

RESEARCH ARTICLE

WILEY

Nonlinear prediction of functional time series

Haixu Wang | Jiguo Cao^{ID}

Department of Statistics and Actuarial
Science, Simon Fraser University,
Burnaby, British Columbia, Canada

Correspondence

Jiguo Cao, Department of Statistics
and Actuarial Science, Simon Fraser
University, Burnaby, BC V5A 1S6, Canada.
Email: jiguo_cao@sfu.ca

Funding information

Natural Sciences and Engineering
Research Council of Canada

Abstract

We propose a nonlinear prediction (NOP) method for functional time series. Conventional methods for functional time series are mainly based on functional principal component analysis or functional regression models. These approaches rely on the stationary or linear assumption of the functional time series. However, real data sets are often nonstationary, and the temporal dependence between trajectories cannot be captured by linear models. Conventional methods are also hard to analyze multivariate functional time series. To tackle these challenges, the NOP method employs a nonlinear mapping for functional data that can be directly applied to multivariate functions without any preprocessing step. The NOP method constructs feature space with forecast information, hence it provides a better ground for predicting future trajectories. The NOP method avoids calculating covariance functions and enables online estimation and prediction. We examine the finite sample performance of the NOP method with simulation studies that consider linear, nonlinear and nonstationary functional time series. The NOP method shows superior prediction performances in comparison with the conventional methods. Three real applications demonstrate the advantages of the NOP method model in predicting air quality, electricity price and mortality rate.

KEYWORDS

dimension reduction, functional data analysis, nonlinear forecast, neural networks

1 | INTRODUCTION

Functional data analysis (FDA) is a growing statistical field for analyzing longitudinal trajectories, curves, images or any manifold objects. One unique feature is that FDA treats each random function observed over the whole domain as a sample element. Functional data are commonly found in many applications such as fitness data from the wearable device, air pollution, longitudinal studies, time-course gene expressions and brain images. Traditionally, functional data are assumed to be independent. When functional data are observed at a sequence of time points, such functional data have temporal correlation and are called functional time series. In this article, we propose a nonlinear forecast model for functional time series.

One conventional framework for analyzing functional time series is based on functional principal component analysis (FPCA). FPCA is a popular method to perform dimension reduction for functional data. It provides low-dimensional feature representations for individual trajectories, and these features can be used to facilitate future modeling tasks. FPCA

This is an open access article under the terms of the [Creative Commons Attribution](https://creativecommons.org/licenses/by/4.0/) License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

© 2023 The Authors. *Environmetrics* published by John Wiley & Sons Ltd.

finds major sources of variation in functional data, which are quantified as functional principal components (FPCs). These FPCs can span a feature space to project individual functional data. That is, individual trajectories are expressed by a linear combination of FPCs. The coefficients in the linear combinations are FPC scores, which are the projections of individual functional data onto the space spanned by FPCs. By selecting a fixed number of FPCs, we can obtain a low-dimensional space to project functional data and use the obtained FPC scores to represent original functional data. Building subsequent models based on FPC scores is more efficient than using original functional data. Textbooks by Ramsay and Silverman (2005) and Vieu and Ferraty (2006) gave a good summary of both FDA and FPCA. The theoretical properties of FPCA have been well investigated by Dauxois et al. (1982), Hall and Hosseini-Nasab (2005), and Hall et al. (2006). Standard FPCA methods focus on finding the major modes of variations among functional data and does not have easy interpretations. Chen and Lei (2015), Lin et al. (2015), and Nie and Cao (2020) imposed sparsity constraints to obtain more interpretable FPCs. Sang et al. (2017) introduced a parametric approach to estimate top FPCs to enhance their interpretability. Nie et al. (2018) proposed supervised FPCA methods by considering the response information. Shi et al. (2021) introduced the informatively missing FPCA method for applications where the longitudinal trajectories are subject to informative missingness. Shi and Cao (2022) developed a robust method based on a new regression framework when the functional data have outliers. In addition, Dai and Müller (2018) considered functional data defined on some Riemannian manifolds where standard FPCA is not applicable. It is common in applications that individual curves are not fully observed. Nie et al. (2022) proposed a sparse orthonormal approximation method to recover the underlying trajectories from sparse and irregular functional data.

FPCA has been well studied for univariate functional data, but cannot be directly applied to multivariate trajectories. The standard approach for dealing with multivariate functional data is introduced by Ramsay and Silverman (2005). It suggests joining multivariate functions into a univariate one, that is, connecting starting points and endpoints of trajectories from all functional variables. Standard FPCA can then be applied to the resulting univariate functions. Following this approach, Chiou et al. (2014) incorporated a normalization approach to balance the degrees of variations across different functional variables. In the meantime, Happ and Greven (2018) investigated the case that trajectories from different functional variables are observed over different domains. Berrendero et al. (2011) pursued another approach by applying principal component analysis on observed multivariate trajectories at available observation points. Then, it connected the obtained principal components over observation points as the FPCs.

Functional regression models are another class of methods used for modeling functional time series. Ramsay and Silverman (2005) presents a good summary of regression models dealing with scalar and functional responses, where the latter case suits our interest in functional time series. Morris (2015) reviewed the applications and developments of functional regression models. Regression models for functions can be simplified by projecting functional data onto smooth eigenfunctions. Yao et al. (2005) utilized such an approach for estimating functional linear models, and Müller and Stadtmüller (2005) extended the method for generalized functional linear models. Kong et al. (2018) introduced the robust estimation method for functional linear models. James et al. (2000) proposed principal component models for sparse functional data. Guo (2002) considered mixed effects models where both the fixed and random effects are functions. Scheipl et al. (2013) extended mixed effect models for multiple functional predictors. Furthermore, Fan et al. (2015) and Müller (2008) studied the functional additive models for both scalar and function responses.

Both FPCA and functional regression models have been applied to analyze functional time series and establish prediction models for future trajectories. Aue et al. (2015) and Hyndman and Ullah (2007) proposed to forecast future trajectories based on the predicted functional principal component scores. Hyndman and Shang (2009) and Kargin and Onatski (2008) used a regression model by treating the future trajectory as the response and the last observed one as the predictor. It is sometimes equivalent to the autoregressive operator for curves. Jiao et al. (2019) further refined the predictions by using partially observed future trajectories. These methods depend on the stationarity and linearity of the functional time series. For the theories of functional time series, Bosq (2000) discussed theoretical properties of time series in function spaces. Later, Horvath et al. (2013) and Horvath et al. (2014) investigated the stationarity of function time series and whether it has the same mean function over time. By framing the functional time series as an autoregressive model, Hörmann and Kokoszka (2010) illustrated how to define and check the length of the temporal dependence of the series.

There are some recent advances in the analysis and prediction of functional time series. Hörmann et al. (2015) proposes to use the spectral density operator and obtains dynamic FPC scores instead of static ones. Elayouty et al. (2022) analyzes nonstationary functional time series by considering a dynamic functional principal component analysis. In addition, Gao et al. (2019) addresses the scenario where the functional time series is multivariate or high-dimensional. Elias et al. (2022) uses a nonparametric way to predict future curves by using the nearest ones. Furthermore, Alavarez Liebana (2017) has outlined a review of other approaches to analyzing functional time series.

The existing methods for predicting functional time series are mainly based on FPCA and functional regression models. There are several limitations to these types of approaches. First, these approaches often operate on the stationary assumption of functional time series. For FPCA-based approaches, this assumption specifies a constant mean and covariance function for all the trajectories from the observed series and future ones. A prediction model is then established in the space defined by FPCs, and we can forecast future trajectories by their predicted FPC scores. One limitation of this approach is related to the validity of the stationary assumption. Real data often do not follow a stationary process, hence the functional principal subspace and FPC scores cannot provide satisfactory predictions when the assumption is violated. Therefore, the prediction performance greatly depends on how well the stationary assumption holds for the observed series and future time points. In fact, both the mean and covariance function may vary over time and be influenced by covariates that are also depending on time. For example, the observed trajectories of electricity prices on a given day may have seasonal or weekend/weekday effects. The second limitation of FPCA-based approaches is whether the FPC scores are the most optimal representations of functional data with respect to the prediction task. FPC scores only contain information of variations among functional covariates, and that information may not be well translated for other models. The third limitation of FPCA-based approaches is that FPC scores are linear projections that cannot capture nonlinear structures in functional data. Consequently, we need to pursue other representation models defining nonlinear mapping rather than linear projections. Another limitation for regression-based approaches is also related to the nonlinearity in functional data. In this case, we need to consider the nonlinear temporal dependence between trajectories. Regression-based approaches use a linear operator to specify the dependence and cannot address more complex relationships between trajectories.

In addition, both FPCA and regression-based approaches are limited to univariate functional time series. The aforementioned limitations will become more influential in a multivariate setting because we have to consider linearity and stationarity across dimensions. One standard technique for dealing with multivariate functions is to connect functions into a univariate one and apply FDA methods to the resulting univariate function. As a result, the predictions on the boundaries of each function are often unsatisfactory. This can become a significant drawback when we are predicting multivariate functional time series. Traditional prediction models are often obtained for a fixed data set. We may need to consider the case where data are streamlined or continuously updated. When the data size is large or new trajectories are continuously observed, re-estimating the model can be computationally burdensome if it includes the calculation of covariance functions. Therefore, predicting functional time series in an online setting is more ideal, such that both the model and predictions can be updated and generated whenever a new observation becomes available.

