Partial Functioal Linear Model Based on RKHS and Ensemble Learning

Mengyao Bian [1, #], Yaqi Pan [2, #], Zhangru Yin [3, *, #]

[1] School of Mathematics and Applied Mathematics, Taizhou University, Taizhou, China, 317000

[2] School of Mathematics and Statistics, Hunan University of Finance and Economics, Changsha, China, 410205

[3] School of Applied Statistics, Zhejiang Agriculture and Forestry University, Hangzhou, China, 311300

* Corresponding Author Email: 19012767228@163.com

[#]These authors contributed equally.

Abstract. A new partial functional linear model is proposed for functional and vector-valued covariates with a scalar response. The proposed model assumes a linear relationship between the functional predictors and the scalar response, which is approximately estimated using a functional principal component basis expansion. For the vector-valued covariates, the flexibility of the model is enhanced by assuming that the connection function with the scalar response resides in a Reproducing Kernel Hilbert Space (RKHS), enabling its representation through a kernel function expansion. Furthermore, instead of relying on model selection, a ensemble learning approach is employed. By constructing partial functional linear models with different truncation numbers and performing a weighted average, the issue of truncation number selection is effectively addressed. Simulation studies and real data analysis demonstrate that the proposed method exhibits superior performance and competitiveness compared to the benchmark models.

Keywords: Functional Data Regression, Regeneration Kernel Approach, Ensemble Learning.

1. Introduction

The continuous advancement of data acquisition and storage technologies, exemplified by sensor networks that enable dynamic and continuous monitoring of fluctuations in various physical quantities, has led to the emergence of data with increasingly complex structures. This type of data not only captures function values at discrete points but also encompasses a variety of function-related characteristics, such as geometric shape, smoothness, and periodic properties. As a result, a vast amount of functional data has been generated. In the context of functional data analysis, functional data regression plays a crucial role and is widely applied across fields such as biomedicine, environmental science, and financial economics [1-2].

In practical problem-solving scenarios, scalar predictor variables that can also influence the target are often present alongside functional predictor variables. In such cases, the partial functional linear model is of particular interest. Several scholars have contributed to the development of this model. For instance, Sun et al. [3] combined the existing functional data principal component analysis model with traditional time-series analysis methods and interval principal component analysis techniques. Zhang et al. [4] used traditional spectral decomposition methods to represent the covariance function, which made the functional part of the partial functional linear model linearizable. Yuan et al. [5] integrated functional and scalar data, utilizing a combination of B-spline smoothing and functional smoothing techniques to perform local sparse estimation of the slope function, while also applying the SCAD method to estimate scalar coefficients. Chen and Ling [6] employed the functional linear model to describe the relationship between functional predictors and scalar responses, reconstructing missing data samples using functional principal component analysis. Yan [7] explored the regression problem of functional response variables with covariates, as well as functional meta-analysis and the theoretical characteristics of functional principal components in a general weighted framework. Yang

and Li [8] examined the endogenous relationship between the response variable and the functional explanatory variable in a Hilbert space, using the Karhunen-Loève theorem to study the Bayesian estimation method of a spatial functional autoregressive model. Zhang and Fan [9] proposed a partial linear model with kernelized functions and convergence conditions based on the reproducing kernel of the Reproducing Kernel Hilbert Space, integrating the kernel trick with the Gauss-Newton iterative algorithm. Li et al. [10] innovatively introduced a generalized partial functional linear model. However, a limitation common to these studies is the assumption that the scalar parts are always linear. Additionally, in many papers, the models for functional data are truncated using principal components, and the issue of determining the number of truncations in the functional principal component expansion remains unresolved.

In response to these limitations, this paper proposes a new partial functional linear model. The key innovation of the model is that it assumes a linear relationship between the functional random variable and the scalar response, which is then estimated using a functional principal component basis expansion. Furthermore, for finite-dimensional Euclidean covariates, the model assumes that the connection function between the covariates and the scalar response resides in a Reproducing Kernel Hilbert Space (RKHS), with the connection function being represented through kernel function expansion. This significantly enhances the flexibility of the model, enabling it to better accommodate complex and variable data scenarios. Moreover, the concept of ensemble learning is introduced: partial functional linear models with different truncation numbers are constructed, weighted, and averaged, effectively addressing the critical issue of truncation number selection. This approach ensures a more scientifically grounded model construction process and improves the adaptability and stability of the model in practical applications. The results of data analysis demonstrate that the proposed method outperforms comparison methods in terms of prediction accuracy and model applicability.

