LSTM Ensemble Learning Model with Additive Gaussian Process Priors

Zhuo Sun[1, #], Yishu Wang[2, #], Hanlin Yin[3, #, *]

[1]School of Science, Yanshan University, Qinhuangdao, China, 066004

[2]School of Business, China University of Political Science and Law, Beijing, China, 102249

[3]School of Management Science and Engineering, Nanjing University of Information Science&Technology, Nanjing, China, 210044

*Corresponding author: yinhlin@126.com

#These authors contributed equally.

Abstract. Deep learning technology is one of the key research directions in the field of machine learning, especially when dealing with high-dimensional and non-linear data prediction tasks, but it can not avoid the problem of overfitting and underfitting of prediction models. Taking LSTM neural network as an example, this paper proposes an ensemble learning model based on additive Gaussian process priors. On the one hand, the proposed method uses bootstrap technology to realize the randomization of neural network model, so that the network can capture effective information of predictor variables from multiple perspectives. On the other hand, this method takes the set of neural network models after randomization as a new input variable, and uses Gaussian process additive model to integrate the results of different models. By designing orthogonal additive kernel, the marginal effect and interaction effect of LSTM neural network are measured. In addition, the proposed method can quantitatively estimate the uncertainty of the forecast results. Simulation experiment and actual data analysis show that the proposed method is more competitive than some classical regression models.

Keywords: LSTM neural network, Gaussian process, Ensemble learning.

1. Introduction

In the contemporary era marked by economic globalization and digital revolution, the field of economic statistical modeling is encountering unparalleled prospects and challenges. On the one hand, the intricacy and interconnected nature of economic activities have intensified considerably, thereby rendering conventional econometric models incapable of adequately capturing the nonlinear interconnections among economic variables. On the other hand, the advent of big data technology has endued economic statistics with a wealth of granular micro-level data, while simultaneously posing the challenge of the "curse of dimensionality". Fan and Li [1] have methodically encapsulated the pivotal issues confronted by high-dimensional statistical learning, with particular emphasis on the propensity of traditional parametric and non-parametric approaches to succumb to the curse of dimensionality, resulting in diminished estimation efficiency when coping with high-dimensional economic datasets. Fan et al. [2] have further delved into the theoretical breakthroughs in high-dimensional mean regression estimation, proffering innovative perspectives for addressing these challenges.

In the recent past, deep learning technology has emerged as a novel paradigm in economic statistics, leveraging its exceptional feature learning capabilities and implicit regularization properties. Gu et al. [3] have meticulously compared the efficacy of diverse machine learning algorithms in the domain of asset pricing, discovering that deep learning models possess marked superiorities in managing extensive financial datasets. Lim and Zohren [4] have expounded in their review that deep learning has achieved significant advancements in time series forecasting, especially in the context of nonlinear and non-stationary data. Among the pantheon of deep learning architectures, LSTM has garnered distinction due to its unique gating mechanism and proficiency in modeling temporal

dependencies, showcasing pronounced utilities in economic and financial domains. Fischer and Krauss [5] have demonstrated the efficacy of LSTM networks in stock market prognostication, adeptly capturing the market's long-term dependency relationships. Zhang et al. [6] have delineated the theoretical underpinnings and application paradigm of LSTM within the realm of deep learning, particularly in financial time series analysis. In the sphere of bond market research, Bianchi et al. [7] have utilized LSTM networks to dissect bond risk premiums, with their model significantly outperforming traditional methodologies in out-of-sample testing. Nevertheless, extant research is still beset by significant constraints. For instance, Molnar et al. [8] have highlighted that the "black box" nature of deep learning models substantially circumscribes their practical application in the economic sphere. Gu et al. [9] underscored the crucial aspect of interpretability within deep learning models, particularly in their exploration of autoencoder asset pricing models. They proposed an explanatory framework hinged on latent factors. In a comprehensive review, Liu and Chen [10] systematically delineated the principal challenges associated with deploying deep learning methodologies in the financial domain, notably spotlighting constraints in risk factor extraction and portfolio management. Bu et al.'s research [11] further elucidated that deep learning models contend with the dual challenges of overfitting and underfitting in financial applications, thereby directly influencing the reliability of these models.

To combat these challenges, this study introduces a novel deep ensemble learning framework founded on additive Gaussian process priors. This framework introduces three principal innovations: (1) the proposal of an ensemble learning model leveraging LSTM neural networks, amalgamated with additive Gaussian process priors to forge a two-layer regression framework; (2) the assessment of the marginal and interaction effects within the LSTM model class through orthogonal additive kernels; and (3) the provision of uncertainty quantification for prediction outcomes from a probabilistic standpoint. Through simulation experiments and real data analysis, the efficacy of the model is empirically validated.