In this article, we have selected two real environmental data sets that are challenging to the existing methodologies of functional time series. The first one includes measurements of chemical pollution in daily air quality. It is a multivariate functional time series where temporal dependencies between days present nonstationary behaviors and nonlinear relationships. Furthermore, such data often includes seasonal effects which further violates the stationarity assumption required by standard methods from FDA. In the application section, we will demonstrate that the proposed method outperforms existing methodologies in terms of predicting future data because our motivation is to propose a model that does not rely on stationarity or linearity of functional time series. The second data set is the daily electricity price of northern Italy. This data set is more unstable than the air quality data. It can easily have fluctuations or shrinkages due to human or environmental factors compared to the air quality data. For example, the price can have a sudden drop on weekends or holidays which do not follow the general trend of previous days. Our method is able to consider external factors in addition to the intrinsic temporal dependencies of the functional time series, hence it can produce better predictions for such kind of functional time series. These two examples demonstrate the importance of the proposed method to environmental data sets which usually exhibit complex dependence and nonstationarity structures due to internal and external factors to the system under study.

In this article, we propose a nonlinear prediction (NOP) model for functional time series. It is able to predict future trajectories that are either univariate or multivariate. The proposed model addresses the aforementioned limitations and has the following advantages over traditional FDA methods. First, the major advantage of our work is to avoid the calculation requirement of covariance functions. The motivation is to construct a latent feature space not depending on the decomposition of covariance functions. Instead, we propose to use a direct nonlinear function ϕ for mapping from the data space (containing curves in the functional time series) to a latent feature space. Assuming that the feature space is a low-dimensional space, any observed trajectories are compressed in vectors that preserve as much information as possible. The nonlinear mapping is more flexible to capture nonlinear structures in functional data. Moreover, we expect the latent feature space to capture the evolution of trajectories and to be less dependent on the stationarity of functional time series. Explicitly specifying ϕ can be exhausting, we employ neural networks as a powerful functional approximation tool

to obtain the nonlinear mapping. Furthermore, we allow the latent feature space to be constructed based on prediction information whereas the functional principal subspace only focuses on variations among functional covariates. Hence, the obtained latent feature space can provide a better ground for predicting future trajectories.

One advantage of using the nonlinear mapping ϕ is that it can be directly applied to a multivariate function. It makes the proposed NOP model more suitable for multivariate functional time series. In addition to applying a nonlinear mapping on trajectories, we introduce a nonlinear function γ to model the temporal relationship between trajectories through their vector representations. The nonlinear γ is similarly approximated by neural networks, and it is flexible when we need to adjust the length of temporal dependence or consider additional covariates into the prediction model. When we forecast the time series for a longer period in advance, the predictions will be more stable under some mild stationary restrictions. The restrictions can be enforced with regularization in the estimation process. Regularization can also help the prediction model from overfitting when the training sample size is small. The last advantage of our NOP model is the ability to perform online estimation and prediction. Because the NOP model does not need to estimate covariance functions, it is straightforward to update the nonlinear functions of our model given any new observation. For our proposed method and neural network-based approaches, updating the current model is roughly equivalent to a single iteration of stochastic gradient update given the new observations. On the other hand, approaches based on functional principal component analysis require the calculation of covariance function which uses all observations. Our model does not need to use all observations at once, and it only requires a pair of observations for updating and training the model.

The remainder of the article is organized as follows. Section 2 presents the model with its neural network implementation and estimation process. It illustrates the design of the training objective which is able to use forecast information to construct a latent representation space. Section 3 summarizes simulation results for three types of functional time series: linear, nonlinear, and with a trend function. Section 4 presents the applications of the proposed model on electricity price, French mortality, and British Columbia air quality data sets. The first resembles a stationary functional time series, whereas the latter two appear to have a trend function and cyclic behaviors respectively. For each application, we will compare the proposed model with approaches based on FPCA and functional regression. Section 5 concludes the article.

2 | THE NONLINEAR PREDICTION MODEL

2.1 | Model specification

Let $f_i(x)$, $i = 1, \dots, n$, be the functional time series observed at time points t_1, \dots, t_n . At each time t , the function $f_t(x)$ is a square-integrable function over the interval $[a, b]$. For simplicity, we assume that all n trajectories are evaluated at the same set of points x_1, \dots, x_J . In this article, we will consider a one-step ahead prediction model as follows:

$$\begin{aligned} \mathbf{z}_t &= \phi(f_t(x)), \\ \mathbf{z}_{t+1} &= \gamma(\mathbf{z}_t), \\ f_{t+1}(x) &= \psi(\mathbf{z}_{t+1}), \end{aligned} \quad (1)$$

where ϕ , ψ , and γ are some nonlinear functions to approximate. The prediction model 1 contains three components as nonlinear functions. First, the function ϕ is responsible for mapping a random function $f_t(x)$ in the data space \mathcal{F} to a latent feature space $\mathcal{Z} \subset \mathbb{R}^D$ for a chosen D . The vector representation \mathbf{z}_t is responsible for conveying and compressing information in the corresponding trajectory. Building a prediction model and tracing out the evolution of functional time series will be easier by using $\{\mathbf{z}_t\}$'s than original functional data. If the number of observations J in each function is large, then low-dimensional representations greatly reduce subsequent modeling efforts. Explicitly defining a nonlinear function is difficult and not flexible with different data sets.

We will utilize neural networks to approximate the nonlinear function ϕ and allow it to be data-dependent. In order to preserve the continuity of trajectories, we use a recurrent neural network to approximate ϕ , that is,

$$\begin{aligned} \phi(f_t(x)) : \quad \mathbf{h}_j &= u_h(\mathbf{W}\mathbf{h}_{j-1} + \mathbf{U}f_t(x_j)) \\ \mathbf{z}_t &= u_z(\mathbf{V}\mathbf{h}_J), \end{aligned} \quad (2)$$

where \mathbf{W} , \mathbf{U} , and \mathbf{V} are parameter matrices of the neural network. Both u_h and u_z are activation functions of neural networks. The hidden unit \mathbf{h}_j is a D -dimensional vector in correspondence to the discrete observations $f_i(x_j)$ of any trajectory. The hidden unit \mathbf{h}_j processes and propagates information in parallel to a given trajectory. We assume that all information $f_i(x)$ can be conveyed by \mathbf{h}_j . A nonlinear mapping on $f_i(x)$ is achieved by applying a nonlinear mapping on \mathbf{h}_j as shown as the second equation in Equation (2).

After obtaining the latent representation \mathbf{z}_t of a trajectory $f_t(x)$, we propose to predict the future $f_{t+1}(x)$ through the evolution from \mathbf{z}_t to \mathbf{z}_{t+1} . In this article, we consider a simple one-step ahead prediction model $\gamma(\mathbf{z}_t)$, and the nonlinear function is estimated by a feed-forward neural network as follows:

$$\mathbf{z}_{t+1} = \gamma(\mathbf{z}_t) = u_\gamma(\beta \mathbf{z}_t), \quad (3)$$

where u_γ is the chosen activation function and β is the matrix of parameters in the neural network. There is great flexibility in modifying the prediction model based on latent representations \mathbf{z}_t 's. First, we can include other scalar covariates in the prediction model. We will use g_{t+1} to represent all the possible covariates as long as they are available before observing the trajectory $f_{t+1}(x)$. The covariates in g_{t+1} can be easily incorporated into the prediction model γ as the following:

$$\mathbf{z}_{t+1} = \gamma([\mathbf{z}_t, g_{t+1}]), \quad (4)$$

where we concatenate two vectors \mathbf{z}_t and g_{t+1} into a single one. For example, we can define $g_{t+1} = \{0, 1\}$ given whether the day is a weekday or weekend. Another modification can be made on the length of temporal dependence or prediction. In the specification (1), we consider the simplest prediction form $\mathbf{z}_{t+1} = \gamma(\mathbf{z}_t)$. The temporal dependence can be extended as $\gamma(\mathbf{z}_{t-h:t})$. In addition, we can allow $\mathbf{z}_{t+1:t+h} = \gamma(\cdot)$ to produce predictions for a future period $[t+1, t+h]$ rather than on only one point $t+1$. The flexibility of γ highlights the importance of using latent feature representations for forecasting functional time series.

If ϕ is a linear transformation or an explicit nonlinear function, it is feasible to specify an inverse mapping from \mathcal{Z} to \mathcal{F} . However, the nonlinear mapping ϕ is approximated by neural networks, and we need to directly estimate an inverse mapping $\psi: \mathcal{Z} \rightarrow \mathcal{F}$. The mapping ψ can be applied to latent feature representations $\mathbf{z}_{1:t}$ for observed trajectories or any future ones $\mathbf{z}_{t+1:t+h}$. In order to produce reconstructions that preserve the continuity of curves, we will again use a recurrent neural network to approximate ψ as follows:

$$\begin{aligned} \psi(\mathbf{z}_t) : \quad \tilde{\mathbf{h}}_j &= u_{\tilde{h}}(\mathbf{M}\tilde{\mathbf{h}}_{j-1} + \mathbf{G}\mathbf{z}_t) \\ f_i(x_j) &= u_f(\mathbf{L}\tilde{\mathbf{h}}_j), \end{aligned} \quad (5)$$

where $u_{\tilde{h}}$ and u_f are the activation functions, and \mathbf{M} , \mathbf{G} , and \mathbf{L} are the parameter matrices of the neural network.