2. Theory and Methodology

2.1. Functional Principal Component Analysis

The goal of functional principal component analysis is to find a set of orthogonal basis functions $\phi_n(t)$ and corresponding eigenvalues λ_n such that the projection of a random function in the directions of these basis functions can maximize the variance. The set of eigen-functions and eigen-vectors are precisely the eigenvalues λ_n and the corresponding eigenfunctions $\phi_n(t)$ obtained from the eigenvalue decomposition of the covariance function $C(s,t)$, which is specifically expressed as follows [11]:

$$C(s,t) = \sum_{n=1}^{\infty} \lambda_n \phi_n(s) \phi_n(t) \tag{1}$$

Where $\phi_n(t)$ is the nth principal component basis function. Based on this sequence of principal component basis functions, the random function $X(t)$ can be expressed as:

$$X(t) = \sum_{n=1}^{\infty} a_n \phi_n(t) \tag{2}$$

Where a_n is the corresponding coefficient of the basis function. In practical applications, we usually retain the first few principal components through the cumulative contribution rate because they can explain most of the data variations. However, in regression prediction tasks, a single principal component truncation scheme may result in the loss of some effective prediction information.

2.2. Reproducing Kernel Theory

The reproducing kernel theory is an important concept in functional analysis. The core of this theory is the reproducing kernel Hilbert space (RKHS). Suppose that H is a reproducing kernel Hilbert space (RKHS) [12] and K(s,t) is its reproducing kernel. For any f, g∈H, we have:

$$\langle f, g \rangle = \int\int f(s)g(t)K(s,t)dsdt \tag{3}$$

Where $\langle \cdot, \cdot \rangle$ represents the inner product. An important property of the reproducing kernel is the reproducing property:

$$f(s) = \langle f, K(s,\cdot) \rangle \tag{4}$$

This means that any function in the space can be represented by the inner product of the reproducing kernel with itself. The application of the reproducing kernel theory is not limited to the field of machine learning. It also plays an important role in many other fields, such as signal processing, image analysis, and control theory. By leveraging the properties and construction methods of reproducing kernels, we can design more flexible and powerful algorithms to solve practical problems.

2.3. Partial Functional Linear Model Based on RKHS and Ensemble Learning

The partial functional linear model is a statistical method that combines functional data analysis and linear models[13]. Suppose the relationship between the response variable Y, the functional predictor variable $X(t)$, and a vecto -valued predictor variable Z is expressed as follows:

$$Y = \int_{t\in \mathrm{T}} X(t)\beta(t)dt + g(Z) + \varepsilon \tag{5}$$

Where $\beta(t)$ is the functional coefficient to be estimated, $g(Z)$ is the linear effect of the scalar predictor variable, and ϵ is the error term that follows a standard normal distribution. Next, we expand $X(t)$ and $\beta(t)$ using the principal component basis functions respectively, as shown below:

$$\beta(t) = \sum_{n=1}^{n} b_n \phi_n(t)$$

$$X(t) = \sum_{i=1}^{n} a_n \phi_n(t)$$

Let $a = (a_1, a_2, ...a_n)^T$, $b = (b_1, b_2, ...b_n)^T$,then the above model can be redefined as follows:

$$Y = a^T b + g(Z) + \varepsilon \tag{6}$$

Where $g(z)$ belongs to the reproducing kernel Hilbert space and can be written in the form of a linear combination of kernel functions:

$$g(z) = \sum_{i=1}^{n} \partial_i k(z_i, z) = k^T \partial \tag{7}$$

Where $k = (k(z_1, z), k(z_2, z)...k(z_n, z))^T$ and $\partial = (\partial_1, \partial_2...\partial_n)^T$. Substituting $g(z)$ into the original equation, we get Y as:

$$Y = a^T b + k^T \partial + \varepsilon \tag{8}$$

Combining the combinations:

$$Y = (a, k)^T (b, \partial)^T + \varepsilon \tag{9}$$

Based on penalized least squares, the coefficient vector estimate can be obtained. For new test data, based on the basis expansion coefficients and coefficient vector estimates, the corresponding prediction value can finally be obtained.