2. LSTM Ensemble Learning Model with Additive Gaussian Process Priors

LSTM network captures complex nonlinear relationships through its internal structure, which makes it perform well in processing nonlinear data. Therefore, in the nonlinear prediction of time series data, we use LSTM model to abstract the input data layer by layer and extract the features to dig out the deep underlying rules. Let's assume the model for Y and X is as follows:

$$\begin{aligned} &Y = \mathrm{f}\left(X\right) + \varepsilon \\ &Y \in R, X = \left(X_1, X_2, \cdots, X_p\right) \in R^p \end{aligned} \tag{1}$$

where X is the P-dimensional input variable, Y is the target value, and f is assumed to where X is the P-dimensional input variable, Y is the target value, and f is assumed to be the LSTM neural network. A single LSTM model is prone to overfitting when the amount of training data is relatively small or the model complexity is high, so we adopt the idea of ensemble learning. Firstly, the training set with sample size n was sampled, and K Bootstrap sample sets were obtained, as follows:

$$\left(X_k, y_k\right) = \left\{\left(X_{\mathrm{jk}}, y_{jk}\right) \mid X_{jk} \in R^p, y_{jk} \in R, k = 1 \cdots K, j = 1 \cdots n\right\} \tag{2}$$

For each generated K Bootstrap sample set, a separate LSTM neural network model was trained, and the predicted value of the k and j samples was obtained

$$M_{jk} = f_k\left(X_{jk}\right) + \varepsilon_{jk} \tag{3}$$

where is the KTH LSTM neural network, then we obtained

$$M_j = \left(M_{j1}, M_{j2}, \cdots M_{jK}\right) \in R^K \tag{4}$$

It is obtained by K LSTM neural networks, and then a re-encoding of X is realized. Let's say we get a new set of samples.

$$(M, y) = \left\{ (M_j, y_j) \mid M_j \in R^K, y_j \in R, j = 1 \cdots n \right\} \tag{5}$$

Given a prior as an additive Gaussian process,

$$y_j = f(M_j) = f_1(M_{j1}) + f_2(M_{j2}) + \cdots + f_{12}(M_{j1}, M_{j2}) + \cdots + f_{12\cdots K}(M_{j1}, M_{j2}, \cdots, M_{jK}) \tag{6}$$

In the Gaussian process model, the additive structure of the function decomposition is enforced by the structure of the kernel, and its decomposition can be constructed as follows: first for each input dimension $i \in \{1 \cdots K\}$ Assign a one-dimensional base kernel $k_i(M_{ji}, M'_{ji})$, Then define the additive kernel functions of order 1, order 2 to order d as follows:

$$\begin{aligned} k_{add_1}(M_j, M'_j) &= \sigma_1^2 \sum_{i=1}^{K} k_i(M_{ji}, M'_{ji}), \\ \mathrm{k}_{add_2}(M_j, M'_j) &= \sigma_2^2 \sum_{i=1}^{K} \sum_{h=i+1}^{K} k_i(M_{ji}, M'_{ji}) k_h(M_{jh}, M'_{jh}), \\ \mathrm{k}_{add_d}(M_j, M'_j) &= \sigma_d^2 \sum_{1 \le i_1 \le i_2 \le \cdots \le i_d \le K} \left[\prod_{l=1}^{d} k_{i_l}(M_{ji_l}, M'_{ji_l}) \right] \end{aligned} \tag{7}$$

Specifically, a first-order additive kernel function is the sum of all fundamental kernel functions, a second-order additive kernel function is the sum of the product of all possible pairwise fundamental kernel functions, and so on, up to the D-order additive kernel function.The parameter σ_d^2 controls how much each additive component contributes to the overall kernel function.In an additive Gaussian process, each function component f_i satisfies the following orthogonality constraint, as follows:

$$\int_{\chi_i} f_i(M_{ji}) p_i(M_{ji}) dM_{ji} = 0 \tag{8}$$

where χ_i and p_i are the sample space and density of the input features, respectively. In order to satisfy the orthogonality constraint, the basic kernel function is adjusted. For each feature i and its underlying kernel k_i, each function component can still be regarded as a Gaussian process given $\int f_i(M_{ji}) p_i(M_{ji}) dM_{ji} = 0$, as follows:

$$f_i(.) \mid \int f_i(M_{ji}) p_i(M_{ji}) dM_{ji} = 0 \square\ GP(0, \tilde{k}_i) \tag{9}$$

However, its kernel function $\tilde{k}_i$ will change, and the changed kernel function $\tilde{k}_i$ satisfies the following formula:

$$\tilde{k}_i(M_{ji}, M'_{ji}) = k_i(M_{ji}, M'_{ji}) - E\left[S_i f_i(M_{ji}) \right] E[S_i^2]^{-1} E[S_i f_i(M'_{ji})], \tag{10}$$

among them,

$$\begin{aligned} E\left[S_i f_i(\cdot) \right] &= \int p_i(M_{ji}) k_i(M_{ji}, \cdot) dM_{ji}, \\ E[S_i^2] &= \iint p_i(M_{ji}) p_i(M'_{ji}) k_i(M_{ji}, M'_{ji}) dM_{ji} dM'_{ji} \end{aligned} \tag{11}$$

We call this the constraint kernel.In order to ensure the orthogonality of higher-order interaction terms, an orthogonality constraint needs to be applied to each function f_u of higher-order interaction terms (where u is a subset of the set [K]), that is, f_u must satisfy the measure integral of its input characteristic dimension is zero, as follows:

$$\int_{\chi_i} f_u\left(M_{ju}\right) p_i\left(M_{ji}\right) dM_{ji} = 0, \forall i \in u, M_{ju} := \left\{M_{ji}\right\}_{i \in u} \tag{12}$$