The prediction process is also illustrated in Figure 1. The top row in Figure 1 illustrates the prediction routine. The prediction of the future trajectory at $t+1$ is obtained by using the last observed curve. The temporal dependence between adjacent trajectories is modeled through a nonlinear function γ with latent representations. Once the predicted $\hat{\mathbf{z}}_{t+1}$ becomes available, we can generate the predicted trajectory $\hat{f}_{t+1}(x)$ with $\psi(\hat{\mathbf{z}}_{t+1})$.

2.2 | Model estimation

For estimating all the nonlinear functions ϕ , ψ , and γ , we can consider the following objective function:

$$\begin{aligned} \mathcal{L} &= \mathcal{L}_1 + \mathcal{L}_2 + \mathcal{P} \\ &= \sum_{t=1}^{n-1} l(f_i(x), \psi(\phi(f_i(x)))) + \sum_{t=2}^n l(f_i(x), \psi(\gamma(\mathbf{z}_{t-1}))) + \lambda \sum_{t=2}^n p(\phi(f_i(x)), \gamma(\mathbf{z}_{t-1})), \end{aligned} \quad (6)$$

where l is a chosen loss function and p is a penalty function. For example, we will set l to be a squared distance function evaluated based on any pair of trajectories. The penalty function p can be a squared distance function or the Kullback–Leibler divergence term for any two vectors of representations. The penalty parameter λ controls the contribution of the penalty term \mathcal{P} in the training objective.

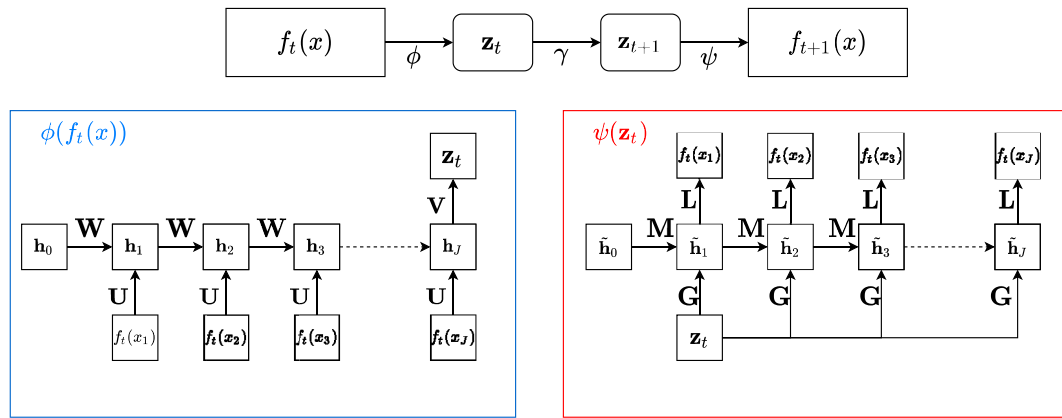


FIGURE 1 The flow of prediction model (at the top) and the construction of model components, ϕ and ψ . The network implementation for ϕ is enclosed by the blue box where the red box demonstrates that of ψ .

Let's take a closer look at each component in the training objective (6). First, the component \mathcal{L}_1 focuses on the estimation of ϕ and ψ by using trajectories that have been observed. We will let l be the squared loss function and $\hat{f}_t(x) = \psi(\phi(f_t(x)))$ to be the reconstruction of a given trajectory. Then, $l(f_t(x), \hat{f}_t(x)) = \sum_{j=1}^J (f_t(x_j) - \hat{f}_t(x_j))^2$ shows the fitting of these two nonlinear functions. It can also demonstrate how well the latent vector z_t can conserve information and be used to recover underlying trajectory given noisy and discrete observations. The second component \mathcal{L}_2 is evaluated based on future trajectories when they become available for updating the model. This component is responsible for the estimation of the predictor γ and reinforces the reconstruction function ψ .

The third component \mathcal{P} is a penalty term to avoid overfitting the training data and balance the contributions between ϕ and γ . Excluding the first trajectory $f_1(x)$, each observed trajectory is used twice in calculating the training objective. That is, we will be obtaining two latent representations for each $f_t(x)$ where $t = 2, \dots, n-1$. One z_t is obtained by applying ϕ and the other is produced by the prediction from $\gamma(z_{t-1})$. For example of a squared distance function $p(\cdot)$, a single penalty $p(\phi(f_t(x)), \psi(\gamma(z_{t-1})))$ is minimized if two representations are the same, and it increases as the prediction of z_t becomes different to its supposed representation obtained after observing the trajectory. This controls ϕ from overfitting the future trajectories in the tuples $(f_t(x), f_{t+1}(x))$, and it only applies ϕ on $f_{t+1}(x)$ when it becomes available. The penalty term also forces γ to use the feature space rather than solely predicting future trajectories, that is, it is subject to both \mathcal{L}_2 and \mathcal{P} instead of a single prediction objective. As a result, the penalty \mathcal{P} balances the contributions of both ϕ and γ and improves the future predictions by avoiding overfitting the training data.

This penalty term does not only prevent the model from overfitting the training data but also adjusts the long-term predictions. The long-term predictions in the one-step ahead model 1 is obtained iteratively from the first predicted trajectory $\hat{f}_{n+1}(x)$, such that $\hat{f}_{n+h}(x) = \psi(\gamma(\phi(\hat{f}_{n+h-1}(x))))$. With the penalty term \mathcal{P} , we are able to enforce the stationary assumption proportional to the choice of λ . Larger penalty parameter λ forces the predicted latent representations to approach the supposed mappings of future trajectories. It suggests that the latent space should not change too much given the new observations and equivalently puts some stationarity constraints on the functional time series including unseen observations. If the functional time series is stationary, we can show that the model with a larger penalty has better long-term predictions than a smaller penalty in the later application examples. On the other hand, if our interest is in the short-term or only one-step ahead predictions, then we can use a small λ to put focus on the other two terms \mathcal{L}_1 and \mathcal{L}_2 in the training objective. The predictor γ has more flexibility in predicting future representations, and the model can produce predictions focusing on the upcoming time and alleviate the long-term stationarity of the series.

The estimates of ϕ , ψ , and γ can be obtained by minimizing \mathcal{L} with respect to the neural network parameters, that is,

$$\hat{\phi}, \hat{\psi}, \hat{\gamma} = \arg \min_{\mathbf{w}} \mathcal{L},$$

where $\mathbf{w} = (\mathbf{W}, \mathbf{U}, \mathbf{V}, \mathbf{M}, \mathbf{G}, \mathbf{L}, \beta)$ is the vector containing the parameters in the networks (2), (3), and (5). The objective (6) is optimized with the gradient descent method. The parameters follow the update rule

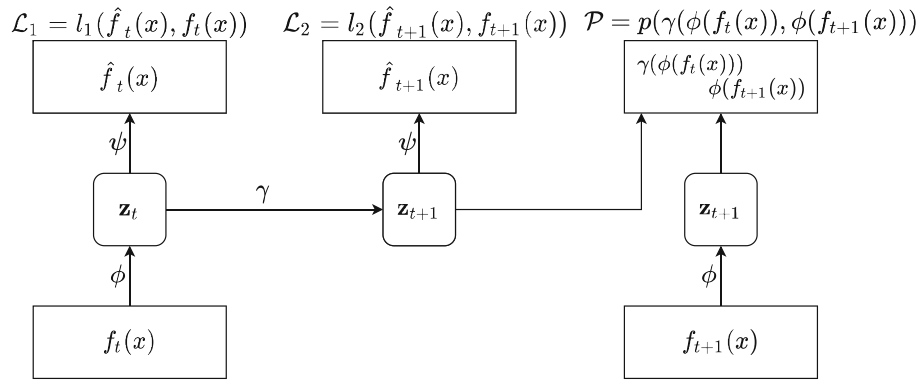


FIGURE 2 The training setup for the proposed model (1).

$$\mathbf{w} \leftarrow \mathbf{w} - \alpha \frac{d\mathcal{L}}{d\mathbf{w}},$$

for a learning rate of α . Figure 2 presents the training setup (inputs and outputs of the networks) for estimating network parameters.

For the observed functional data $\{f_t(x)\}_{t=1}^n$, we split them into a set of tuples $\{(f_t(x), f_{t+1}(x))\}_{t=1}^{n-1}$. Each tuple is then fed into the training setup for calculating the objective function and evaluating the gradient for backpropagating training errors. The training objective is designed to facilitate the optimization process with the stochastic gradient descent method. The objective (6) can be easily evaluated based on a pair of adjacent trajectories $(f_t(x), f_{t+1}(x))$. Furthermore, it is convenient to extend the proposed model into the online setting. After the model is fitted to a satisfactory degree, we can discard all observations except the last observed curve $f_n(x)$. We concatenate any new trajectory to the last observed one and update the network parameters with the calculated gradient. Then, the model can generate the predicted trajectory for the next time points.