Next, for different numbers of basis function stages, corresponding predicted values of the test data can be obtained. That is, for the n samples in the training set, we can get a sequence of predicted values under M different sequential truncation numbers. Then, the partial linear models with different numbers of basis function expansions are regarded as sub - models, and a weighted linear model is used for integration.Specifically, in the training process of each sub - model, we use the k-fold cross - validation method. The training data set$D_{train}$is divided into k folds, with k-1folds as the training set and the remaining 1 fold as the validation set. For example, for the basis model j after repeating the process k times, the predicted values of D_{train} under the basis model j can be obtained.For the test data D_{test}, the predicted value $\hat{y}^{j}_{ik_*}$of sample i under the basis model j is:

$$\frac{\sum_{k=1}^{k} \hat{y}_{ik_*}^{j}}{k}$$

Record the predicted results of D_{train} under M basis models and combine them with the output results of the training set to form a new training data set D_{new}, which is used to fit the weighted linear model.

$$D_{new} = \left\{\left(\left(\hat{y}_1^{(1)}, \hat{y}_1^{(2)} \ldots . \hat{y}_1^{(m)}\right), y_1\right), \left(\left(\hat{y}_2^{(1)}, \hat{y}_2^{(2)} \ldots . \hat{y}_2^{(m)}\right), y_2\right) \ldots . \left(\left(\hat{y}_n^{(1)}, \hat{y}_n^{(2)} \ldots . \hat{y}_n^{(m)}\right), y_n\right)\right\}$$

In addition, each cross - validation produces a prediction vector for the test dataset D_{test} Since the entire process of traversing the training dataset D_{train}involves k experiments, we use the simple arithmetic mean for the first - layer output of the test data D_{test}Next, we select a suitable weighted linear model, such as linear regression, random forest regression, or Lasso regression. Here, we take linear regression as an example. We use D_{new}as the training data to fit the following linear regression model:

$$y_i = \beta_o + \beta_1 \hat{y}_i{}^{(1)} + \beta_2 \hat{y}_i{}^{(2)} + \beta_3 \hat{y}_i{}^{(3)} + \ldots . \beta_n \hat{y}_i{}^{(n)} + \epsilon_i \quad (10)$$

Where β_o is the intercept, $\beta_1, \beta_2, \ldots . \beta_n$, is the linear regression coefficient,$\epsilon_i$is the error term, $\epsilon_i \sim N(0, \sigma^2{}_i)$, where $\hat{y}_i{}^{(j)}$ is the prediction value of the ith sample under the jth kernel function, and$y_i$is the actual target value of sample i. We use the least - squares method to minimize the absolute error function to obtain the optimal estimate of the regression coefficient vector $\hat{\beta} = (Z^T Z)^{-1} Z^T Y$, where Z is the feature matrix:

$$\begin{bmatrix} 1 & \hat{y}_1{}^{(1)} & \cdots & \hat{y}_1{}^{(n)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \hat{y}_n{}^{(1)} & \cdots & \hat{y}_n{}^{(n)} \end{bmatrix}$$

Where Y is the column vector of output data. Taking the simple arithmetic mean output result $(\hat{y}_i^{(1)}, \hat{y}_i^{(2)}, \hat{y}_i^{(3)}, \ldots . \hat{y}_i^{(m)})$ of D_{train} at the first layer as the input features of the above - trained linear regression model, We can obtain the corresponding predicted output results for the test data, which are specifically as follows:

$$\hat{y}_* = \sum_{j=1}^{m} \hat{y}_i^{(j)} \hat{\beta}_j + \beta_0 \quad (11)$$

This integrated learning model, if the single thread running, is relatively time-consuming, we recommend parallel computing or distributed computing to optimize the computing time.

3. Data Analysis

3.1. Simulation studies

To test the predictive performance of the proposed method, simulated experimental studies with various model parameter settings will be conducted. Initially, functional data is generated using

decreasing sequences, the product of normally distributed random numbers, and Fourier basis functions, as described below:

$$X(t) = \sum_{k=1}^{10} \frac{1}{k^{3/4}} a_k \phi_k(t) \tag{12}$$

Where ak follows a standard normal distribution. For the corresponding coefficient function, the specific form is shown as follows:

$$\beta(t) = \sum_{k=1}^{10} \frac{1}{k^2} \phi_k(t) \tag{13}$$

The connections between the response and predictor variables are assumed as follows:

$$Y = \int_{t \in \mathcal{T}} X(t)\beta(t)dt + sin(\frac{2}{3}\pi S) + \xi \tag{14}$$