In order to satisfy the above orthogonality constraints, we construct a special constraint kernel for the higher-order interaction term function f_u . This constraint kernel is obtained by multiplying the one-dimensional constraint kernel, that is, for any u$\subseteq$[K]:

$$\tilde{k}_u\left(M_j, M'_j\right) = \prod_{i \in u} \tilde{k}_i\left(M_{ji}, M'_{ji}\right) \tag{13}$$

Due to the construction of the constrained kernel and the assumption of independence between the input feature dimensions, it can be proved that the higher-order interaction function constructed using the constrained kernel satisfies the orthogonal constraint condition. Since the sum and product of a kernel function can be used to build a kernel with a specific structure, including an orthogonal structure, we can devise a way to build our model by substituting a one-dimensional, appropriately constrained kernel function (10) into an additive kernel function (7). The Kernel function built by this method is called a Orthogonal Additive Kernel (OAK). If the kernel function is a Gaussian kernel function:

$$k_i(M_i, M_i^{'}) = \exp(-\frac{(M_i - M_i^{'})^2}{2l_i^2}) \tag{14}$$

The analytic expression of ortho-intersection kernel obtained by OAK method,

$$\tilde{k}(M_i, M_i^{'}) := \exp(-\frac{(M_i - M_i^{'})^2}{2l^2}) - \frac{l\sqrt{l^2 + 2\delta^2}}{l^2 + \delta^2} \times \exp\left(-\frac{\left((M_i - \mu)^2 + (M_i^{'} - \mu)^2\right)}{2(l^2 + \delta^2)}\right) \tag{15}$$

After derivation, the posterior mean function of each f_u in the additive Gaussian regression model is

$$m_u(\square) = \sigma_u^2 \left(\square_{i \in u} \tilde{k}_i(\square, M_i)\right) K(M, M)^{-1} y \tag{16}$$

where $K(M, M)$ represents the total covariance matrix of the recharacterized feature, M_i represents the i th column of the recharacterized feature matrix, and $\sigma_{|u|}^2$ is the interactive weight parameter associated with the order |u|. The corresponding post-verified difference function is

$$\begin{aligned} &V[m_u(\square)] = \sigma_{|u|}^4 y^T K(M, M)^{-1} \square_{i \in u} \\ &\left(\int \tilde{k}_i(\square, M_i) \tilde{k}_i(\square, M_i)^T dp_i(\square)\right) K(M, M)^{-1} y \end{aligned} \tag{17}$$

As can be seen from (16), the final prediction result of the proposed method is the sum of (17). In the OAK model, the variance of each function component can be used to measure the marginal effect of a single feature or the interaction effect between features, which is also called the Sobol index. In this context, this means that the relative importance of different models and the relative importance of different model interactions can be measured. In order to make the sum of Sobol index equal to 1, the Sobol index is normalized and the normalized Sobol index is obtained as follows:

$$\tilde{R}_{\mathrm{u}} = \frac{V\left[\mathrm{m}_{u}(\square)\right]}{V\left[\sum_{u\subseteq[K]} \mathrm{m}_{u}(\square)\right]} \tag{18}$$

Faced with the task of fitting complex nonlinear data, a single LSTM model is often powerless, and its anti-interference ability is relatively weak. In order to solve this problem, this paper adopts the advanced concept of ensemble learning and integrates multiple LSTM models to effectively deal with the phenomenon of model overfitting and significantly improve the generalization ability of the model. Specifically, we use multiple LSTM models to deeply learn the original sample data X, and output the re-encoding results of the original data. These newly generated sample data are then input into the additive Gaussian process regression model to further improve the interpretability of the model. It is worth mentioning that compared with the traditional high-dimensional interaction model, the additive Gaussian process achieves the low-dimensional representation of the data by cleverly introducing orthogonal constraints. This feature not only reduces the complexity of the model, but also significantly reduces the computational cost, making the model more efficient and practical in practical applications. In summary, by integrating multiple LSTM models and introducing additive Gaussian process regression, the method proposed in this paper has shown significant advantages in solving model overfitting, improving generalization ability, reducing model complexity and reducing computation cost.

3. Data Analysis

3.1. Simulation experiment analysis

In this paper, model performance of the proposed AGP_LSTM model was evaluated and the applicability of the model was explored by changing the sample size, the distribution of predictor variables, the dimension of predictor variables and the connection function between response variables and predictor variables. For comparison methods, we chose LASSO regression model, ridge regression model, random forest, XGBOOST model and other models to compare and analyze the prediction effect. Monte Carlo simulation experiment was used in the experiment. Specifically, the mean and standard deviation of absolute error of each experimental result were calculated through repeated experiments as evaluation indexes. The smaller the mean value, the higher the prediction accuracy of the model, and the smaller the standard deviation, the stronger the robustness of the model. First, this paper substituted the simulation data with sample sizes of 100, 150 and 200 into the AGP_LSTM model, LASSO regression model, Ridge regression model, random forest and XGBOOST model respectively, and obtained the absolute error and standard deviation of the prediction as shown in Table.1. below:

Table.1. The predictions of the five models (sample size change)

method	100 Sample size		150 Sample size		200 Sample size	
	AE	SD	AE	SD	AE	SD
AGP_LSTM	0.0468	0.0203	0.0307	0.0087	0.0240	0.0050
LASSO	0.2502	0.0446	0.2524	0.0316	0.2625	0.0190
Ridge	0.2659	0.0525	0.2759	0.0429	0.2889	0.0189
RF	0.0905	0.0202	0.0853	0.0171	0.0835	0.0110
Xgboost	0.0977	0.0200	0.0966	0.0217	0.0838	0.0132

As can be seen from the above table, the absolute error and standard deviation of the AGP_LSTM model gradually decrease with the increase of the sample size, indicating that the model constructed in this paper is suitable for the prediction problem with a large amount of data. Meanwhile, the absolute error and standard deviation of the AGP_LSTM model are smaller than that of other models, indicating

that the prediction performance of the AGP_LSTM model is better than that of other models. With the sample size of 100, 150 and 200 for the five models, the absolute error box lines 1-3 of the ten simulation prediction results are as follows:

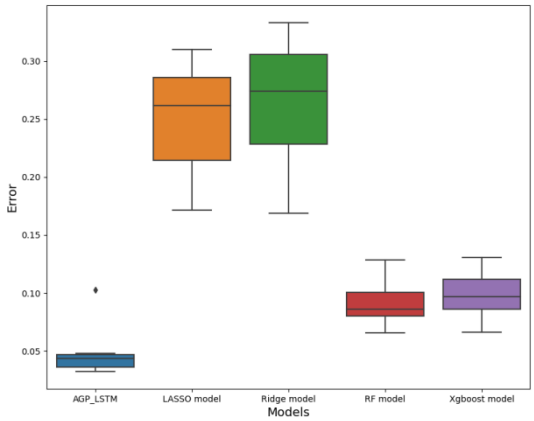


Figure 1. Absolute error (n=100)

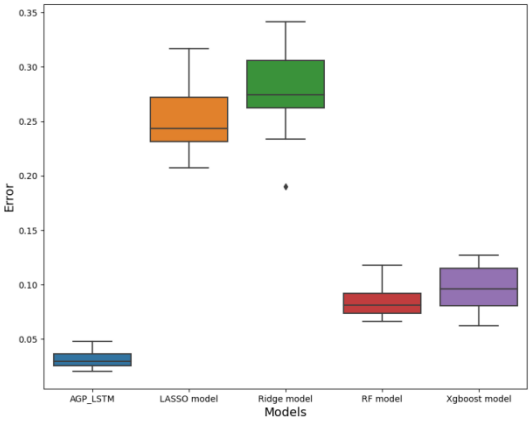


Figure 2. Absolute error (n=150)

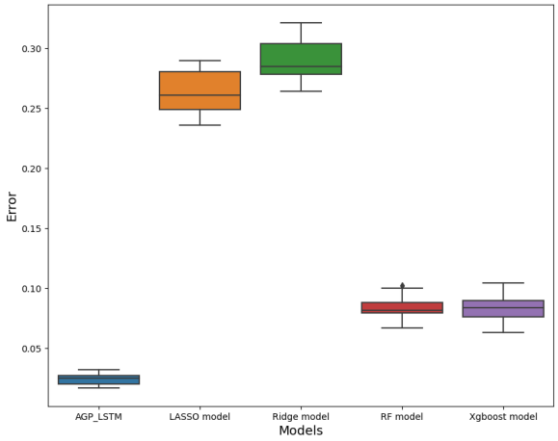


Figure 3. Absolute error box diagram (n=200)

It can be seen from the box plot of the absolute errors of the 10 simulation prediction results with sample sizes of 100, 150 and 200 respectively that the absolute errors of the AGP_LSTM model are smaller and more concentrated than those of the other four models, indicating that the AGP_LSTM model has better prediction effect and is more stable.

Secondly, simulation data with normal distribution of mean 0, variance 0.1, mean 0, variance 0.5, mean 0, variance 1 and sum were substituted into AGP_LSTM model, LASSO regression model, Ridge regression model, random forest and XGBOOST model respectively, and the absolute error and standard deviation of prediction were obtained as shown in table.2. below:

Table.2. Absolute error and standard deviation of the five models(change in distribution)

	σ=0.1		σ=0.1		σ=0.1	
	AE	SD	AE	SD	AE	SD
AGP_LSTM	0.0240	0.0050	0.4233	0.0343	0.7635	0.1057
LASSO	0.2625	0.0190	0.9660	0.0754	1.0536	0.1658
Ridge	0.2889	0.0189	1.2420	0.0750	2.5965	0.3309
RF	0.0835	0.0110	0.5940	0.0381	0.8281	0.0941
Xgboost	0.0838	0.0132	0.6069	0.0553	0.8117	0.1018

It can be concluded from the above table that when predicting X with normal distribution and t distribution subject to different means and variances, AGP_LSTM model has smaller absolute error and better prediction effect than other models. Under different X distribution of the five models, the absolute error box diagram 4-6 of the ten simulation prediction results is as follows:

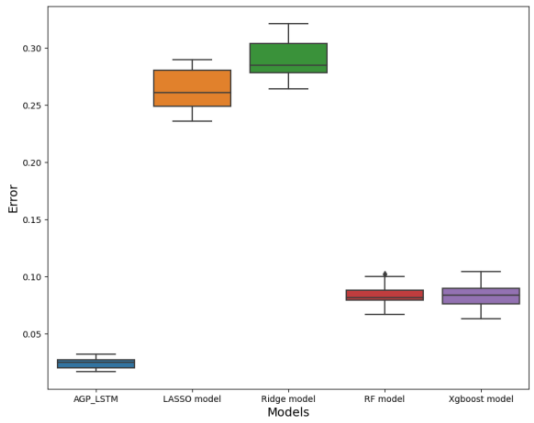


Figure 4. Absolute error (μ=0,σ=0.1)

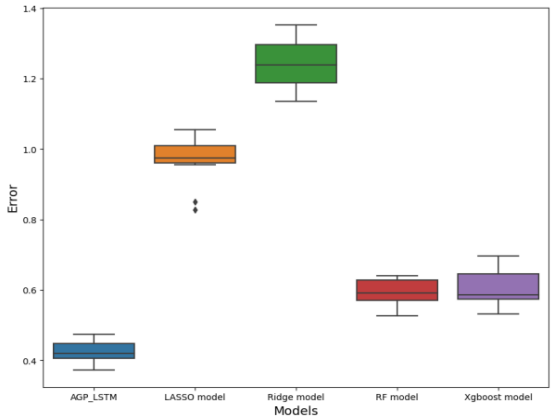


Figure 5. Absolute error (μ=0,σ=0.5)

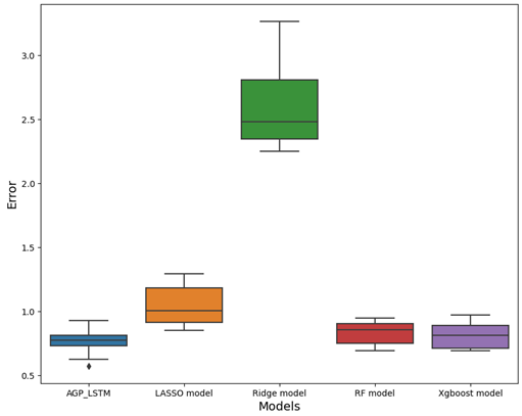


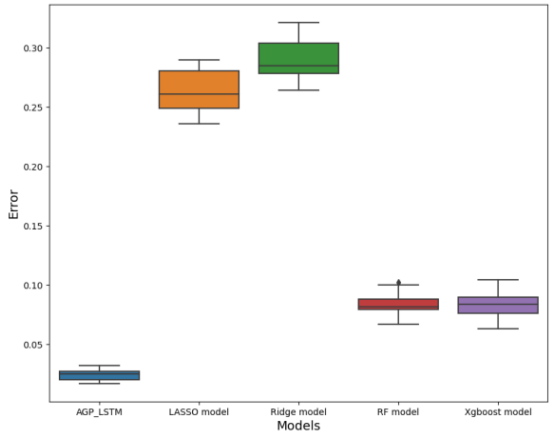
Figure 6. Absolute error box diagram (μ=0,σ=1)

As can be seen from the box diagram of the absolute error of the ten simulation prediction results of the five models under different X distributions, the absolute error of the AGP_LSTM model is smaller and more concentrated than that of the other four models, indicating that the AGP_LSTM model has better prediction effect and is more stable. Then, we substituted the simulation data with 5-dimensional features and 10-dimensional features into the AGP_LSTM model, LASSO regression model, ridge regression model, random forest model and XGBOOST model respectively, and obtained the absolute error and standard deviation of the prediction as shown in table.3..

Table.3. Absolute error and standard deviation (change of characteristic dimension)

	5 dimension		10 dimension	
	AE	SD	AE	SD
AGP_LSTM	0.0240	0.0050	0.1300	0.0204
LASSO	0.2625	0.0190	0.2829	0.0373
Ridge	0.2889	0.0189	0.3181	0.0398
RF	0.0835	0.0110	0.1531	0.0117
Xgboost	0.0838	0.0132	0.1664	0.0137

As can be seen from the above table, when predicting X with different feature dimensions, AGP_LSTM model has smaller absolute error and better prediction effect than other models. Under X with different feature dimensions of the five models, the absolute error box diagram 7-8 of the ten simulation prediction results is as follows:



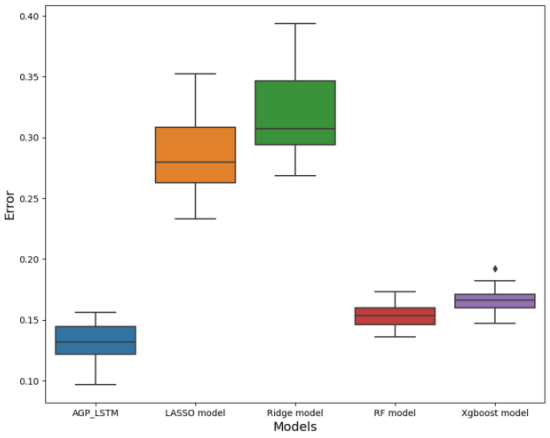


Figure 7. Absolute error (5 dimension) Figure 8. Absolute error (10 dimension)