2.3 | Generalization bound on predictions

To address the prediction ability of our proposed method, we have to investigate whether it can be generalized for any unseen data. For that matter, we could use the generalization error which corresponds to the difference between the expected loss (model errors) and the empirical loss. By providing a bound on the generalization error, we can better understand the theoretical properties of our method with respect to predictions. Detailed proof is included in the Supplementary Material.

Definition 1 (*Expected and empirical loss*). Let \mathcal{D} be the distribution of any data pair $(f_t(\cdot), f_{t+1}(\cdot))$, and q corresponds to the entire prediction model, $f_{t+1} = q(f_t) = \psi(\gamma(\phi(f_t)))$. Then, the expected loss of our prediction model with respect to \mathcal{D} is

$$l_0(q) = \mathbb{E}_{(f_t, f_{t+1}) \sim \mathcal{D}}[l(q(f_t), f_{t+1})], \quad (7)$$

and the empirical loss is

$$\hat{l}(q) = \mathbb{E}_{(f_t, f_{t+1}) \sim \mathcal{S}}[l(q(f_t), f_{t+1})], \quad (8)$$

where \mathcal{S} contains the trajectories, f_1, \dots, f_n , from a observed functional time series and l is the loss or error function.

Theorem 1 (*Generalization bound*). For any $\delta, \omega > 0$, with probability $\geq 1 - \delta$ over a set of n data pairs, the following bound holds

$$l_0(q) - \hat{l}(q) \leq \mathcal{O}\left(\sqrt{\frac{E(\mathbf{w}) + \ln(n/\delta)}{n-1}}\right)$$

$$\text{where } E(\mathbf{w}) = \frac{Hk \ln k}{\omega^2} \left(\frac{\|\mathbf{L}\|_F^2}{\|\mathbf{L}\|_F^2} + \frac{\|\mathbf{L}\|_F^2}{\|\mathbf{G}\|_F^2} + \frac{\|\mathbf{L}\|_F^2}{\|\mathbf{M}\|_F^2} + \frac{\|\mathbf{L}\|_F^2}{\|\boldsymbol{\beta}\|_F^2} + \frac{\|\mathbf{L}\|_F^2}{\|\mathbf{V}\|_F^2} + \frac{\|\mathbf{L}\|_F^2}{\|\mathbf{U}\|_F^2} + \frac{\|\mathbf{L}\|_F^2}{\|\mathbf{W}\|_F^2} \right).$$

3 | SIMULATION STUDIES

First, we conduct three simulation studies to examine the finite sample performance of the proposed NOP method. We will also compare the prediction performances of the proposed NOP method with methods proposed in Hyndman and Ullah (2007), Hyndman and Shang (2009), and Hörmann et al. (2015), and denoted here by FTSA, FPLSR, and DFPCA respectively. Three simulation settings are listed as follows:

- Setting 1: FAR(1)

$$f_t(x) = \int_a^b \Gamma(x, u) f_{t-1}(u) du + w_t(x),$$

- Setting 2: FAR(1) with a trend function

$$f_t(x) = 2 \sin\left(2\pi \frac{t}{n}\right) \sin(2\pi x) + \int_a^b \Gamma(x, u) f_{t-1}(u) du + w_t(x),$$

- Setting 3: Nonlinear Autoregressive Model

$$f_t(x) = 0.75 * \int_a^b \Gamma(x, u) f_{t-1}(u) du * \exp\left(\int_a^b \Gamma(x, u) f_{t-1}(u) du\right) + w_t(x),$$

where $\Gamma(x, u) = 0.7 \exp(-0.5(x^2 + u^2))$ and $w_t(x)$ is an independent Brownian motion over $[a, b]$ with zero mean and variance $1/n$. The simulation settings consider three scenarios: a functional autoregressive model with order of one (FAR(1)) and a trend function, a nonlinear autoregressive model, and a standard FAR(1) model. We will also consider different lengths of time series for each simulation setting to be 100, 500, 1000, and 2000. For each simulated functional time series, we will keep the last 20% trajectories in the time series as the testing sample.

In the simulation studies, we will only focus on the one-step ahead prediction \hat{f}_{n+1} by using the previous trajectories $f_1(x), \dots, f_n(x)$. The one-step ahead predictions are obtained as follows for each trajectory in the testing sample. Let n be the time index of the last observed curve, we will obtain the one-step ahead prediction \hat{f}_{n+1} and increase the training sample by 1. Then, we will repeat the prediction process until the last trajectory in the testing sample. We use the mean squared prediction errors (MSPEs) for trajectories in the test sample to compare the prediction performances of the three methods. The MSPE at a future time point t is defined as follows:

$$\text{MSPE}(t) = \frac{1}{b-a} \int_a^b [\hat{f}_t(x) - f_t(x)]^2 dx \approx \frac{1}{b-a} \sum_{j=1}^J [\hat{f}_t(x_j) - f_t(x_j)]^2.$$

Each simulation setting is replicated for 100 times. For each replication, we calculate the median of MSPEs for all the trajectories in the testing sample, which is denoted as MSPE_m in the following comparisons. The prediction performances and comparisons are summarized in the following Table 1

Table 1 summarizes the simulation results from three settings. In the simulation setting I - FAR(1) model, we could observe that all four methods demonstrate similar performances in predicting future functional data in terms of MSPE_m. Our proposed NOP method slightly outperforms others while FTSA falls behind when the time series length increases. Even with training data of 80 samples, the nonlinear mappings of our method can still be well approximated and hence produce satisfactory predictions of future trajectories.

In simulation setting II where the functional time series has a trend function, the performance of linear and static methods gets worse over larger training samples or longer time series where our proposed NOP method gives the best prediction results. Both the FPLSR and FTSA methods are based on the stationary assumption of the series and do not consider a trend function. The existing trend function violates the stationary assumption of these two methods. As the training size becomes larger, the violation gets worse hence leading to poor prediction performances. On the other hand, our method does not require the calculation of covariance functions, thus it depends less on the stationarity of the series.

TABLE 1 The mean and standard deviation of MSPE_m for one-step-ahead predictions in the testing sample over 100 replications under simulation setting I, II, and III (ordered from top to bottom).

Setting	Method	$n = 100$	$n = 500$	$n = 1000$	$n = 2000$
I	NOP	2.87 (0.58)	2.65 (0.48)	2.59 (0.51)	2.73 (0.53)
	FPLSR	2.97 (0.71)	2.75 (0.61)	2.79 (0.59)	2.98 (0.68)
	FTSA	3.23 (0.93)	3.18 (0.89)	3.04 (0.84)	3.63(1.26)
	DFPCA	2.98 (0.66)	2.79 (0.61)	2.81 (0.62)	3.01 (0.69)
II	NOP	0.99 (0.054)	0.98 (0.054)	0.99 (0.55)	0.98 (0.052)
	FPLSR	2.64 (0.29)	6.63(1.30)	11.38(1.97)	22.22(4.01)
	FTSA	2.53 (0.33)	5.07(1.26)	7.38(2.21)	11.71(3.91)
	DFPCA	1.02 (0.056)	1.02 (0.056)	1.02 (0.058)	1.23 (0.054)
III	NOP	0.99 (0.058)	0.99(0052)	0.98 (0.054)	0.98 (0.052)
	FPLSR	1.98 (0.084)	1.98 (0.081)	1.99 (0.10)	1.99 (0.10)
	FTSA	1.99 (0.093)	1.99 (0.082)	1.99 (0.11)	1.99 (0.093)
	DFPCA	1.00 (0.057)	1.00 (0.052)	0.99 (0.054)	1.01 (0.051)

Hence, we can see that the NOP method provides the most consistent and best predictions regardless of the size of the training sample or the structure of the functional time series. The DFPCA method is slightly worse than our method but much better than the method using static functional principal components.

In simulation setting III where the dependence between functions is nonlinear, the proposed method slightly outperforms the DFPCA method while the other two linear approaches FPLSR and FTSA become significantly worse compared to the case of simulation setting I. In summary, we can see that our NOP method is the most consistent method and always produces the best predictions regardless of the structure of the functional time series.

4 | APPLICATIONS

In this article, we consider three applications of predicting air quality, electricity price, and mortality rate. The air quality data are composed of multivariate functions over time. The other two applications are univariate functional time series. The electricity price data resembles a stationary functional time series. The mortality rate appears to have a decreasing trend function over the years. For all three applications, we employ the proposed NOP method and compare its prediction performances with the FTSA and FPLR methods proposed in Hyndman and Ullah (2007) and Hyndman and Shang (2009), respectively. The detailed results are summarized in the following subsections.