Among them, the kernel function is set as the combination kernel function.The combination kernel function is composed of a constant multiplied by a Gaussian regression process plus white noise, S takes the value of D-dimensional standard normal distribution, serving as an error term that is independent of the predictor variable and follows the standard normal distribution. We divided the dataset into training and test sets, with the training set comprising 60% and the test set 40%. The training set was utilized to fit the model parameters, while the test set was used to assess the model's predictive performance. To comprehensively evaluate the model's prediction performance, we employed Monte Carlo simulation, randomly dividing the training and test sets multiple times. We calculated the mean and standard deviation of the experimental error for each iteration and used these values as criteria for evaluating the model's performance. The error in each experiment was quantified using the mean square error (MSE). Additionally, for hyperparameter selection in the model, this paper employs the cross-validation method for optimization. To further validate the effectiveness of the proposed method, several comparative methods were also utilized, including the XGBoost regression model, random forest regression model (RF), support vector regression model (SVR), and Gaussian process regression model (GPR). This paper examines three distinct experimental configurations: various nonlinear connection functions, different sequences for reducing the base expansion coefficient, and various sequences for decreasing the base expansion coefficient functions. The sample size is uniformly set to 100, the dimension of S is uniformly set to 1, and the values are taken from the standard normal distribution. The specific experimental results are presented in Tables 1–3 and Figures 1–3.

Table 1. Means and standard deviations of various methods under nonlinear connection functions

X (t)	FSGPR	XGBoost	RF	SVR	GPR
1/k^ (3/4)	0.003650 (0.006279)	0.268002 (0.077305)	0.235308 (0.054645)	0.297350 (0.119538)	0.238384 (0.101835)
1/k^ (2/3)	0.007528 (0.009180)	0.245825 (0.102654)	0.229472 (0.078490)	0.366616 (0.085397)	0.253431 (0.090531)
1/k^2	0.001141 (0.001122)	0.092323 (0.041994)	0.076397 (0.032652)	0.041336 (0.016448)	0.035965 (0.008571)

The table indicates that FSGPR generally exhibits low Mean Squared Error (MSE) values, high predictive power, and stability, with small standard deviations. In contrast, XGBoost has a significantly higher MSE than FSGPR, along with a large standard deviation, suggesting substantial error variation and inferior stability. RF's MSE is comparable to XGBoost but demonstrates greater stability. SVR suffers from high MSE, poor fitting ability, substantial standard deviation fluctuations, and lacks sufficient stability. GPR's MSE is lower than SVR's but is generally outperformed by FSGPR, XGBoost, and RF. The boxplot reveals that FSGPR has the lowest MSE value, the best prediction performance, minimal errors, and stable results. XGBoost displays low MSE and good

performance, yet it has high variability. RF has high MSE values, moderate performance, and high variability in prediction results. SVR has higher MSE, large errors, and GPR, while relatively stable, also shows higher errors compared to other models.

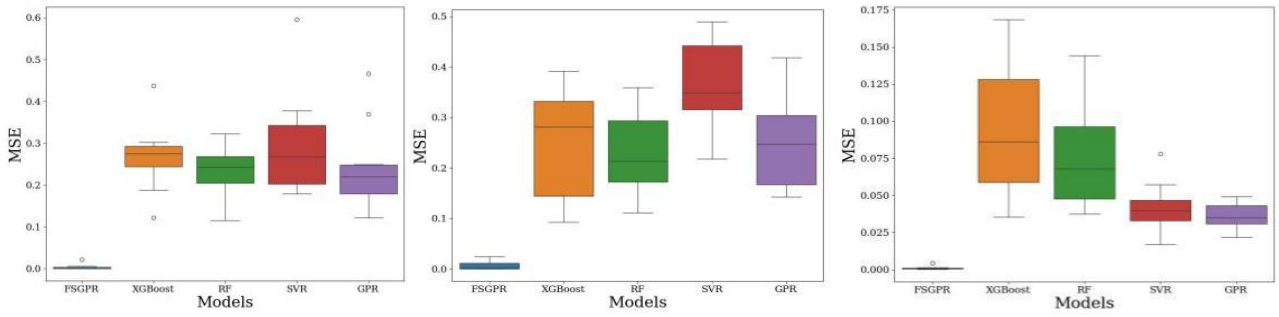

Figure 1. Boxplot of different contrast methods under different nonlinear connection functions

Table 2. Means and standard deviations of different methods across various coefficients