It can be seen from the box diagram of the absolute error of the ten simulation prediction results of the five models with different feature dimensions X that the absolute error of the AGP_LSTM model is smaller and more concentrated than that of the other four models, indicating that the AGP_LSTM model has better prediction effect and is more stable. Finally, the influence of different connection functions between the predictor variable and the response variable on the prediction effect is tested. Two connection functions are constructed in this paper for experiment. The connection functions are as follows

connection function 1:

$$y = \sin\left(\frac{3\pi}{4}X_1\right) + \cos\left(\frac{2\pi}{3}\left(X_2 + X_3 + X_4\right)\right) + 2X_5 \tag{19}$$

connection function 2:

$$y = \sin\left(\frac{3\pi}{4}X_1\right) + X_2^3 + X_3X_4 + 2X_5 \tag{20}$$

The simulation data were substituted into the AGP_LSTM model, LASSO regression model, Ridge regression model, random forest and XGBOOST model, and the absolute errors and standard deviations of the prediction results of different connection functions were obtained, as shown in table.4. below:

Table.4. Absolute error and standard deviation (change of connection function)

	connection function 1		connection function 2	
	AE	SD	AE	SD
AGP_LSTM	0.0240	0.0050	0.0151	0.0043
LASSO	0.2625	0.0190	0.2727	0.0239
Ridge	0.2889	0.0189	0.3087	0.0260
RF	0.0835	0.0110	0.0548	0.0100
Xgboost	0.0838	0.0132	0.0560	0.0132

It can be concluded from the above table that under different connection functions between y and X, AGP_LSTM model has smaller absolute error in prediction results and better prediction effect than other models, indicating that the model has good stability. Under different connection functions between y and X for the five models, the absolute error box diagram 9-10 of the ten simulation prediction results is as follows:

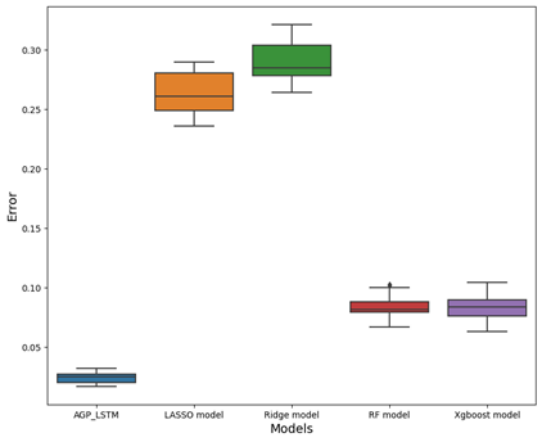


Figure 9. Absolute error box diagram 1

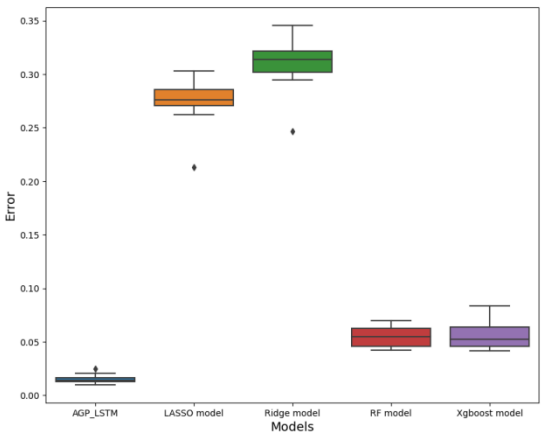


Figure10. Absolute error box diagram 2

As can be seen from the box diagram of the absolute error of the ten simulation prediction results of the five models under different connection functions between y and X, the absolute error of the AGP_LSTM model is smaller and more concentrated than that of the other four models, indicating that the AGP_LSTM model has better prediction effect and is more stable.

3.2. Empirical data analysis

In order to verify the effectiveness of the AGP_LSTM model, this paper selects the spectral data of meat and corn samples measured by the near-infrared spectroscopy analyzer, and compares the prediction results with the LASSO regression model, ridge regression model, random forest, and XGBOOST model. In this paper, Absolute Error (AE) and Standard Deviation (SD) are used to evaluate the prediction accuracy of the model. Among them, the absolute error is used to measure the absolute gap between the predicted value and the actual value, and the standard deviation is used to reflect the dispersion degree of the forecast error. The smaller the statistical values of the two indicators are, the better the forecasting performance of the model is.

Firstly, the first prediction task is meat protein content prediction. Specifically, under the strategic background that China attaches great importance to food safety and nutrition and health, the prediction of meat protein content is not only related to the improvement of the quality and efficiency of the food industry, but also a key link to achieve the goal of national health and promote agricultural modernization. Meat is an important source of protein intake for human beings, and its protein content directly affects the nutritional value of food, market competitiveness and the health and well-being of consumers. Therefore, the AGP_LSTM prediction model of meat protein content was constructed, aiming to provide scientific basis and decision support for the relevant national departments to formulate meat product quality standards and guide meat production and processing.