4.1 | Air quality

This application focus on a multivariate functional time series of the air quality. Each multivariate trajectory contains hourly recordings of particle density of three chemicals NO₂, O₃, and SO₃. In this application, we collect data in 2015–2020 from the BC air quality data archive (<https://www2.gov.bc.ca/gov/content/environment/air-land-water/air/air-quality/current-air-quality-data/bc-air-data-archive>). Figure 3 displays trajectories of the functional time series in the first 100 days. Figure 3 shows that the hourly recordings of particle density are quite different for each chemical. The density of NO₂ appears to have multiple peaks throughout the day, and the trend of trajectories is cyclic. The overall density seems to increase from Day 1 and starts to fall off on Day 5. The density of O₃ peaks in the afternoon and may rise around midnight. SO₃ density curves are more consistent, where the majority of them have a peak around noon and diminish to zero afterward.

Both the FPLSR and FTSA methods cannot be directly applied to multivariate functions. The standard technique is concatenating the multivariate functions into a univariate one, thus the resulting functional time series will be univariate.

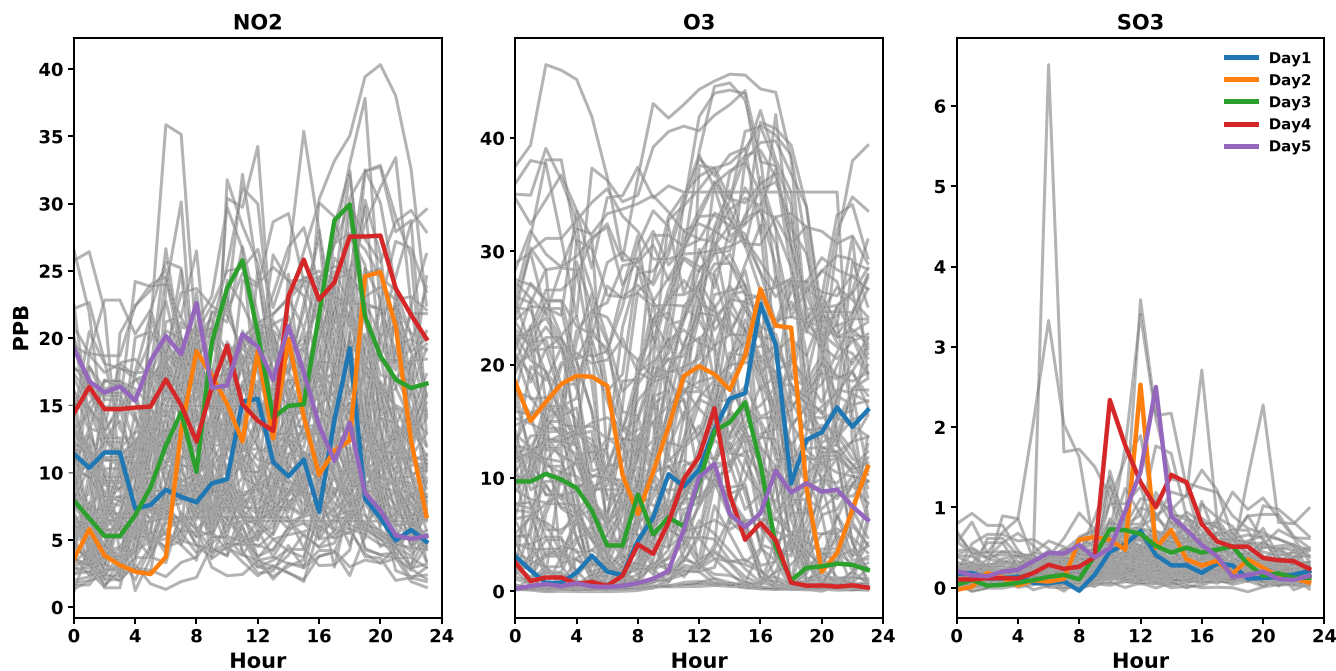


FIGURE 3 Hourly recordings of particle density of three chemicals NO_2 , O_3 , and SO_3 in the first 100 days in 2015. Gray curves are individual trajectories. Highlighted curves are from five consecutive days to show the evolution of functional time series. Parts per billion (PPB) is the number of units of mass of a contaminant per 1 billion units of total mass.

The NOP method can directly take multivariate inputs without any modification. We will keep the remaining 20% of the series as the testing sample,

Figure 4 illustrates the one-step-ahead predictions for the first 3 days in the testing sample. One disadvantage of concatenating multivariate functions is the difficulty in predicting trajectories around connecting points. Figure 4 shows that FPCA-based approaches produce worse predictions at the junction of curves, for example, 0 and 24 h from O_3 curves. The NOP method does not have such disadvantages, and it can consistently predict future trajectories for all chemicals. In addition, when we concatenate multivariate functions into univariate ones, the temporal dependence becomes more complicated to capture and may not follow a linear relationship. The predictions from the NOP method are significantly better than linear models.

We obtain the one-step-ahead prediction and calculate the MSPE for each trajectory in the testing sample. Given the case of multivariate functions, we will compare the MSPE with respect to each dimension after obtaining predictions. The boxplots of MSPEs in the testing sample are summarized in Figure 5. Figure 5 shows that the MSPEs from the NOP method are the smallest compared to the other two methods. This conclusion is consistent across all three chemicals. It is evident that our NOP method excels in dealing with multivariate functional data. The proposed NOP method is also easier to be applied to real data sets because the NOP method does not need to calculate covariance functions.

4.2 | Electricity price

We have the hourly electricity price of northern Italy from January 6th, 2015 to September 20th, 2020. The entire functional time series contains 2096 trajectories. Figure 6 displays some sample trajectories from the data set.

The prediction performance is evaluated based on the predictions for the last 50 days. For the proposed NOP method, we will consider two prediction models. The first one ignores the difference between weekdays and weekends, that is, we define the prediction $\mathbf{z}_{t+1} = \gamma(\mathbf{z}_t)$ to depend only on the previous latent representation. This model is denoted as NOP in the following comparisons. The second model considers the weekend/weekday effect by adding an indicator g_{t+1} , where $g_{t+1} = 1$ if $t + 1$ th day is a weekday and zero otherwise. This model will be referred to as NOP_weekday in the following sections. The predicted representation from NOP_weekday is obtained by $\mathbf{z}_{t+1} = \gamma(\mathbf{z}_t, g_{t+1})$.

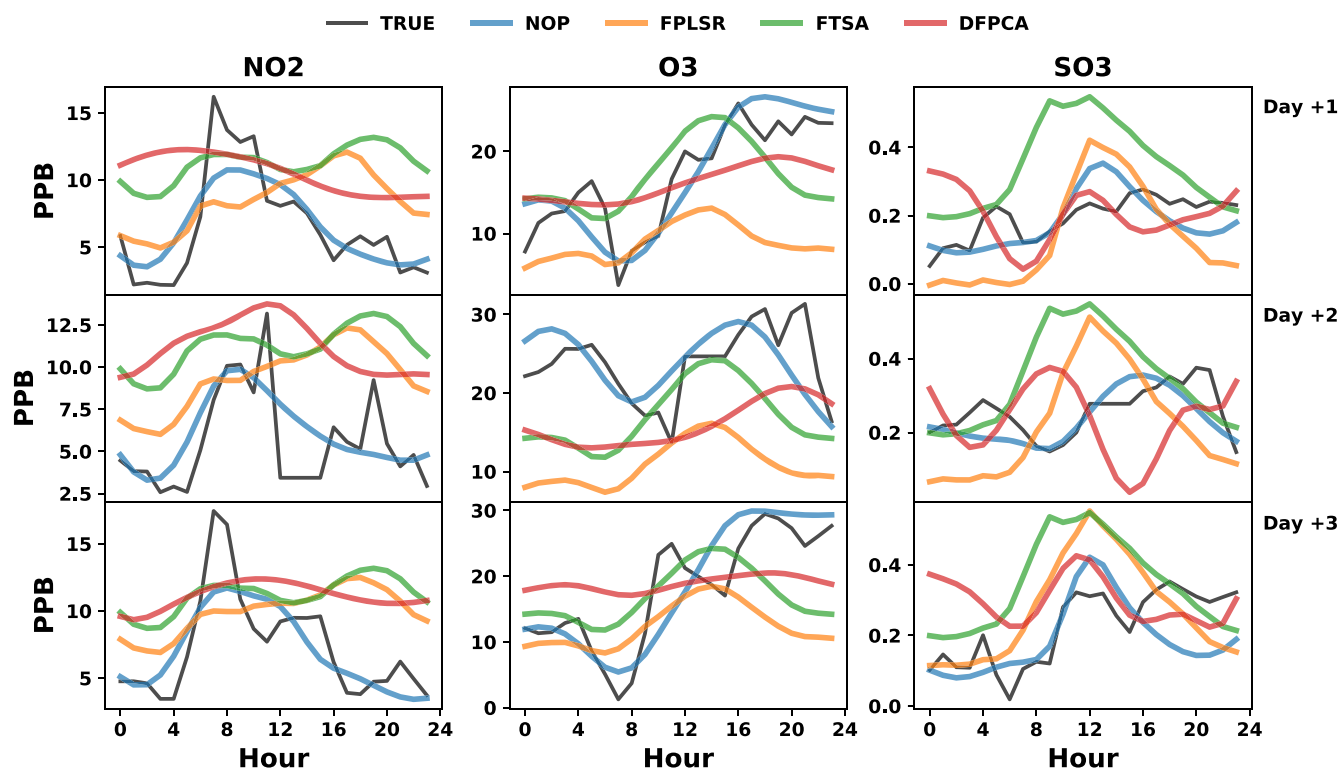


FIGURE 4 One-step ahead predictions for the first 3 days in the testing sample. The black curves are the true observations. The blue, orange, and green curves are the predicted trajectories from the proposed nonlinear prediction model (NOP), the functional partial least square method (FPLSR), the functional time series analysis (FTSA), and the dynamic functional principal component (DFPCA) method respectively. The chemicals, NO₂, O₃, and SO₂, are ordered from left to right. The prediction for Day+1, Day+2, and Day+3 are ordered from top to bottom. Parts per billion (PPB) is the number of units of mass of a contaminant per 1 billion units of total mass.