Bata (t)	FSGPR	XGBoost	RF	SVR	GPR
1/k^2	0.003325 (0.003388)	0.182084 (0.057214)	0.211391 (0.053634)	0.269641 (0.058331)	0.197418 (0.048196)
1/k^3	0.004563 (0.004148)	0.224231 (0.088149)	0.215666 (0.069000)	0.218285 (0.077965)	0.194355 (0.083572)
1/k^4	0.002355 (0.002102)	0.218758 (0.116464)	0.203651 (0.080644)	0.272898 (0.063967)	0.201503 (0.039311)

The table indicates that FSGPR exhibits the lowest MSE value among the three functions, accompanied by a small standard deviation and stable predictions. In contrast, XGBoost has a higher MSE value than FSGPR, along with a larger standard deviation and fluctuating prediction performance. RF displays uniform MSE values ranging from 0.203651 to 0.215666, indicating a good fit and moderate stability. SVR, however, suffers from high MSE values, fluctuating standard deviations, and unstable, complex function processing, resulting in large prediction errors. GPR's MSE value is lower than SVR's and slightly higher than FSGPR's, with a small standard deviation and good stability. The boxplot reveals that FSGPR has the lowest MSE, close to 0, signifying high prediction accuracy and consistently stable results. XGBoost's MSE is lower than SVR's but higher than FSGPR's, appearing more discretized with outliers present. RF's MSE value is second only to XGBoost's and is stable, yet the overall ability requires improvement; the box is evenly distributed with a slightly higher number of outliers. SVR generally has high MSE values, long box lengths, numerous outliers, and unstable accuracy. GPR's performance is moderate, generally better than SVR's, with good reliability, a moderate box size, and some existing volatility.

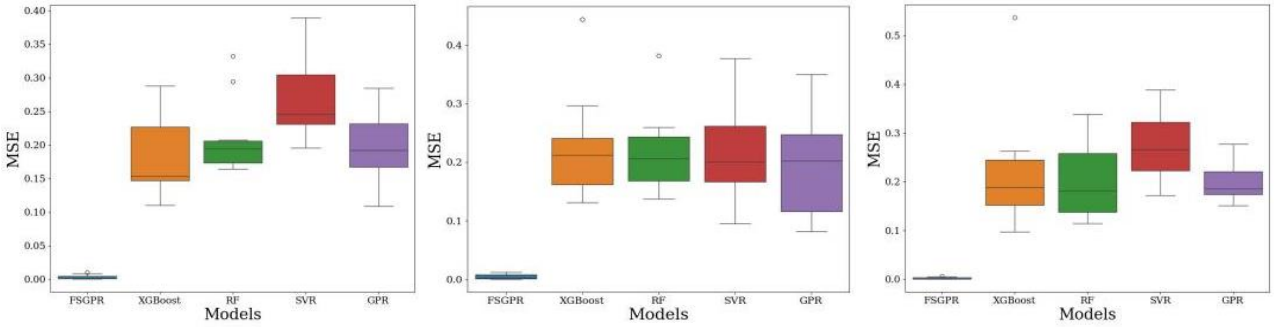

Figure 2. Boxplots of different contrast methods under different expansion coefficients

Table 3. Means and standard deviations of various methods for different coefficient functions

gp	FSGPR	XGBoost	RF	SVR	GPR
Sin (2/3*pi*s)	0.002633 (0.002145)	0.281877 (0.107604)	0.241255 (0.076325)	0.255979 (0.088925)	0.206094 (0.063367)
Sin (3/4*pi*s)	0.007078 (0.007955)	0.270366 (0.109345)	0.264177 (0.061711)	0.328512 (0.078906)	0.268655 (0.053502)
Cos (3/4*pi*s)	0.005986 (0.007101)	0.266291 (0.108365)	0.242801 (0.107047)	0.341716 (0.089923)	0.258335 (0.089280)

The table illustrates the FSGPR model, which boasts the lowest MSE, the best fit, robust stability, a small standard deviation, and is both accurate and consistent. The MSE for XGBoost is higher than that of FSGPR, yet it still provides a good fit and an acceptable standard deviation. RF's MSE is balanced but less accurate than XGBoost. SVR exhibits a high MSE and standard deviation, indicating poor accuracy and stability. GPR performs well in terms of MSE but falls short of FSGPR, with a moderately fluctuating standard deviation.