Meat samples of spectral data set can be downloaded from http://lib.stat.cmu.edu/datasets/tecator, containing 240 meat samples data, Each sample included the moisture, fat and protein content of the meat, as well as the absorption spectrum measured by the near-infrared spectrometer in the band range of 850-1050 nm, with a total of 100 absorption spectrum data at every two bands interval. This paper aims to predict the protein content of this meat sample based on the absorption spectrum data. In this paper, the first 50 absorption spectrum data were selected as independent variables, and protein content was selected as dependent variable. Then, the first 90 sample data are selected as the training set to construct the AGP_LSTM prediction model, and the constructed model is used to predict the subsequent 10 sample data. By comparing the prediction results with the real observed values, the prediction performance of the model was evaluated, and compared with the LASSO regression model, ridge regression model, random forest and XGBOOST model.The absolute errors and standard deviations of the prediction results of different models are shown in Table.5.

As can be seen from Table.5, the absolute error of the prediction result of AGP_LSTM model is 1.15597, which is the best among the five models, indicating that the deviation between the prediction result and the true value is the smallest and the prediction accuracy is the highest. At the same time,

the standard deviation of AGP_LSTM model is 0.238084, which is also at a relatively low level, ensuring the stability of the prediction results.

Table.5. The prediction results of meat protein content by different models

index	AGP_LSTM	LASSO	Ridge	RF	XGBOOST
AE	1.1559	2.6366	3.0266	1.4658	1.3803
SD	0.2380	0.1685	0.1447	0.1775	0.2741

Figure 11 visually shows the comparison between the AGP_LSTM model and the other four machine learning models (LASSO, Ridge, RF and Xgboost) in terms of the error distribution. The figure shows that the AGP_LSTM model has a smaller error distribution range and a lower median error. This indicates that the model has high accuracy and stability in forecasting. In contrast, the error distribution of the other four models is more dispersed with higher median errors, showing greater volatility and uncertainty. In conclusion, the AGP_LSTM model is an excellent machine learning model with high accuracy and stability in meat protein content prediction.

The second prediction task is corn oil content prediction. Specifically, corn is a strategic crop in China, and the prediction of oil content plays a pivotal role in ensuring national food security, promoting agricultural supply-side structural reform and promoting the high-quality development of corn industry. Under the background of the new era, with the great attention of China to agricultural modernization and the vigorous development of corn processing industry, accurate prediction of corn oil content has become the key to enhance the overall competitiveness of corn industry chain and optimize resource allocation. In this paper, the AGP_LSTM prediction model of corn oil content was constructed. Through accurate prediction of corn oil content, it will help the in-depth implementation of the national food security strategy.

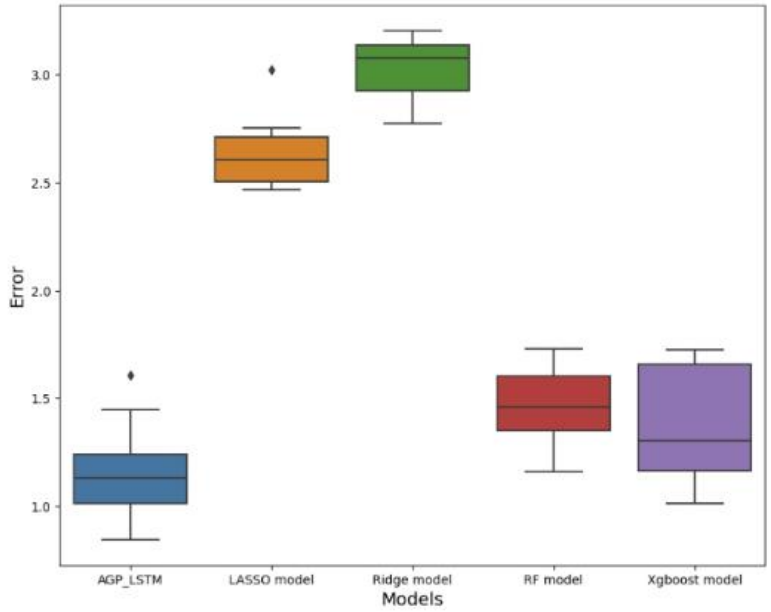


Figure 11. Error box plots of each model

Corn data set can be downloaded from http://www.eigenvector.com/data/Corn/index.html, containing 80 corn sample data, Each sample included the oil, water, starch content of the corn and the absorption spectrum measured by the near-infrared spectroscopy analyzer in the band of 1100-2498 nanometers, with a total of 700 absorption spectrum data at intervals of every two bands. The purpose of this paper is to predict the oil content of this corn based on the absorption spectrum data. In this paper, the first 100 absorption spectrum data were selected as independent variables, and protein content was selected as dependent variables. Then, the first 90 sample data were selected as the training set, the AGP_LSTM prediction model was constructed, and the subsequent 10 sample data were predicted by using the constructed model. By comparing the prediction results with the real observed

values, the prediction performance of the model was evaluated, and compared with LASSO regression model, Ridge regression model, random forest and XGBOOST model. The absolute error and standard deviation of the prediction results of different models were shown in Table.6. As can be seen from Table.6, the absolute error of the prediction result of AGP_LSTM model is 0.14545, which is the best among the five models, indicating that the deviation between the prediction result and the true value is the smallest and the prediction accuracy is the highest. At the same time, the standard deviation of AGP_LSTM model is 0.0209, which is also the best among the five groups of models, which proves that the prediction results of this model have strong stability.