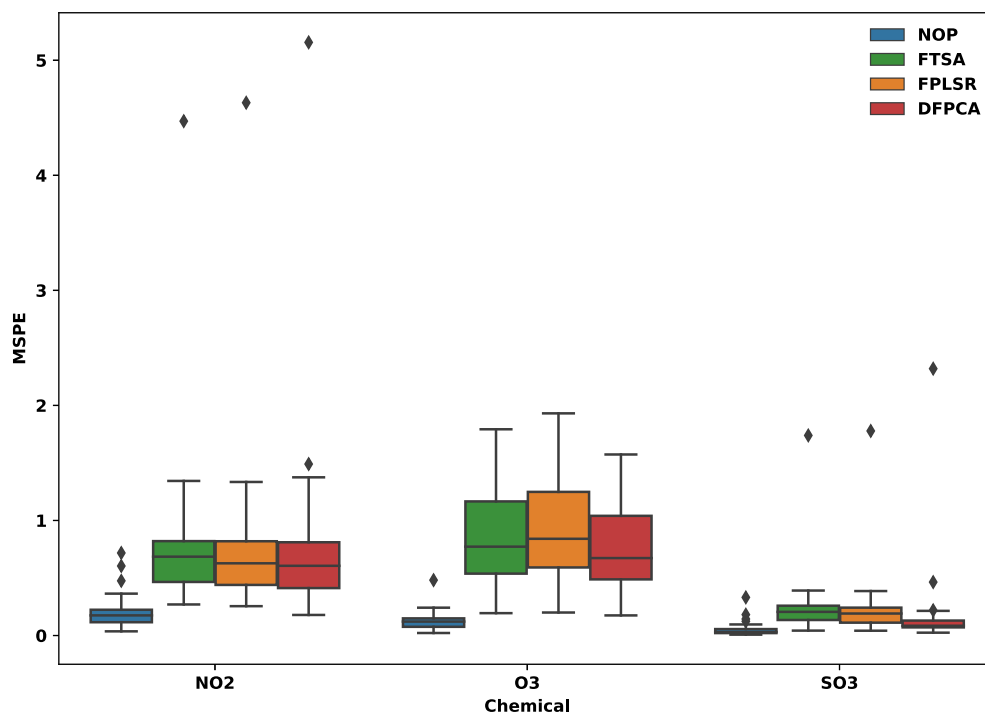


FIGURE 5 The boxplots of mean squared prediction errors (MSPEs) of one-step-ahead predictions in the testing sample. The blue, orange, and green curves are the predicted trajectories from the proposed nonlinear prediction model (NOP), the functional partial least square method (FPLSR), the functional time series analysis (FTSA), and the dynamic functional principal component (DFPCA) method respectively.

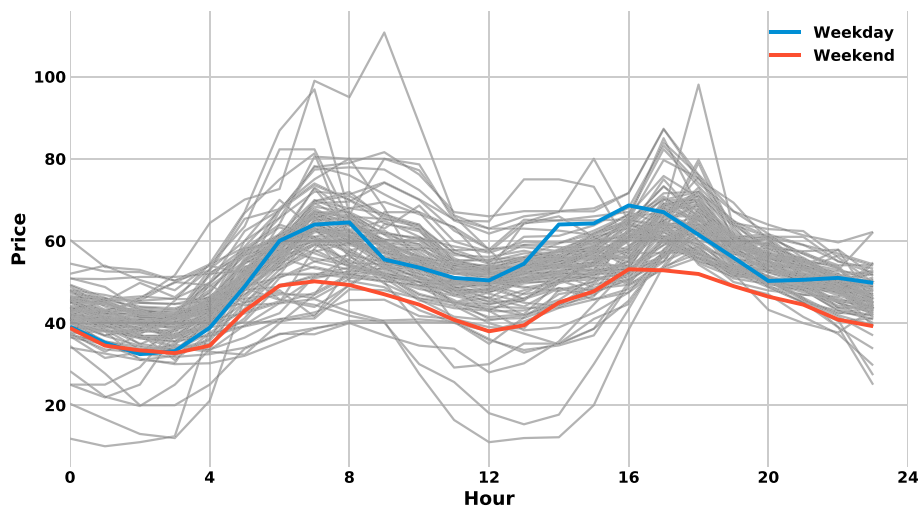


FIGURE 6 Hourly electricity prices recorded for northern Italy. We highlight 2 trajectories for showcasing the data on weekends and weekdays. The blue and red curve corresponds to the price on one weekday and one weekend, respectively.

Figure 7 displays the predicted hourly price for the following week by three methods. The plus sign “+” indicates the number of days to predict after the last observation. Because Day+3 and Day+4 are Saturday and Sunday, the NOP method including the weekend effect provides significantly better predictions for electricity prices in comparison with the FPLSR and FTSA methods and the NOP model without weekend effects. The future and long-term predictions are obtained simultaneously. The proposed NOP model with weekend effects can adjust predictions when encountering weekends, hence it produces stable and better forecasts after weekends as shown for Day+5 and Day+6 prediction curves in Figure 7.

We compare the MSPE for future predictions in the following 20 days. The prediction results are summarized in Figure 8. Figure 8 shows that the proposed NOP model with the weekend effect outperforms other methods when predicting prices for a longer period in advance. The reason is that the predictions are made iteratively rather than simultaneously. Weekend effects (or other possible extrinsic effects) offer more information in comparison with only using the previously predicted trajectory. Including the weekend effect forces the prediction to be more reliable and close to the true observation. This will also help in predicting subsequent trajectories by having some stable points in the iterative prediction process. We make this conclusion because the MSPE using the NOP model with the weekend effect is the most stable in comparison with other methods. In general, we observe that our proposed NOP method consistently has smaller MSPEs than those methods based on FPCA.

Figure 9 compares MSPEs of predicted daily electricity prices for the following 30 days given different values of penalty parameter λ . Smaller λ 's give better predictions in the short term, that is, the first few days after the last observed trajectories. However, the predictions can be unstable (the orange curve) given that we are not enforcing the importance of the established latent space. That is, we allow the predicted representations to be too flexible yet do not consider how it is supposed to be in the current latent space. As λ increases, the prediction errors are more stable in the long run. Larger λ implies that the future trajectories are more similar to the observed ones, and it suggests that the established latent space is well established to include unseen curves. On the other hand, smaller λ prioritizes predicting future curves and does not enforce the importance of the latent feature space. In other words, we can use larger λ 's to adhere to the stationary assumption of the functional time series. We can use smaller λ 's to focus on short-term predictions.

4.3 | Mortality rate

The left panel in Figure 10 displays the log mortality rate curves in France in 1916–2016. The mortality rate data appear to have a decreasing trend over the years. That is, the most recent year has the lowest rate compared to previous years. This functional data is similar to Setting 1 in the simulation studies where the functional time series is nonstationary. We