The boxplot indicates that FSGPR has the lowest MSE, approaching zero, the best fit performance, high prediction accuracy, and stable, consistent results. XGBoost's MSE is low but not as low as FSGPR's, and it still predicts well. RF has a moderate MSE with more fluctuations and outliers compared to XGBoost. SVR generally has a higher MSE than the other models, particularly evident in the second picture with a wide error range and large outliers. GPR's MSE is higher than SVR's and RF's but is less than FSGPR's and XGBoost's, yet it maintains a relatively stable performance with bins of variability and outliers.

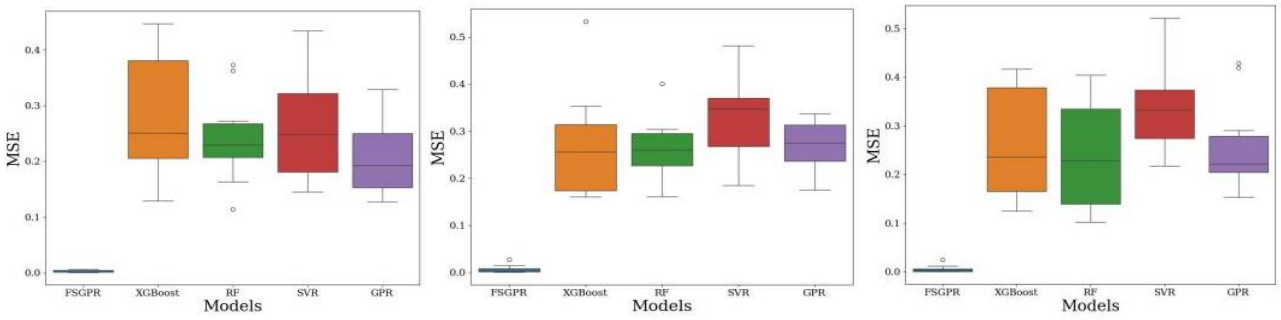

Figure 3. Box plots of different contrast methods

3.2. Real data analysis

The real dataset originates from the website http://lib.stat.cmu.edu/datasets/tecator, which contains 240 samples of meat with moisture, fat, and protein content, as well as absorption spectrum data measured in the 850-1050 nm (nanometer) range. A total of 100 observation points were recorded from the absorption spectrum data for every two bands. The objective of our study was to predict the protein content of meat samples based on their absorption profile data and their moisture and fat content. Through the preprocessed analysis of the data, we plotted the spectral data and further plotted the correlation heat maps, as shown in Figure 4. It can be observed from the thermal maps that the stronger the collinearity between the spectral observation points, the darker the corresponding region appears as red. This indicates that traditional multiple regression methods may not be effective in addressing the relationship between these variables. Therefore, considering the possible nonlinear association between the variables, we should use a functional linear relationship model for the analysis.

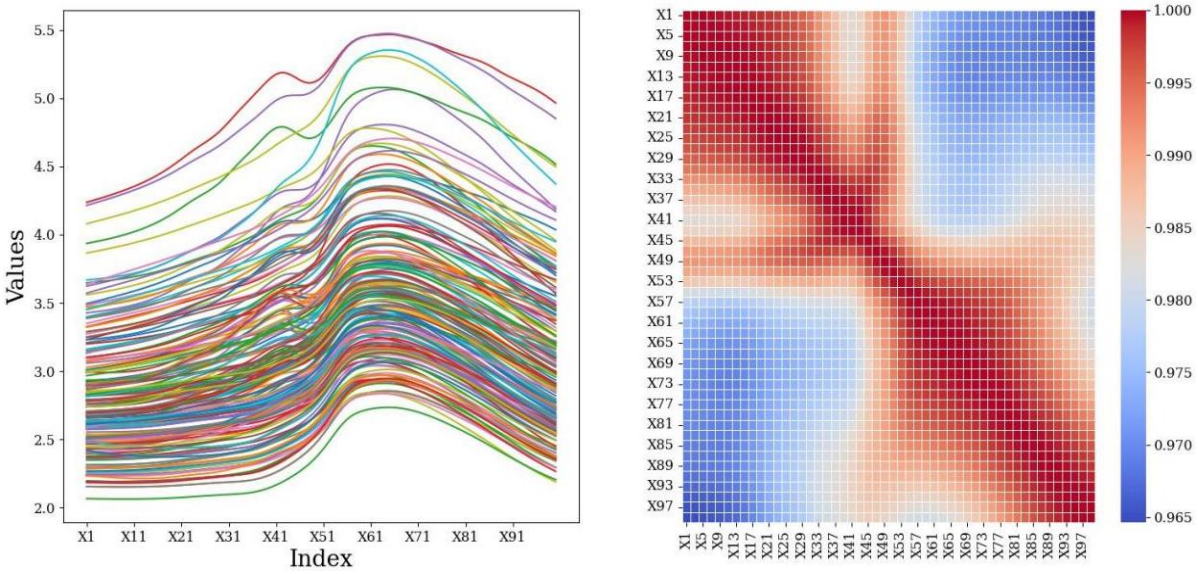

Figure 4. Graph of the spectral data (left) and thermal map (right)