Figure 12 shows a comparison of the error distributions of the five models. As can be seen from the figure, the median error of AGP_LSTM model is low, indicating that the overall error of its prediction results is relatively small. Moreover, the prediction results of AGP_LSTM model are relatively stable, which reflects its strong robustness. In summary, the AGP_LSTM model has high accuracy and stability in the prediction of corn oil content.

Table.6. The prediction results of corn oil content by different models

index	AGP_LSTM	LASSO	Ridge	RF	XGBOOST
AE	0.1454	0.1501	0.1526	0.1510	0.1642
SD	0.0209	0.0258	0.0257	0.0266	0.0288

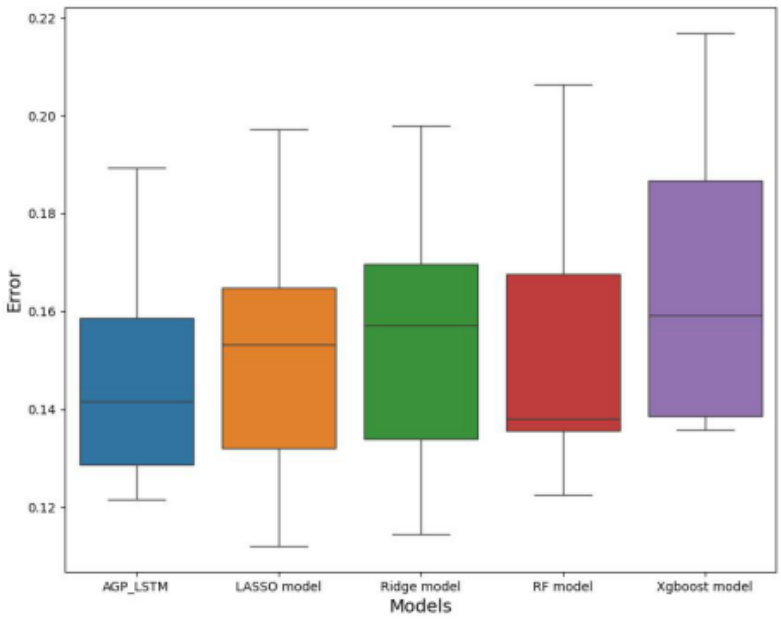


Figure 12. Error box plots of each model

4. Conclusion

The present study introduces a deep learning ensemble framework that adeptly balances the bias-variance trade-off of neural network models by incorporating the outputs of neural network models as features and employing an additive Gaussian process regression model for prediction. On one hand, it orchestrates the integration of information from individual neural network submodels. On the other hand, it quantifies the uncertainty of predictive outcomes from a probabilistic standpoint. Moreover, the proposed methodology evaluates the marginal and interaction effects of neural network submodels via an additive orthogonal kernel. Simulation experiments and empirical data analyses have confirmed that the proposed approach exhibits enhanced predictive accuracy compared to several canonical prediction models. As for future research trajectories, the first pertains to the selection of hyperparameters, including the number of neural network submodels, the order of interaction effects, and the quantity of bootstrap samples. The second concerns the issue of computational complexity, as an increase in the number of submodels escalates the computational expenditure of the additive

Gaussian process regression model, necessitating the selection of numerous submodels to ensure effective prediction while streamlining the model. Finally, this paper exemplifies the application of the proposed framework using LSTM neural networks, with further investigation required to ascertain its applicability to other neural network architectures.

References

[1] Fan, J., & Li, R. (2020). Statistical Learning with High-Dimensional Data. Annual Review of Statistics and Its Application, 7, 149-176.

[2] Fan, J., Li, Q., & Wang, Y. (2020). Estimation of High Dimensional Mean Regression in the Absence of Symmetry and Light Tail Assumptions. Journal of the Royal Statistical Society: Series B, 82(2), 321-354.

[3] Gu, S., Kelly, B., & Xiu, D. (2020). Empirical asset pricing via machine learning. The Review of Financial Studies, 33(5), 2223-2273.

[4] Lim, B., & Zohren, S. (2021). Time-series forecasting with deep learning: a survey. Philosophical Transactions of the Royal Society A, 379(2194), 20200209.

[5] Fischer, T., & Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. European Journal of Operational Research, 270(2), 654-669.

[6] Zhang, A., Lipton, Z. C., Li, M., & Smola, A. J. (2021). Dive into deep learning. arXiv preprint arXiv:2106.11342.

[7] Bianchi, D., Büchner, M., & Tamoni, A. (2021). Bond risk premiums with machine learning. The Review of Financial Studies, 34(2), 1046-1089.

[8] Molnar, C., König, G., Herbinger, J., Freiesleben, T., Dandl, S., Scholbeck, C. A., ... & Bischl, B. (2022). General pitfalls of model-agnostic interpretation methods for machine learning models. Nature Machine Intelligence, 4(6), 541-547.

[9] Gu, S., Kelly, B., & Xiu, D. (2021). Autoencoder asset pricing models. Journal of Econometrics, 222(1), 429-450.

[10] Gu, S., Kelly, B., & Xiu, D. (2023). "Machine Learning in Asset Pricing: A Methodological Survey." The Review of Financial Studies, 36(11), 4640-4694.

[11] Bu, F., Chen, W., & Qian, P. (2022). Deep learning in asset pricing: overfitting and risk decomposition. International Journal of Financial Engineering, 9(2), 2250011.