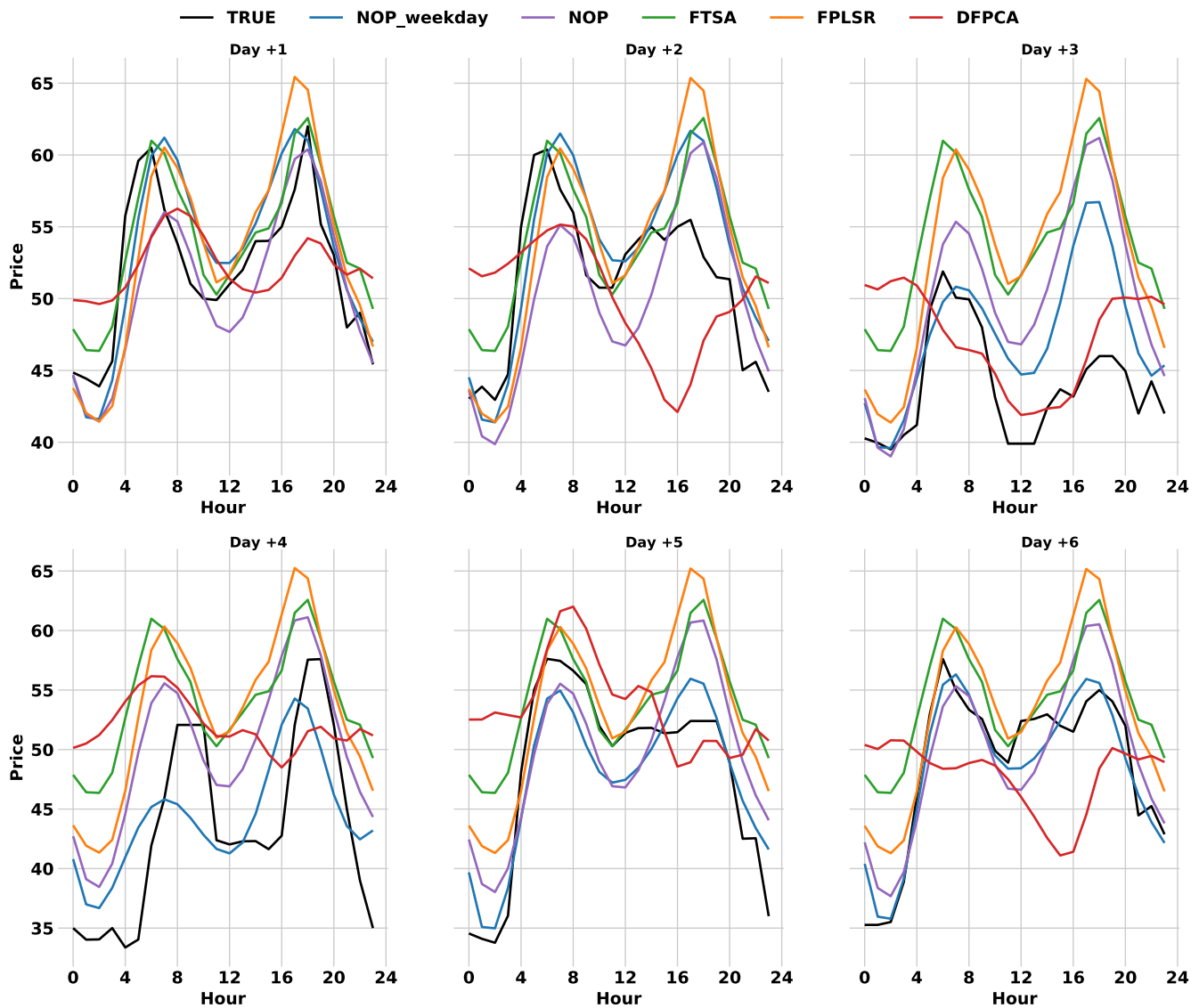


FIGURE 7 The predicted hourly electricity price in northern Italy for the following week. The black curve is the true observation. The blue and purple curves are the prediction from the nonlinear prediction method (NOP) with and without the weekend effect. The green, orange, and red curves are predictions from the functional partial least square method (FPLSR), the functional time series analysis (FTSA), and the dynamic functional principal component (DFPCA) method respectively.

keep the trajectories in 1996–2016 as the testing sample, and the prediction performance is investigated for this period. We predict the mortality rate in the last 20 years from 1996 to 2016 simultaneously with the NOP method. The right panel in Figure 10 shows that the predicted trajectories are decreasing over the years, indicating that our NOP method can detect the decreasing trend in the functional time series.

Then, we compare the prediction performance of the NOP method with that of both FTSA and FPLSR methods. As illustrated in the simulation studies when the functional time series has a trend function, both FTSA and FPLSR method are not satisfactory in comparison to the NOP method. We expect similar conclusions in this application. Figure 11 summarizes the MSPEs for mortality rate curves in 1996–2016. Because both FTSA and FPLSR methods require the functional time series to be stationary, the long-term predictions from these two methods are not satisfactory. Moreover, the prediction becomes worse quickly when predicting curves that are farther away from the last observed curve. On the other hand, the MSPE of our method is steady regardless of the gap between a future trajectory and the last observed one. This demonstrates that the NOP method can capture the trend even without explicitly specifying such temporal relationships.

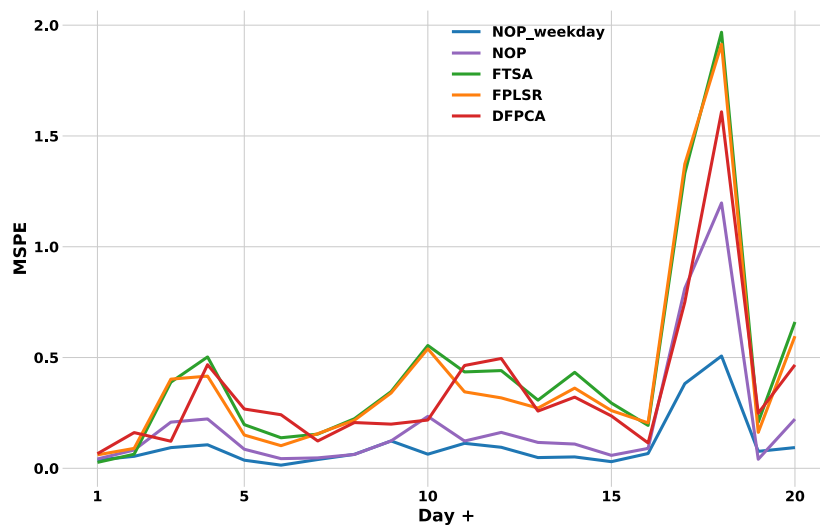


FIGURE 8 Mean squared prediction errors (MSPEs) of five methods for the following 20 days after the last observation. The blue and purple curves are the prediction from the nonlinear prediction method (NOP) with and without the weekend effect. The green, orange, and red curves are predictions from the functional partial least square method (FPLSR), the functional time series analysis (FTSA), and the dynamic functional principal component (DFPCA) method respectively.

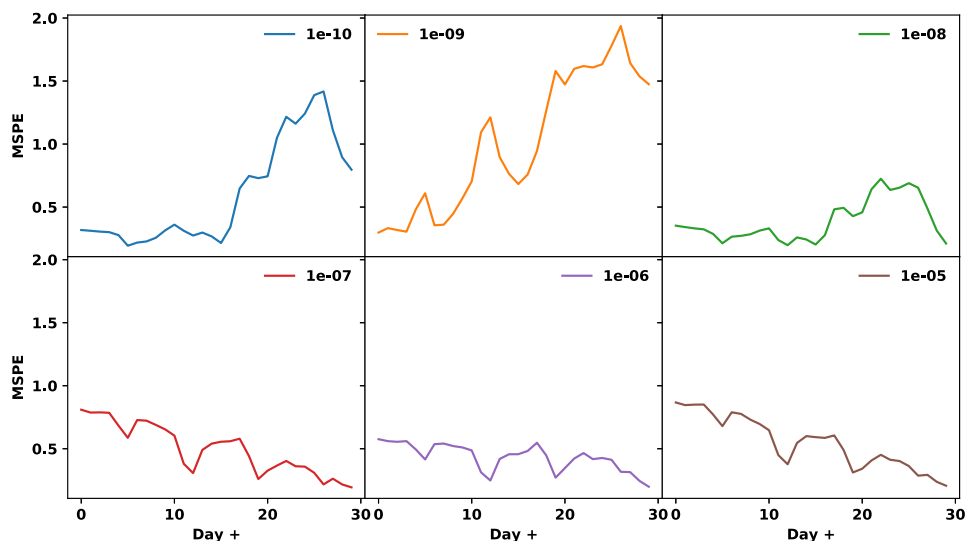


FIGURE 9 The mean squared prediction error (MSPE) of predicted daily electricity prices for the following 30 days when the smoothing parameter $\lambda = 10^{-10}, 10^{-9}, 10^{-8}, 10^{-7}, 10^{-6}$, and 10^{-5} .

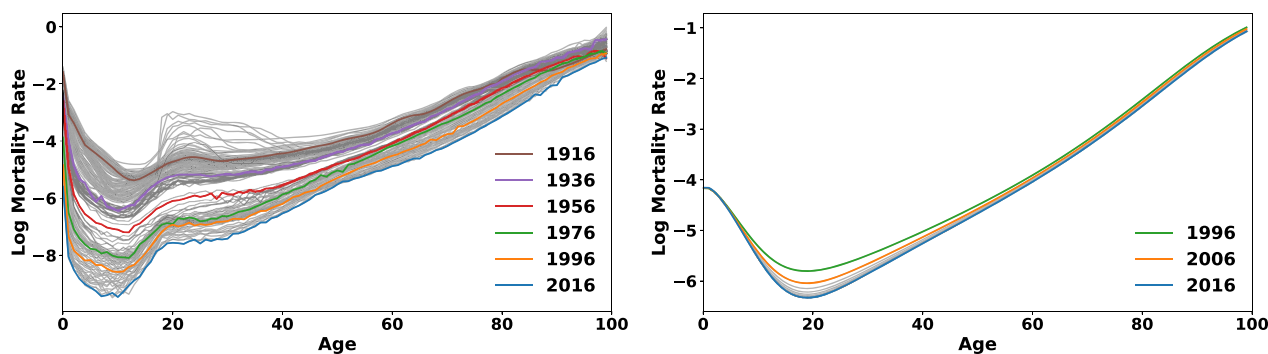


FIGURE 10 The left panel shows the log mortality rate in France from 1916 to 2016. Six curves are highlighted to show the decreasing trend in the rate. The right panel demonstrates the predicted mortality curves in the last 20 years from 1996 to 2016.