Table 4. Means and standard deviations of mean squared errors for different prediction methods

Method	Mean	SE
FSGPR	0.118658	0.022759
XGBoost	0.164473	0.056658
RF	0.149499	0.045552
SVR	0.181829	0.047648
GPR	0.241874	0.070799

The Monte Carlo method is utilized for prediction performance testing. Specifically, the dataset is divided into a training set and a test set at a ratio of 6:4. The training set is employed for model training, while the test set is used to evaluate the model's performance, with the mean squared error on the test set serving as the evaluation index. This procedure was repeated 10 times, and the mean and standard deviation of the mean squared errors from these 10 repetitions were taken as the final evaluation metrics. The comparison methods selected encompass the XGBoost regression model, the random forest regression model (RF), the support vector regression model (SVR), and the Gaussian process regression model (GPR). The experimental results are presented in Table 4-5 and Figure 5.

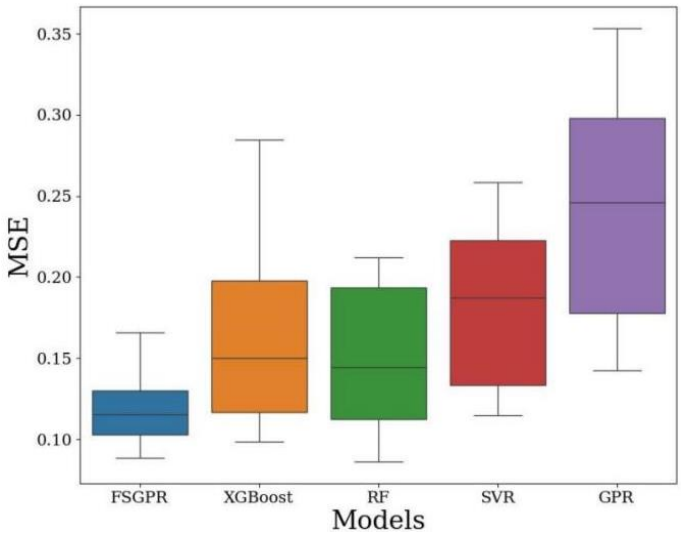

Figure 5. Boxplot of multiple mean square error for different prediction methods

Firstly, the table reveals that the proposed method has the lowest MSE, suggesting that it yields relatively optimal results in terms of prediction performance. Additionally, the standard deviation of the mean square error indicates that the robustness of the proposed method significantly improves with the enhancement of ensemble learning. Secondly, analysis of the boxplot leads to the following conclusions: the median MSE of the FSGPR model is the lowest (approximately 0.10), suggesting it has the best prediction effect with the smallest error and a small variation range (short box), indicating high prediction stability. In contrast, the median of the XGBoost model is around 0.15, performing well, but with greater volatility (longer boxes), indicating that its prediction effect is more susceptible to data fluctuations. The median of the random forest (RF) model is near 0.20, showing average performance, and while the boxes and whiskers indicate some fluctuations, it still outperforms other subsequent models. The median support vector regression (SVR) model was also close to 0.20, similar to RF, but with more outliers, suggesting that the prediction may be unstable in some samples. The Gaussian process regression (GPR) model has the highest median MSE and the largest box range, indicating that the prediction effect of the model is poor. Overall, it is evident from the boxplot that the FSGPR model performs the best, followed by XGBoost and RF, while SVR and GPR performed relatively poorly, especially GPR, with high MSE values and significant fluctuations. Therefore, the partial function linear model based on regenerative kernel theory and ensemble learning performs the best in terms of model fitting and the accuracy of experimental results. By selecting the load prediction results of 403 and 411 lines. We can see that the actual values of the lines basically match the predicted values, but there are also some errors, especially in the peak period of electricity consumption, as shown in Table.1. From the comparison between prediction data and actual data, the BP neural network has better prediction performance and relatively small error, which can meet the demand completely, and has fast prediction speed and convenient operation.

