-nearest neighbor predictor.

The results show that the parametric predictors work significantly better than the non-paramtric ones. In addition, the performance of the linear predictor does not vary significantly from the neural network predictor. The latter fact may be taken as an indication that a linear model suffices for one-step ahead prediction of $foF2$. In addition, it seems that the k-nearest neighbor predictor exhibits some degree of overfitting on the training set.

Below, we briefly give some future guidelines for further investigation. First, we intend to apply nonlinearity tests on the observed data series, in order to gain some further insight on the nature of the process that produces the observed foF2 measurements (see, e.g. Schreiber and Schmitz, 2000).

Furthermore, an interesting variation during the training of the predictor would be to supply additional information related to the presence or the absence of a disturbance.

In addition, it seems interesting to see how the above predictors can be adapted in a time-varying environment. In such an environment the time series at hand exhibits significant variations in time and, thus, the predictor has to adapt its parameters in order to be able to follow these changes. In this case we say that we deal with adaptive predictors. In the prediction of the foF2, the above idea may be utilized as follows: first, we use the data of a short time period to train a specific predictor. Then, this predictor is used with the current parameter values for prediction for a short time period in the future. Then, its parameters are re-evaluated in light of the new observations and the procedure is repeated.

Finally, an obvious extension of the above work is the multi-step ahead prediction, where, of course, the error estimate is expected to increase, compared to that of the one-step ahead prediction. It should be noted however, that allowing the time delay T to take values other than 1, interesting results may be obtained in this direction. For example, if we set $T=26$ (which corresponds to 6.5 h, since the sampling rate for the data set at hand is 15 min), the predictions exhibits a mean square error slightly greater than 1 MHz.

Acknowledgements. A preliminary version of this work was presented in the first European Space Weather week as a contribution to the COST724 European action. Part of this work was funded by the DIAS project, sponsored by the eContent programme of the European Commission. The authors would like to thank the two reviewers for their constructive comments.

Topical Editor M. Pinnock thanks E. Tulunay and another referee for their help in evaluating this paper.

References

Anderson, D. N., Buonsanto, M. J., Codrescu, M., Decker, D., Fesen, C. G., Fuller-Rowell, T. J., Reinisch, B. W., Richards, P. G., Roble, R. G., Schunk, R. W., and Sojka, J. J.: Intercomparison of physical models and observations of the ionosphere, J. Geophys. Res., 103, 2179–2192, 1998.

Fuller-Rowell, T. J., Codrescu, M. V., and Araujo-Pradere, E.: Capturing the storm-time F-region ionospheric response in an empirical model, AGU Geophysical Monograph, 125, 393–402, 2001.

Haykin, S.: Neural Networks: A comprehensive foundation, McMillan, 1994.

Hegger, R., Kantz, H., and Schreiber, T.: Practical implementation of nonlinear time series methods: The TISEAN package, Chaos, 9, 413–440, 1999.

Kennel, B. K., Brown, R. and Abarbanel, H. D. I.: Determining embedding dimension for phase-space reconstruction using a geometrical construction, Physical Review A, 45(6), 3403–3411, 1992.

McKinnell, L. A. and Poole, A. W. V.: The development of a neural network based short term foF2 forecast program, Phys. Chem. Earth, Part C, 25(4), 287–290, 2000.

Mendenhall, W. and Sincich, T.: Statistics for engineering and the sciences, Prentice Hall, 4th edition, 1995.

Muhtarov, P. and Kutiev, I.: Autocorrelation method for temporal interpolation and short-term prediction of ionospheric data, Radio Science, 34(2), 459–464, 1999.

Muhtarov, P., Kutiev, I., and Cander, L.: Geomagnetically correlated autoregression model for short-term prediction of ionopsheric parameters, Inverse Problems, 18, 49–65, 2002.

Pao, Y.-H.: Adaptive pattern recognition and neural networks, Addison-Wesley, 1989.

Rumelhart, D. E. and McClelland, J. L.: Parallel distributed processing: Explorations in the microstructure of cognition. Vol. 1: Foundations, Cambridge, MA: MIT Press, 1986.

Schreiber, T. and Schmitz, A.: Surrogate time series, Physica D, 142, 4092–4120, 2000.

Theodoridis, S. and Koutroumbas, K.: Pattern Recognition (2nd edition), Academic Press, 2003.

Tsonis, A. A.: Chaos: From theory to applications, Plenum Press, 1992.

Tulunay, Y., Tulunay, E., and Senalp, E. T.: The neural network technique – 1: a general exposition, Adv. Space Res., 33, 983–987, 2004a.

Tulunay, Y., Tulunay, E., and Senalp, E. T.: The neural network technique – 2: an ionospheric example illustrating its application, Adv. Space Res., 33, 988–992, 2004b.

Wintoft, P. and Cander, L. R.: Twenty-four hour predictions of foF2 using time delay neural networks, Radio Science, 35, 2, 395–408, March–April 2000.