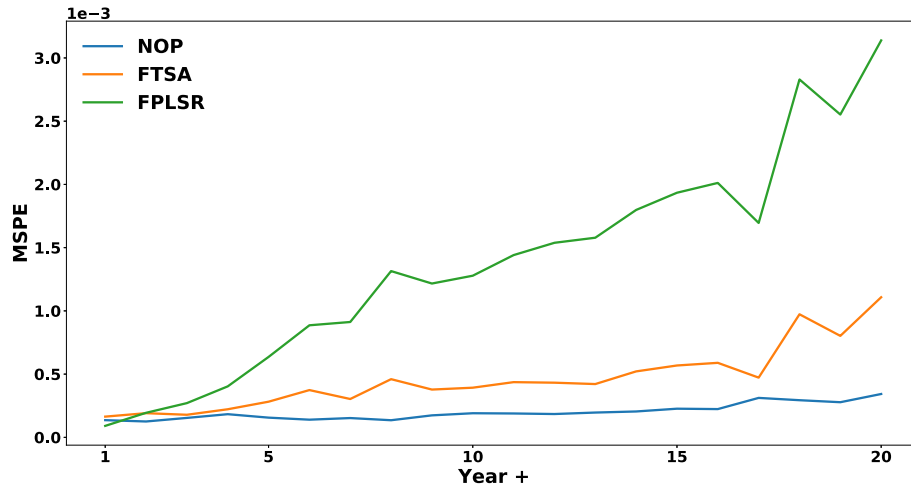


FIGURE 11 Mean squared prediction errors (MSPEs) for the mortality rate in 1996–2016. The blue, red and green curves are the MSPEs for using the proposed nonlinear prediction model (NOP), the functional time series analysis (FTSA), and the functional partial least square method (FPLSR) respectively.

5 | CONCLUSION

In this article, we propose a nonlinear prediction (NOP) method for functional time series. The model components are nonlinear functions that are flexible for analyzing real data sets. Instead of using the linear latent space spanned by FPCs, we establish a more useful latent space for compressing observed functional data. This space provides a better ground for predicting future trajectories because it is constructed with forecast information. Moreover, the NOP method can predict multivariate functional time series much easier than the conventional methods. At the same time, the NOP method includes extrinsic effects to produce better long-term predictions. The NOP method can also be implemented with online estimation and prediction since it does not require using all observations but the last observed trajectory. We demonstrate the advantages of the NOP method over FPCA-based ones with simulation studies and three applications.

ORCID

Jiguo Cao  <https://orcid.org/0000-0001-7417-6330>

REFERENCES

- Alvarez Lieba, J. (2017). Functional time series: a review and comparative study. preprint, arXiv:1706/06288.
- Aue, A., Norinho, D., & Hormann, S. (2015). On the prediction of stationary functional time series. *Journal of the American Statistical Association*, 110(509), 378–392.
- Berrendero, J. R., Justel, A., & Svarc, M. (2011). Principal components for multivariate functional data. *Computational Statistics and Data Analysis*, 55, 2619–2634.
- Bosq, D. (2000). *Linear Processes in Function Spaces: Theory And Applications*. Springer.
- Chen, K., & Lei, J. (2015). Localized functional principal component analysis. *Journal of the American Statistical Association*, 110(511), 1266–1275.
- Chiou, J., Chen, Y., & Yang, Y. (2014). Multivariate functional principal component analysis: a normalization approach. *Statistica Sinica*, 24, 1571–1597.
- Dai, X., & Müller, H. (2018). Principal component analysis for functional data on riemannian manifolds and spheres. *The Annals of Statistics*, 46, 3334–3361.
- Dauxois, J., Pousse, A., & Romain, Y. (1982). Asymptotic theory for the principal component analysis of a vector random function: Some applications to statistical inference. *Journal of Multivariate Analysis*, 12, 136–154.
- Elayouty, A., Scott, M., & Miller, C. (2022). Time-varying functional principal components for non-stationary epcO₂ in freshwater systems. *Journal of Agricultural, Biological and Environmental Statistics*, 27, 506–522.
- Elias, A., Jimenez, R., & Shang, H. (2022). On projection methods for functional time series forecasting. *Journal of Multivariate Analysis*, 189(104890), 1–20.
- Fan, Y., James, G., & Radchenko, P. (2015). Functional additive regression. *The Annals of Statistics*, 43, 2296–2325.

- Gao, Y., Shang, H., & Yang, Y. (2019). High-dimensional functional time series forecasting: An application to age-specific mortality rates. *Journal of Multivariate Analysis*, 170, 232–243.
- Guo, W. (2002). Functional mixed effects models. *Biometrics*, 58, 121–128.
- Hall, P., & Hosseini-Nasab, M. (2005). On properties of functional principal components analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68, 109–126.
- Hall, P., Müller, H.-G., & Wang, J.-L. (2006). Properties of principal component methods for functional and longitudinal data analysis. *The Annals of Statistics*, 34, 1493–1517.
- Happ, C., & Greven, S. (2018). Multivariate functional principal component analysis for data observed on different (dimensional) domains. *Journal of the American Statistical Association*, 113, 649–659.
- Horvath, L., Kokoszka, P., & Reeder, R. (2013). Estimation of the mean of functional time series and a two-sample problem. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 75, 103–122.
- Horvath, L., Kokoszka, P., & Rice, G. (2014). Testing stationarity of functional time series. *Journal of Econometrics*, 179, 66–82.
- Hyndman, R., & Shang, H. L. (2009). Forecasting functional time series. *Journal of the Korean Statistical Society*, 38(3), 199–211.
- Hyndman, R., & Ullah, S. (2007). Robust forecasting of mortality and fertility rates: A functional data approach. *Computational Statistics and Data Analysis*, 51(10), 4942–4956.
- Hörmann, S., Kidziński, L., & Hallin, M. (2015). Dynamic functional principal components. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 77(2), 319–348.
- Hörmann, S., & Kokoszka, P. (2010). Weakly dependent functional data. *The Annals of Statistics*, 38, 1845–1884.
- James, G., Hastie, T., & Sugar, C. (2000). Principal component models for sparse functional data. *Biometrika*, 87(3), 587–602.
- Jiao, S., Aue, A., & Ombao, H. (2019). Functional time series prediction under partial observation of the future curve. *Journal of the American Statistical Association*, 5(1), 1–12.
- Kargin, V., & Onatski, A. (2008). Curve forecasting by functional autoregression. *Journal of Multivariate Analysis*, 99, 2508–2526.
- Kong, D., Bondell, H., & Shen, W. (2018). *Outlier detection and robust estimation in nonparametric regression*. In *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics* (Vol. 84, pp. 208–216). PMLR.
- Lin, Z., Wang, L., & Cao, J. (2015). Interpretable functional principal component analysis. *Biometrics*, 72, 846–854.
- Morris, J. S. (2015). Functional regression. *Annual Review of Statistics and Its Application*, 2, 321–359.
- Müller, H.-G., & Stadtmüller, U. (2005). Generalized functional linear models. *The Annals of Statistics*, 33, 774–805.
- Müller, H. (2008). Functional additive models. *Journal of the American Statistical Association*, 103, 1534–1544.
- Nie, Y., & Cao, J. (2020). Sparse functional principal component analysis in a new regression framework. *Computational Statistics and Data Analysis*, 152, 107016.
- Nie, Y., Wang, L., Liu, B., & Cao, J. (2018). Supervised functional principal component analysis. *Statistics and Computing*, 28, 713–723.
- Nie, Y., Yang, Y., Wang, L., & Cao, J. (2022). Recovering the underlying trajectory from sparse and irregular longitudinal data. *Canadian Journal of Statistics*, 50, 122–141.
- Ramsay, J., & Silverman, B. (2005). *Functional Data Analysis*. Springer.
- Sang, P., Wang, L., & Cao, J. (2017). Parametric functional principal component analysis. *Biometrics*, 73, 802–810.
- Scheipl, F., Staicu, A.-M., & Greven, S. (2013). Functional additive mixed models. *Journal of Computational and Graphical Statistics*, 24, 477–501.
- Shi, H., & Cao, J. (2022). Robust functional principal component analysis based on a new regression framework. *Journal of Agricultural, Biological and Environmental Statistics*, 27, 523–543.
- Shi, H., Dong, J., Wang, L., & Cao, J. (2021). Functional principal component analysis for longitudinal data with informative dropout. *Statistics in Medicine*, 40, 712–724.
- Vieu, P., & Ferraty, F. (2006). *Nonparametric Functional Data Analysis: Theory and Practice*. Springer.
- Yao, F., Müller, H.-G., & Wang, J.-L. (2005). Functional linear regression analysis for longitudinal data. *The Annals of Statistics*, 33, 2873–2903.

SUPPORTING INFORMATION

Additional supporting information can be found online in the Supporting Information section at the end of this article.

How to cite this article: Wang, H., & Cao, J. (2023). Nonlinear prediction of functional time series. *Environmetrics*, 34(5), e2792. <https://doi.org/10.1002/env.2792>