Table 5. Comparison of power load forecasting of 403 line

Comparison	Power	Forecasting
A	12937	92387
B	928735	29837
C	894523	23894

4. Conclusion

Considering that the covariables are functional random variables and finite dimensional European random variables, and the response variables are scalar, a new partially functional linear model is proposed in this paper. On the one hand, the proposed method assumes that there is a linear connection structure between the functional random variable and the scalar response, and uses the functional principal component basis expansion to approximate the structure. On the other hand, for finite dimensional Euclidene covariates, we assume that the connection function between them and the scalar response is in a regenerated kernel Hilbert space, which can be represented by kernel expansion. This design increases the flexibility of the proposed model. In addition, this paper uses the idea of ensemble learning instead of model selection; by building a partial functional linear model with different number of truncations and weighted average, we can deal with the problem of truncation number selection. In the future, more complex models or ensemble learning methods can be used to integrate the results of these sub models. Simulation experiment and actual data analysis show the superiority and applicability of the proposed method. This article is a research-oriented article, and it has also been validated in actual data. For other more application scenarios, we will conduct empirical research in the future.

In addition, there are some issues worth thinking about. First of all, apart from constructing partial functional linear models with different number of truncations in this paper, is there any other ensemble learning method that can be considered in the weighted average processing of these models? Moreover, in the theory of regenerating nuclei, different kernel functions will generate different regenerating Hilbert Spaces, so how to choose a better kernel function is also a problem; In addition, if there are

random effects between the parts of the predictor variables, such as autocorrelation effects, the regenerated kernel theory is not applicable. Therefore, we can also consider the next step, if the random variable is a Gaussian process, combined with the Gaussian process regression model to construct a more efficient solution. And explore the application of different nuclear functions in different scenarios in the future.

References

[1] Yang Weiming, Li Mingjie. Spatial function autoregressive model based on Bayesian estimation and its application [J]. Journal of Chongqing Technology and Business University (Natural Science Edition), 2024, 41 (03): 104 - 112. (in Chinese).

[2] Gertheiss J, Rügamer D, Liew B X W, et al. Functional data analysis: An introduction and recent developments[J]. Biometrical Journal, 2024, 66(7): e202300363.

[3] Sun Qantang. Functional data analysis method and its application in financial field [D]. Tianjin University, 2012.

[4] Zhu Rong, Zou Guohua, Zhang Xinyu. ensemble learning method for Partial function linear models [J]. Systems Science and Mathematics,2018, 38 (07): 777 - 800. (in Chinese).

[5] Yuan Xiaohui, Cao Ruya, Jin Wanlin. Functional partial linear model parameter estimation and variable selection [J]. Journal of jilin normal university (natural science edition), 2022 lancet (02): 48 - 56.

[6] Chen Yuqing, Ling Nengxiang. Empirical analysis of the relationship between PM2.5 concentration and temperature based on functional data model [J]. University Mathematics, 2024, 40 (04): 1 - 6.

[7] Yan Xingyu. Linear regression function types a number of research [D]. East China normal university, 2020.

[8] Li Songxuan. Research and application of generalized partial functional linear models of unknown connection functions [D]. The northern industrial university, 2024.

[9] Zhang Bo, Fan Chao. Partial linear models with kernel functions and their applications [J]. Statistical Research, 2020, 37 (01): 110 - 128. (in Chinese).

[10] Lu Peng, Wang Xiaoling, Yu Hongling, et al. Dynamic extension evaluation model for dam seepage safety based on FDA [J]. Journal of Hohai University (Natural Science), 2020, 48 (5): 433 - 439. (in Chinese).

[11] Zhang Baoxue, Chen Qingmei, Wang Bingshen, et al. Dynamic comprehensive evaluation of ecological construction in the Yangtze River Economic Belt based on functional data [J]. Journal of Technical Economics, 2019, 39 (6): 131 - 140.

[12] Zhang Bo, Fan Chao. Partial linear models with kernel functions and their applications [J]. Statistical Research, 2020, 37 (01): 110 - 128. (in Chinese).

[13] Lu Peng, Wang Xiaoling, Yu Hongling, et al. Dynamic extension evaluation model for dam seepage safety based on FDA [J]. Journal of Hohai University (Natural Science), 2020, 48 (5): 433 - 439. (in Chinese).